H8S/2633 Series

H8S/2633

HD6432633

H8S/2632

HD6432632

H8S/2631

HD6432631

H8S/2633 F-ZTAT™

HD64F2633

Hardware Manual

HITACHI

ADE-602-165B
Rev. 3.0
3/20/01
Hitachi, Ltd.

Hitachi semiconductor

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

Preface

The H8S/2633 Series is a series of high-performance microcontrollers with a 32-bit H8S/2600 CPU core, and a set of on-chip supporting functions required for system configuration.

The H8S/2600 CPU can execute basic instructions in one state, and is provided with sixteen 16-bit general registers with a 32-bit internal configuration, and a concise and optimized instruction set. The CPU can handle a 16 Mbyte linear address space (architecturally 4 Gbytes). Programs based on the high-level language C can also be run efficiently.

The address space is divided into eight areas. The data bus width and access states can be selected for each of these areas, and various kinds of memory can be connected fast and easily.

Single-power-supply flash memory (F-ZTAT™*[1]), PROM (ZTAT™*[2]), and mask ROM versions are available, providing a quick and flexible response to conditions from ramp-up through full-scale volume production, even for applications with frequently changing specifications.

On-chip supporting functions include a 16-bit timer pulse unit (TPU), programmable pulse generator (PPG), 8-bit timer, 14-bit PWM timer (PWM), watchdog timer (WDT), serial communication interface (SCI, IrDA), A/D converter, D/A converter, and I/O ports. It is also possible to incorporate an on-chip PC bus interface (IIC) as an option.

In addition, DMA controller (DMAC) and data transfer controller (DTC) are provided, enabling high-speed data transfer without CPU intervention.

Use of the H8S/2633 Series enables easy implementation of compact, high-performance systems capable of processing large volumes of data.

This manual describes the hardware of the H8S/2633 Series. Refer to the H8S/2600 Series and H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Note: * F-ZTAT (Flexible-ZTAT) is a trademark of Hitachi, Ltd.

HITACHI

Main Revisions and Additions in this Edition

Page	Item		Revisions (See Manual for Details)
7	1.3.1	Pin Arrangement	Note on FWE pin added
13	1.3.2	Pin Functions in Each Operating Mode	Note 2 added
75	3.4	Pin Functions in Each Operating Mode	Table 3-3 Pin Functions in Each Mode
			Port G added
570	15.2.2	Timer Control/Status Register (TCSR)	WDT0 input clock selection
			Overflow cycle value amended
571			WDT1 input clock selection
			Overflow cycle value amended
703	18.2.4	I^2C Bus Mode Register (ICMR)	Bits 5 to 3—Transfer clock select
			Note added
731	18.3.9	Sample Flowcharts	Figure 18-14 Flowchart for Master Transmit Mode (Example)
			Completely revised
732			Figure 18-15 Flowchart for Master Receive Mode (Example)
			Completely revised
	22.1	Overview	Added
	22.2	Register Descriptions	Added
	22.3	Operation	Added
	22.4	Flash Memory Overview	Added
811	22.7.1	Program Mode	Description added
			Maximum number of writes
			WDT overflow cycle
811	22.7.2	Program-Verify Mode	Maximum number of writes amended
814			Figure 22-12 Program/Program-Verify Flowchart amended
815	22.7.3	Erase Mode	WDT overflow cycle amended
839	22.13	Flash Memory Programming and Erasing Precautions	“Do not perform overwriting. Erase the memory before reprogramming” amended

Page	Item	Revisions (See Manual for Details)
858, 859	24.1 Overview	Table 24-1 LSI Internal States in Each Mode Note added
884, 886	25.2 DC Characteristics	Table 25-2 DC Characteristics (1) Note 5 added
887, 889		Table 25-2 DC Characteristics (2) Note 6 added
918	25.4 A/D Conversion Characteristics	Position of Conversion Time amended
920	25.6 Flash Memory Characteristics	Table 25-14 Flash Memory Characteristics Erase time typ. value amended
953, 955, 960	A.2 Instruction Codes	CLRMAC, LDMAC, MAC, STMAC instructions modified

Contents

Section 1 Overview 1
1.1 Overview 1
1.2 Internal Block Diagram 6
1.3 Pin Description 7
1.3.1 Pin Arrangement 7
1.3.2 Pin Functions in Each Operating Mode 9
1.3.3 Pin Functions 14

Section 2 CPU 21
2.1 Overview 21
2.1.1 Features 21
2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU 22
2.1.3 Differences from H8/300 CPU 23
2.1.4 Differences from H8/300H CPU 23
2.2 CPU Operating Modes 24
2.3 Address Space 29
2.4 Register Configuration 30
2.4.1 Overview 30
2.4.2 General Registers 31
2.4.3 Control Registers 32
2.4.4 Initial Register Values 34
2.5 Data Formats 35
2.5.1 General Register Data Formats 35
2.5.2 Memory Data Formats 37
2.6 Instruction Set 38
2.6.1 Overview 38
2.6.2 Instructions and Addressing Modes 39
2.6.3 Table of Instructions Classified by Function 41
2.6.4 Basic Instruction Formats 48
2.7 Addressing Modes and Effective Address Calculation 50
2.7.1 Addressing Mode 50
2.7.2 Effective Address Calculation 53
2.8 Processing States 57
2.8.1 Overview 57
2.8.2 Reset State 58
2.8.3 Exception-Handling State 59
2.8.4 Program Execution State 62
2.8.5 Bus-Released State 62
2.8.6 Power-Down State 62
2.9 Basic Timing 63

2.9.1 Overview 63
2.9.2 On-Chip Memory (ROM, RAM) 63
2.9.3 On-Chip Supporting Module Access Timing 65
2.9.4 External Address Space Access Timing 66
2.10 Usage Note 66
2.10.1 TAS Instruction 66

Section 3 MCU Operating Modes 67
3.1 Overview 67
3.1.1 Operating Mode Selection 67
3.1.2 Register Configuration 68
3.2 Register Descriptions 68
3.2.1 Mode Control Register (MDCR) 68
3.2.2 System Control Register (SYSCR) 69
3.2.3 Pin Function Control Register (PFCR) 71
3.3 Operating Mode Descriptions 74
3.3.1 Mode 4 74
3.3.2 Mode 5 74
3.3.3 Mode 6 74
3.3.4 Mode 7 74
3.4 Pin Functions in Each Operating Mode 75
3.5 Address Map in Each Operating Mode 75

Section 4 Exception Handling 79
4.1 Overview 79
4.1.1 Exception Handling Types and Priority 79
4.1.2 Exception Handling Operation 80
4.1.3 Exception Vector Table 80
4.2 Reset 82
4.2.1 Overview 82
4.2.2 Types of Reset 82
4.2.3 Reset Sequence 83
4.2.4 Interrupts after Reset 85
4.2.5 State of On-Chip Supporting Modules after Reset Release 85
4.3 Traces 86
4.4 Interrupts 87
4.5 Trap Instruction 88
4.6 Stack Status after Exception Handling 89
4.7 Notes on Use of the Stack 90

Section 5 Interrupt Controller 91
5.1 Overview 91
5.1.1 Features 91
5.1.2 Block Diagram 92

5.1.3 Pin Configuration 93
5.1.4 Register Configuration 93
5.2 Register Descriptions 94
5.2.1 System Control Register (SYSCR) 94
5.2.2 Interrupt Priority Registers A to L, O (IPRA to IPRL, IPRO) 95
5.2.3 IRQ Enable Register (IER) 96
5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL) 97
5.2.5 IRQ Status Register (ISR) 98
5.3 Interrupt Sources 99
5.3.1 External Interrupts 99
5.3.2 Internal Interrupts 100
5.3.3 Interrupt Exception Handling Vector Table 100
5.4 Interrupt Operation 105
5.4.1 Interrupt Control Modes and Interrupt Operation 105
5.4.2 Interrupt Control Mode 0 108
5.4.3 Interrupt Control Mode 2 110
5.4.4 Interrupt Exception Handling Sequence 112
5.4.5 Interrupt Response Times 113
5.5 Usage Notes 114
5.5.1 Contention between Interrupt Generation and Disabling 114
5.5.2 Instructions that Disable Interrupts 115
5.5.3 Times when Interrupts are Disabled 115
5.5.4 Interrupts during Execution of EEPMOV Instruction 116
5.6 DTC and DMAC Activation by Interrupt 116
5.6.1 Overview 116
5.6.2 Block Diagram 116
5.6.3 Operation 117

Section 6 PC Break Controller (PBC) 119
6.1 Overview 119
6.1.1 Features 119
6.1.2 Block Diagram 120
6.1.3 Register Configuration 121
6.2 Register Descriptions 121
6.2.1 Break Address Register A (BARA) 121
6.2.2 Break Address Register B (BARB) 122
6.2.3 Break Control Register A (BCRA) 122
6.2.4 Break Control Register B (BCRB) 124
6.2.5 Module Stop Control Register C (MSTPCRC) 124
6.3 Operation 125
6.3.1 PC Break Interrupt Due to Instruction Fetch 125
6.3.2 PC Break Interrupt Due to Data Access 125
6.3.3 Notes on PC Break Interrupt Handling 126

6.3.4 Operation in Transitions to Power-Down Modes 126
6.3.5 PC Break Operation in Continuous Data Transfer 127
6.3.6 When Instruction Execution is Delayed by One State 128
6.3.7 Additional Notes 129

Section 7 Bus Controller 131
7.1 Overview 131
7.1.1 Features 131
7.1.2 Block Diagram 133
7.1.3 Pin Configuration 134
7.1.4 Register Configuration 135
7.2 Register Descriptions 136
7.2.1 Bus Width Control Register (ABWCR) 136
7.2.2 Access State Control Register (ASTCR) 137
7.2.3 Wait Control Registers H and L (WCRH, WCRL) 138
7.2.4 Bus Control Register H (BCRH) 142
7.2.5 Bus Control Register L (BCRL) 144
7.2.6 Pin Function Control Register (PFCR) 146
7.2.7 Memory Control Register (MCR) 149
7.2.8 DRAM Control Register (DRAMCR) 151
7.2.9 Refresh Timer Counter (RTCNT) 153
7.2.10 Refresh Time Constant Register (RTCOR) 153
7.3 Overview of Bus Control 154
7.3.1 Area Partitioning 154
7.3.2 Bus Specifications 155
7.3.3 Memory Interfaces 156
7.3.4 Interface Specifications for Each Area 157
7.3.5 Chip Select Signals 158
7.4 Basic Bus Interface 159
7.4.1 Overview 159
7.4.2 Data Size and Data Alignment 159
7.4.3 Valid Strobes 161
7.4.4 Basic Timing 162
7.4.5 Wait Control 170
7.5 DRAM Interface 172
7.5.1 Overview 172
7.5.2 Setting up DRAM Space 172
7.5.3 Address Multiplexing 173
7.5.4 Data Bus 173
7.5.5 DRAM Interface Pins 174
7.5.6 Basic Timing 174
7.5.7 Precharge State Control 176
7.5.8 Wait Control 177

7.5.9 Byte Access Control 179
7.5.10 Burst Operation 181
7.5.11 Refresh Control 185
7.6 DMAC Single Address Mode and DRAM Interface 189
7.6.1 DDS=1 189
7.6.2 DDS=0 190
7.7 Burst ROM Interface 191
7.7.1 Overview 191
7.7.2 Basic Timing 191
7.7.3 Wait Control 193
7.8 Idle Cycle 194
7.8.1 Operation 194
7.8.2 Pin States in Idle Cycle 198
7.9 Write Data Buffer Function 199
7.10 Bus Release 200
7.10.1 Overview 200
7.10.2 Operation 200
7.10.3 Pin States in External Bus Released State 201
7.10.4 Transition Timing 202
7.10.5 Notes 203
7.11 Bus Arbitration 204
7.11.1 Overview 204
7.11.2 Operation 204
7.11.3 Bus Transfer Timing 205
7.12 Resets and the Bus Controller 205

Section 8 DMA Controller 207
8.1 Overview 207
8.1.1 Features 207
8.1.2 Block Diagram 208
8.1.3 Overview of Functions 209
8.1.4 Pin Configuration 211
8.1.5 Register Configuration 212
8.2 Register Descriptions (1) (Short Address Mode) 213
8.2.1 Memory Address Registers (MAR) 214
8.2.2 I/O Address Register (IOAR) 215
8.2.3 Execute Transfer Count Register (ETCR) 215
8.2.4 DMA Control Register (DMACR) 216
8.2.5 DMA Band Control Register (DMABCR) 220
8.3 Register Descriptions (2) (Full Address Mode) 225
8.3.1 Memory Address Register (MAR) 225
8.3.2 I/O Address Register (IOAR) 225
8.3.3 Execute Transfer Count Register (ETCR) 226

8.3.4 DMA Control Register (DMACR) 227
8.3.5 DMA Band Control Register (DMABCR) 231
8.4 Register Descriptions (3) 236
8.4.1 DMA Write Enable Register (DMAWER) 236
8.4.2 DMA Terminal Control Register (DMATCR) 238
8.4.3 Module Stop Control Register (MSTPCR) 239
8.5 Operation 240
8.5.1 Transfer Modes 240
8.5.2 Sequential Mode 242
8.5.3 Idle Mode 245
8.5.4 Repeat Mode 248
8.5.5 Single Address Mode 252
8.5.6 Normal Mode 255
8.5.7 Block Transfer Mode 258
8.5.8 DMAC Activation Sources 264
8.5.9 Basic DMAC Bus Cycles 267
8.5.10 DMAC Bus Cycles (Dual Address Mode) 268
8.5.11 DMAC Bus Cycles (Single Address Mode) 276
8.5.12 Write Data Buffer Function 282
8.5.13 DMAC Multi-Channel Operation 283
8.5.14 Relation Between External Bus Requests, Refresh Cycles, the DTC, and the DMAC 285
8.5.15 NMI Interrupts and DMAC 286
8.5.16 Forced Termination of DMAC Operation 287
8.5.17 Clearing Full Address Mode 288
8.6 Interrupts 289
8.7 Usage Notes 290

Section 9 Data Transfer Controller (DTC) 295
9.1 Overview 295
9.1.1 Features 295
9.1.2 Block Diagram 296
9.1.3 Register Configuration 297
9.2 Register Descriptions 298
9.2.1 DTC Mode Register A (MRA) 298
9.2.2 DTC Mode Register B (MRB) 300
9.2.3 DTC Source Address Register (SAR) 301
9.2.4 DTC Destination Address Register (DAR) 301
9.2.5 DTC Transfer Count Register A (CRA) 301
9.2.6 DTC Transfer Count Register B (CRB) 302
9.2.7 DTC Enable Registers (DTCER) 302
9.2.8 DTC Vector Register (DTVECR) 303
9.2.9 Module Stop Control Register A (MSTPCRA) 304

9.3 Operation ... 305
9.3.1 Overview ... 305
9.3.2 Activation Sources ... 307
9.3.3 DTC Vector Table ... 308
9.3.4 Location of Register Information in Address Space ... 312
9.3.5 Normal Mode ... 313
9.3.6 Repeat Mode ... 314
9.3.7 Block Transfer Mode ... 315
9.3.8 Chain Transfer ... 317
9.3.9 Operation Timing ... 318
9.3.10 Number of DTC Execution States ... 319
9.3.11 Procedures for Using DTC ... 321
9.3.12 Examples of Use of the DTC ... 322
9.4 Interrupts ... 325
9.5 Usage Notes ... 325

Section 10 I/O Ports ... 327
10.1 Overview ... 327
10.2 Port 1 ... 332
10.2.1 Overview ... 332
10.2.2 Register Configuration ... 333
10.2.3 Pin Functions ... 335
10.3 Port 3 ... 347
10.3.1 Overview ... 347
10.3.2 Register Configuration ... 347
10.3.3 Pin Functions ... 350
10.4 Port 4 ... 353
10.4.1 Overview ... 353
10.4.2 Register Configuration ... 354
10.4.3 Pin Functions ... 354
10.5 Port 7 ... 355
10.5.1 Overview ... 355
10.5.2 Register Configuration ... 356
10.5.3 Pin Functions ... 358
10.6 Port 9 ... 361
10.6.1 Overview ... 361
10.6.2 Register Configuration ... 362
10.6.3 Pin Functions ... 362
10.7 Port A ... 363
10.7.1 Overview ... 363
10.7.2 Register Configuration ... 364
10.7.3 Pin Functions ... 367
10.7.4 MOS Input Pull-Up Function ... 367

10.8 Port B 369
10.8.1 Overview 369
10.8.2 Register Configuration 370
10.8.3 Pin Functions 373
10.8.4 MOS Input Pull-Up Function 374
10.9 Port C 375
10.9.1 Overview 375
10.9.2 Register Configuration 376
10.9.3 Pin Functions for Each Mode 379
10.9.4 MOS Input Pull-Up Function 381
10.10 Port D 382
10.10.1 Overview 382
10.10.2 Register Configuration 383
10.10.3 Pin Functions 385
10.10.4 MOS Input Pull-Up Function 386
10.11 Port E 387
10.11.1 Overview 387
10.11.2 Register Configuration 388
10.11.3 Pin Functions 390
10.11.4 MOS Input Pull-Up Function 391
10.12 Port F 392
10.12.1 Overview 392
10.12.2 Register Configuration 393
10.12.3 Pin Functions 395
10.13 Port G 397
10.13.1 Overview 397
10.13.2 Register Configuration 398
10.13.3 Pin Functions 400

Section 11 16-Bit Timer Pulse Unit (TPU) 403
11.1 Overview 403
11.1.1 Features 403
11.1.2 Block Diagram 407
11.1.3 Pin Configuration 408
11.1.4 Register Configuration 410
11.2 Register Descriptions 412
11.2.1 Timer Control Register (TCR) 412
11.2.2 Timer Mode Register (TMDR) 417
11.2.3 Timer I/O Control Register (TIOR) 419
11.2.4 Timer Interrupt Enable Register (TIER) 432
11.2.5 Timer Status Register (TSR) 435
11.2.6 Timer Counter (TCNT) 439
11.2.7 Timer General Register (TGR) 440

11.2.8 Timer Start Register (TSTR) 441
11.2.9 Timer Synchro Register (TSYR) 442
11.2.10 Module Stop Control Register A (MSTPCRA) 443
11.3 Interface to Bus Master 444
11.3.1 16-Bit Registers 444
11.3.2 8-Bit Registers 444
11.4 Operation 446
11.4.1 Overview 446
11.4.2 Basic Functions 447
11.4.3 Synchronous Operation 453
11.4.4 Buffer Operation 455
11.4.5 Cascaded Operation 459
11.4.6 PWM Modes 461
11.4.7 Phase Counting Mode 466
11.5 Interrupts 473
11.5.1 Interrupt Sources and Priorities 473
11.5.2 DTC/DMAC Activation 475
11.5.3 A/D Converter Activation 475
11.6 Operation Timing 476
11.6.1 Input/Output Timing 476
11.6.2 Interrupt Signal Timing 480
11.7 Usage Notes 484

Section 12 Programmable Pulse Generator (PPG) 495
12.1 Overview 495
12.1.1 Features 495
12.1.2 Block Diagram 496
12.1.3 Pin Configuration 497
12.1.4 Registers 498
12.2 Register Descriptions 499
12.2.1 Next Data Enable Registers H and L (NDERH, NDERL) 499
12.2.2 Output Data Registers H and L (PODRH, PODRL) 500
12.2.3 Next Data Registers H and L (NDRH, NDRL) 501
12.2.4 Notes on NDR Access 501
12.2.5 PPG Output Control Register (PCR) 503
12.2.6 PPG Output Mode Register (PMR) 505
12.2.7 Port 1 Data Direction Register (P1DDR) 508
12.2.8 Module Stop Control Register A (MSTPCRA) 508
12.3 Operation 509
12.3.1 Overview 509
12.3.2 Output Timing 510
12.3.3 Normal Pulse Output 511
12.3.4 Non-Overlapping Pulse Output 513

12.3.5 Inverted Pulse Output 516
12.3.6 Pulse Output Triggered by Input Capture 517
12.4 Usage Notes 518

Section 13 8-Bit Timers (TMR) 521
13.1 Overview 521
13.1.1 Features 521
13.1.2 Block Diagram 522
13.1.3 Pin Configuration 523
13.1.4 Register Configuration 524
13.2 Register Descriptions 525
13.2.1 Timer Counters 0 to 3 (TCNT0 to TCNT3) 525
13.2.2 Time Constant Registers A0 to A3 (TCORA0 to TCORA3) 525
13.2.3 Time Constant Registers B0 to B3 (TCORB0 to TCORB3) 526
13.2.4 Timer Control Registers 0 to 3 (TCR0 to TCR3) 526
13.2.5 Timer Control/Status Registers 0 to 3 (TCSR0 to TCSR3) 529
13.2.6 Module Stop Control Register A (MSTPCRA) 532
13.3 Operation 533
13.3.1 TCNT Incrementation Timing 533
13.3.2 Compare Match Timing 534
13.3.3 Timing of External RESET on TCNT 536
13.3.4 Timing of Overflow Flag (OVF) Setting 536
13.3.5 Operation with Cascaded Connection 537
13.4 Interrupts 538
13.4.1 Interrupt Sources and DTC Activation 538
13.4.2 A/D Converter Activation 538
13.5 Sample Application 539
13.6 Usage Notes 540
13.6.1 Contention between TCNT Write and Clear 540
13.6.2 Contention between TCNT Write and Increment 541
13.6.3 Contention between TCOR Write and Compare Match 542
13.6.4 Contention between Compare Matches A and B 543
13.6.5 Switching of Internal Clocks and TCNT Operation 543
13.6.6 Interrupts and Module Stop Mode 545

Section 14 14-Bit PWM D/A 547
14.1 Overview 547
14.1.1 Features 547
14.1.2 Block Diagram 548
14.1.3 Pin Configuration 549
14.1.4 Register Configuration 549
14.2 Register Descriptions 550
14.2.1 PWM D/A Counter (DACNT) 550

14.2.2 PWM D/A Data Registers A and B (DADRA and DADRB) 551
14.2.3 PWM D/A Control Register (DACR) 552
14.2.4 Module Stop Control Register B (MSTPCRB) 554
14.3 Bus Master Interface 555
14.4 Operation 558

Section 15 Watchdog Timer 563
15.1 Overview 563
15.1.1 Features 563
15.1.2 Block Diagram 564
15.1.3 Pin Configuration 566
15.1.4 Register Configuration 566
15.2 Register Descriptions 567
15.2.1 Timer Counter (TCNT) 567
15.2.2 Timer Control/Status Register (TCSR) 567
15.2.3 Reset Control/Status Register (RSTCSR) 572
15.2.4 Pin Function Control Register (PFCR) 573
15.2.5 Notes on Register Access 574
15.3 Operation 576
15.3.1 Watchdog Timer Operation 576
15.3.2 Interval Timer Operation 578
15.3.3 Timing of Setting Overflow Flag (OVF) 578
15.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF) 579
15.4 Interrupts 580
15.5 Usage Notes 580
15.5.1 Contention between Timer Counter (TCNT) Write and Increment 580
15.5.2 Changing Value of CKS2 to CKS0 581
15.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode 581
15.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal 581
15.5.5 Internal Reset in Watchdog Timer Mode 581

Section 16 Serial Communication Interface (SCI, IrDA) 583
16.1 Overview 583
16.1.1 Features 583
16.1.2 Block Diagram 585
16.1.3 Pin Configuration 586
16.1.4 Register Configuration 587
16.2 Register Descriptions 589
16.2.1 Receive Shift Register (RSR) 589
16.2.2 Receive Data Register (RDR) 589
16.2.3 Transmit Shift Register (TSR) 590
16.2.4 Transmit Data Register (TDR) 590
16.2.5 Serial Mode Register (SMR) 591

16.2.6 Serial Control Register (SCR) 594
16.2.7 Serial Status Register (SSR) 598
16.2.8 Bit Rate Register (BRR) 602
16.2.9 Smart Card Mode Register (SCMR) 611
16.2.10 IrDA Control Register (IrCR) 612
16.2.11 Module Stop Control Registers B and C (MSTPCRB, MSTPCRC) 613
16.3 Operation 615
16.3.1 Overview 615
16.3.2 Operation in Asynchronous Mode 618
16.3.3 Multiprocessor Communication Function 629
16.3.4 Operation in Clocked Synchronous Mode 637
16.3.5 IrDA Operation 645
16.4 SCI Interrupts 648
16.5 Usage Notes 650

Section 17 Smart Card Interface 659
17.1 Overview 659
17.1.1 Features 659
17.1.2 Block Diagram 660
17.1.3 Pin Configuration 661
17.1.4 Register Configuration 662
17.2 Register Descriptions 664
17.2.1 Smart Card Mode Register (SCMR) 664
17.2.2 Serial Status Register (SSR) 666
17.2.3 Serial Mode Register (SMR) 668
17.2.4 Serial Control Register (SCR) 670
17.3 Operation 671
17.3.1 Overview 671
17.3.2 Pin Connections 671
17.3.3 Data Format 673
17.3.4 Register Settings 675
17.3.5 Clock 677
17.3.6 Data Transfer Operations 679
17.3.7 Operation in GSM Mode 686
17.3.8 Operation in Block Transfer Mode 687
17.4 Usage Notes 688

Section 18 I^2C Bus Interface [Option] 691
18.1 Overview 691
18.1.1 Features 691
18.1.2 Block Diagram 692
18.1.3 Input/Output Pins 694
18.1.4 Register Configuration 695

18.2 Register Descriptions 696
18.2.1 I²C Bus Data Register (ICDR) 696
18.2.2 Slave Address Register (SAR) 699
18.2.3 Second Slave Address Register (SARX) 700
18.2.4 I²C Bus Mode Register (ICMR) 701
18.2.5 I²C Bus Control Register (ICCR) 704
18.2.6 I²C Bus Status Register (ICSR) 711
18.2.7 Serial Control Register X (SCRX) 716
18.2.8 DDC Switch Register (DDCSWR) 717
18.2.9 Module Stop Control Register B (MSTPCRB) 718
18.3 Operation 719
18.3.1 I²C Bus Data Format 719
18.3.2 Master Transmit Operation 720
18.3.3 Master Receive Operation 722
18.3.4 Slave Receive Operation 724
18.3.5 Slave Transmit Operation 726
18.3.6 IRIC Setting Timing and SCL Control 728
18.3.7 Operation Using the DTC 729
18.3.8 Noise Canceler 730
18.3.9 Sample Flowcharts 730
18.3.10 Initialization of Internal State 734
18.4 Usage Notes 736

Section 19 A/D Converter 745
19.1 Overview 745
19.1.1 Features 745
19.1.2 Block Diagram 746
19.1.3 Pin Configuration 747
19.1.4 Register Configuration 748
19.2 Register Descriptions 749
19.2.1 A/D Data Registers A to D (ADDRA to ADDRD) 749
19.2.2 A/D Control/Status Register (ADCSR) 750
19.2.3 A/D Control Register (ADCR) 753
19.2.4 Module Stop Control Register A (MSTPCRA) 754
19.3 Interface to Bus Master 755
19.4 Operation 756
19.4.1 Single Mode (SCAN = 0) 756
19.4.2 Scan Mode (SCAN = 1) 758
19.4.3 Input Sampling and A/D Conversion Time 760
19.4.4 External Trigger Input Timing 761
19.5 Interrupts 762
19.6 Usage Notes 762

Section 20 D/A Converter ... 769
20.1 Overview ... 769
20.1.1 Features ... 769
20.1.2 Block Diagram ... 769
20.1.3 Input and Output Pins ... 771
20.1.4 Register Configuration ... 771
20.2 Register Descriptions ... 772
20.2.1 D/A Data Registers 0 to 3 (DADR0 to DADR3) ... 772
20.2.2 D/A Control Register 01 and 23 (DACR01 and DACR23) ... 772
20.2.3 Module Stop Control Register A and C (MSTPCRA and MSTPCRC) ... 774
20.3 Operation ... 776

Section 21 RAM ... 777
21.1 Overview ... 777
21.1.1 Block Diagram ... 777
21.1.2 Register Configuration ... 778
21.2 Register Descriptions ... 778
21.2.1 System Control Register (SYSCR) ... 778
21.3 Operation ... 779
21.4 Usage Notes ... 779

Section 22 ROM ... 781
22.1 Overview ... 781
22.1.1 Block Diagram ... 781
22.1.2 Register Configuration ... 781
22.2 Register Descriptions ... 782
22.2.1 Mode Control Register (MDCR) ... 782
22.3 Operation ... 782
22.4 Flash Memory Overview ... 785
22.4.1 Features ... 785
22.4.2 Overview ... 786
22.4.3 Flash Memory Operating Modes ... 787
22.4.4 On-Board Programming Modes ... 788
22.4.5 Flash Memory Emulation in RAM ... 790
22.4.6 Differences between Boot Mode and User Program Mode ... 791
22.4.7 Block Configuration ... 792
22.4.8 Pin Configuration ... 792
22.4.9 Register Configuration ... 793
22.5 Register Descriptions ... 793
22.5.1 Flash Memory Control Register 1 (FLMCR1) ... 793
22.5.2 Flash Memory Control Register 2 (FLMCR2) ... 796
22.5.3 Erase Block Register 1 (EBR1) ... 797

22.5.4 Erase Block Register 2 (EBR2) 798
22.5.5 RAM Emulation Register (RAMER) 799
22.5.6 Flash Memory Power Control Register (FLPWCR) 801
22.5.7 Serial Control Register X (SCRX) 801
22.6 On-Board Programming Modes 802
22.6.1 Boot Mode 803
22.6.2 User Program Mode 807
22.7 Programming/Erasing Flash Memory 809
22.7.1 Program Mode 810
22.7.2 Program-Verify Mode 811
22.7.3 Erase Mode 815
22.7.4 Erase-Verify Mode 815
22.8 Protection 817
22.8.1 Hardware Protection 817
22.8.2 Software Protection 818
22.8.3 Error Protection 819
22.9 Flash Memory Emulation in RAM 821
22.10 Interrupt Handling when Programming/Erasing Flash Memory 823
22.11 Flash Memory Programmer Mode 823
22.11.1 Socket Adapter Pin Correspondence Diagram 824
22.11.2 Programmer Mode Operation 826
22.11.3 Memory Read Mode 827
22.11.4 Auto-Program Mode 830
22.11.5 Auto-Erase Mode 832
22.11.6 Status Read Mode 834
22.11.7 Status Polling 835
22.11.8 Programmer Mode Transition Time 835
22.11.9 Notes on Memory Programming 836
22.12 Flash Memory and Power-Down States 837
22.12.1 Note on Power-Down States 837
22.13 Flash Memory Programming and Erasing Precautions 838
22.14 Note on Switching from F-ZTAT Version to Mask ROM Version 843

Section 23 Clock Pulse Generator 845
23.1 Overview 845
23.1.1 Block Diagram 845
23.1.2 Register Configuration 846
23.2 Register Descriptions 846
23.2.1 System Clock Control Register (SCKCR) 846
23.2.2 Low-Power Control Register (LPWRCR) 847
23.3 Oscillator 848
23.3.1 Connecting a Crystal Resonator 848
23.3.2 External Clock Input 851

23.4 PLL Circuit 853
23.5 Medium-Speed Clock Divider 853
23.6 Bus Master Clock Selection Circuit 853
23.7 Subclock Oscillator 854
23.8 Subclock Waveform Shaping Circuit 855
23.9 Note on Crystal Resonator 855

Section 24 Power-Down Modes 857
24.1 Overview 857
24.1.1 Register Configuration 861
24.2 Register Descriptions 862
24.2.1 Standby Control Register (SBYCR) 862
24.2.2 System Clock Control Register (SCKCR) 864
24.2.3 Low-Power Control Register (LPWRCR) 865
24.2.4 Timer Control/Status Register (TCSR) 868
24.2.5 Module Stop Control Register (MSTPCR) 869
24.3 Medium-Speed Mode 870
24.4 Sleep Mode 871
24.4.1 Sleep Mode 871
24.4.2 Exiting Sleep Mode 871
24.5 Module Stop Mode 872
24.5.1 Module Stop Mode 872
24.5.2 Usage Notes 874
24.6 Software Standby Mode 874
24.6.1 Software Standby Mode 874
24.6.2 Exiting Software Standby Mode 874
24.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode 875
24.6.4 Software Standby Mode Application Example 876
24.6.5 Usage Notes 877
24.7 Hardware Standby Mode 877
24.7.1 Hardware Standby Mode 877
24.7.2 Hardware Standby Mode Timing 878
24.8 Watch Mode 878
24.8.1 Watch Mode 878
24.8.2 Exiting Watch Mode 879
24.8.3 Notes 879
24.9 Sub-Sleep Mode 880
24.9.1 Sub-Sleep Mode 880
24.9.2 Exiting Sub-Sleep Mode 880
24.10 Sub-Active Mode 881
24.10.1 Sub-Active Mode 881
24.10.2 Exiting Sub-Active Mode 881
24.11 Direct Transitions 882

24.11.1 Overview of Direct Transitions 882
24.12 ø Clock Output Disabling Function 882

Section 25 Electrical Characteristics 883
25.1 Absolute Maximum Ratings 883
25.2 DC Characteristics 884
25.3 AC Characteristics 892
25.3.1 Clock Timing 893
25.3.2 Control Signal Timing 895
25.3.3 Bus Timing 897
25.3.4 DMAC Timing 906
25.3.5 Timing of On-Chip Supporting Modules 910
25.4 A/D Conversion Characteristics 918
25.5 D/A Conversion Characteristics 919
25.6 Flash Memory Characteristics 920
25.7 Usage Note 921

Appendix A Instruction Set 923
A.1 Instruction List 923
A.2 Instruction Codes 947
A.3 Operation Code Map 962
A.4 Number of States Required for Instruction Execution 966
A.5 Bus States During Instruction Execution 980
A.6 Condition Code Modification 994

Appendix B Internal I/O Register 1000
B.1 Addresses 1000
B.2 Functions 1010

Appendix C I/O Port Block Diagrams 1108
C.1 Port 1 Block Diagram 1108
C.2 Port 3 Block Diagram 1114
C.3 Port 4 Block Diagram 1122
C.4 Port 7 Block Diagram 1123
C.5 Port 9 Block Diagram 1130
C.6 Port A Block Diagram 1131
C.7 Port B Block Diagram 1135
C.8 Port C Block Diagram 1136
C.9 Port D Block Diagram 1138
C.10 Port E Block Diagram 1139
C.11 Port E Block Diagram 1140
C.12 Port G Block Diagram 1148

Appendix D Pin States 1152
D.1 Port States in Each Mode 1152

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode 1156

Appendix F Product Code Lineup 1157

Appendix G Package Dimensions 1158

Section 1 Overview

1.1 Overview

The H8S/2633 Series is a series of microcomputers (MCUs: microcomputer units), built around the H8S/2600 CPU, employing Hitachi's proprietary architecture, and equipped with peripheral functions on-chip.

The H8S/2600 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit general registers and a concise, optimized instruction set designed for high-speed operation, and can address a 16-Mbyte linear address space. The instruction set is upward-compatible with H8/300 and H8/300H CPU instructions at the object-code level, facilitating migration from the H8/300, H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include DMA controller (DMAC), data transfer controller (DTC) bus masters, ROM and RAM memory, a16-bit timer-pulse unit (TPU), programmable pulse generator (PPG), 8-bit timer, 14-bit PWM timer (PWM) watchdog timer (WDT), serial communication interface (SCI, IrDA), A/D converter, D/A converter, and I/O ports. It is also possible to incorporate an on-chip PC bus interface (IIC) as an option.

On-chip ROM is available as 256-kbyte flash memory (F-ZTAT™ version)* or as 256-, 128-, or 64-kbyte mask ROM. ROM is connected to the CPU via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching has been speeded up, and processing speed increased.

Four operating modes, modes 4 to 7, are provided, and there is a choice of single-chip mode or external expansion mode.

The features of the H8S/2633 Series are shown in table 1-1.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Table 1-1 Overview

Item	Specification
CPU	• General-register machine — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers) • High-speed operation suitable for realtime control — Maximum clock rate: 25 MHz — High-speed arithmetic operations 8/16/32-bit register-register add/subtract : 40 ns 16 × 16-bit register-register multiply : 160 ns 16 × 16 + 42-bit multiply and accumulate : 160 ns 32 ÷ 16-bit register-register divide : 800 ns • Instruction set suitable for high-speed operation — Sixty-nine basic instructions — 8/16/32-bit move/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Multiply-and accumulate instruction — Powerful bit-manipulation instructions • Two CPU operating modes — Normal mode: 64-kbyte address space (cannot be used in the H8S/2633 Series) — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications settable independently for each area • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each area • Number of program wait states can be set for each area • Burst ROM directly connectable • Possible to connect a maximum of 8 MB of DRAM (alternatively, it is also possible to use an interval timer) • External bus release function
PC break controller	• Supports debugging functions by means of PC break interrupts • Two break channels
DMA controller (DMAC)	• Short address mode and full address mode selectable • Short address mode: 4 channels Full address mode: 2 channels • Transfer possible in repeat mode/block transfer mode Transfer possible in single address mode • Activation by internal interrupt possible

Item	Specification
Data transfer controller (DTC)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one activation source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit timer-pulse unit (TPU)	• 6-channel 16-bit timer on-chip • Pulse I/O processing capability for up to 16 pins' • Automatic 2-phase encoder count capability
Programmable pulse generator (PPG)	• Maximum 16-bit pulse output possible with TPU as time base • Output trigger selectable in 4-bit groups • Non-overlap margin can be set • Direct output or inverse output setting possible
8-bit timer 4 channels	• 8-bit up counter (external event count possible) • Time constant register × 2 • 2 channel connection possible
Watchdog timer 2 channels	• Watchdog timer or interval timer selectable • Operation using sub-clock supported (WDT1 only)
14-bit PWM timer (PWM)	• Maximum of 4 outputs • Resolution: 1/16384 • Maximum carrier frequency: 390.6 kHz (operating at 25 MHz)
Serial communication interface (SCI) 5 channels (SCI0 to SCI4)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
IrDA-equipped SCI 1 channel (SCI0)	• Supports IrDA standard version 1.0 • TxD and RxD encoding/decoding in IrDA format • Start/stop synchronization mode or clock synchronization mode selectable • Multiprocessor communications function • Smart card interface function

Item	Specification
A/D converter	• Resolution: 10 bits • Input: 16 channels • High-speed conversion: 10.72 µs minimum conversion time (at 25 MHz operation) • Single or scan mode selectable • Sample and hold circuit • A/D conversion can be activated by external trigger or timer trigger
D/A converter	• Resolution: 8 bits • Output: 4 channels
I/O ports	• 73 I/O pins, 16 input-only pins
Memory	• PROM or mask ROM • High-speed static RAM

Product Name	ROM	RAM
H8S/2633	256 kbytes	16 kbytes
H8S/2632	192 kbytes	12 kbytes
H8S/2631	128 kbytes	8 kbytes

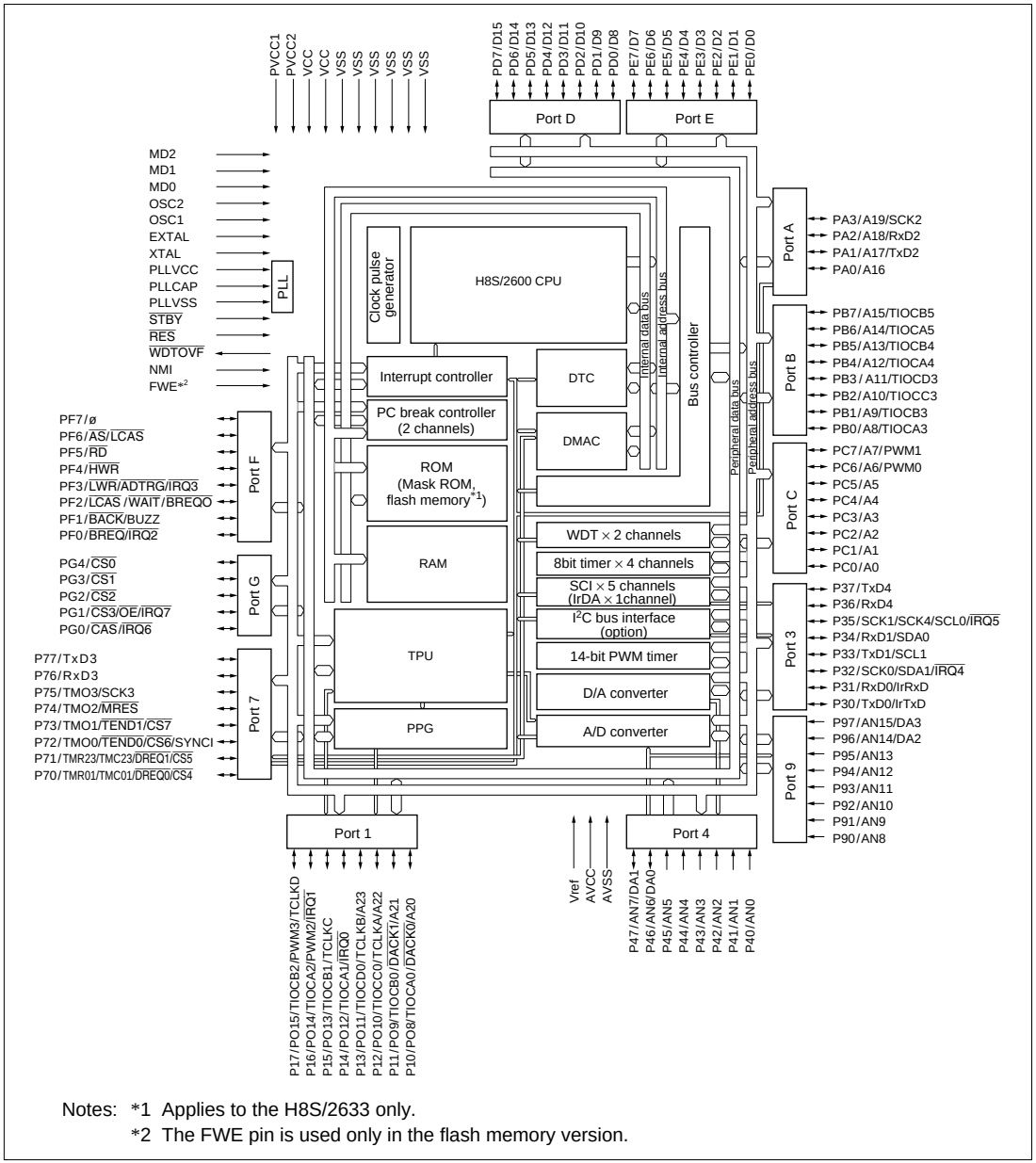
Interrupt controller	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) • 72 internal interrupt sources (including options) • Eight priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode • Sub-clock operation (sub-active mode, sub-sleep mode, watch mode)

Item	Specification				
Operating modes	Four MCU operating modes				
	Mode	CPU Operating Mode	Description	On-Chip ROM	External Data Bus: Initial Value	External Data Bus: Maximum Value
	4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits	16 bits
	5		On-chip ROM disabled expansion mode	Disabled	8 bits	16 bits
	6		On-chip ROM enabled expansion mode	Enabled	8 bits	16 bits
	7		Single-chip mode	Enabled	—	—
Clock pulse generator	• On-chip PLL circuit (×1, ×2, ×4) • Input clock frequency: 2 to 25 MHz					
Packages	• 120-pin plastic TQFP (TFP-120) • 128-pin plastic QFP (FP-128)					
I^2C bus interface (IIC) 2 channels (optional)	• Conforms to I^2C bus interface type advocated by Philips • Single master mode/slave mode • Possible to determine arbitration lost conditions • Supports two slave addresses					
Product lineup		Model Name				
	Mask ROM Version	F-ZTAT Version	ROM/RAM (Bytes)	Packages		
	HD6432633*	HD64F2633	256 k/16 k	TFP-120 FP-128		
	HD6432632*	—	192 k/12 k	TFP-120 FP-128		
	HD6432631*	—	128 k/8 k	TFP-120 FP-128		

Note: * In the planning stage.

1.2 Internal Block Diagram

Figure 1-1 shows an internal block diagram.



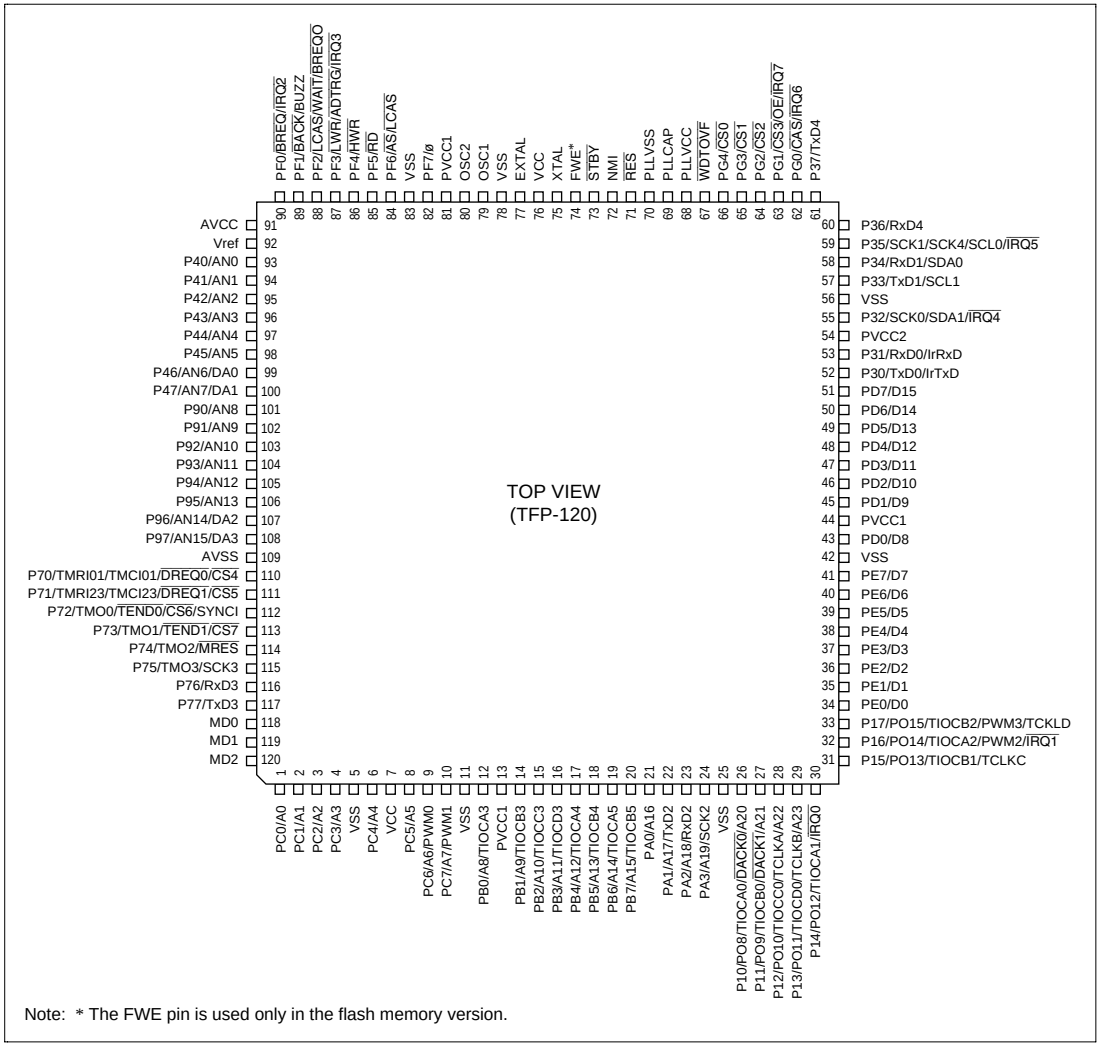
Notes: *1 Applies to the H8S/2633 only.
*2 The FWE pin is used only in the flash memory version.

Figure 1-1 Internal Block Diagram

1.3 Pin Description

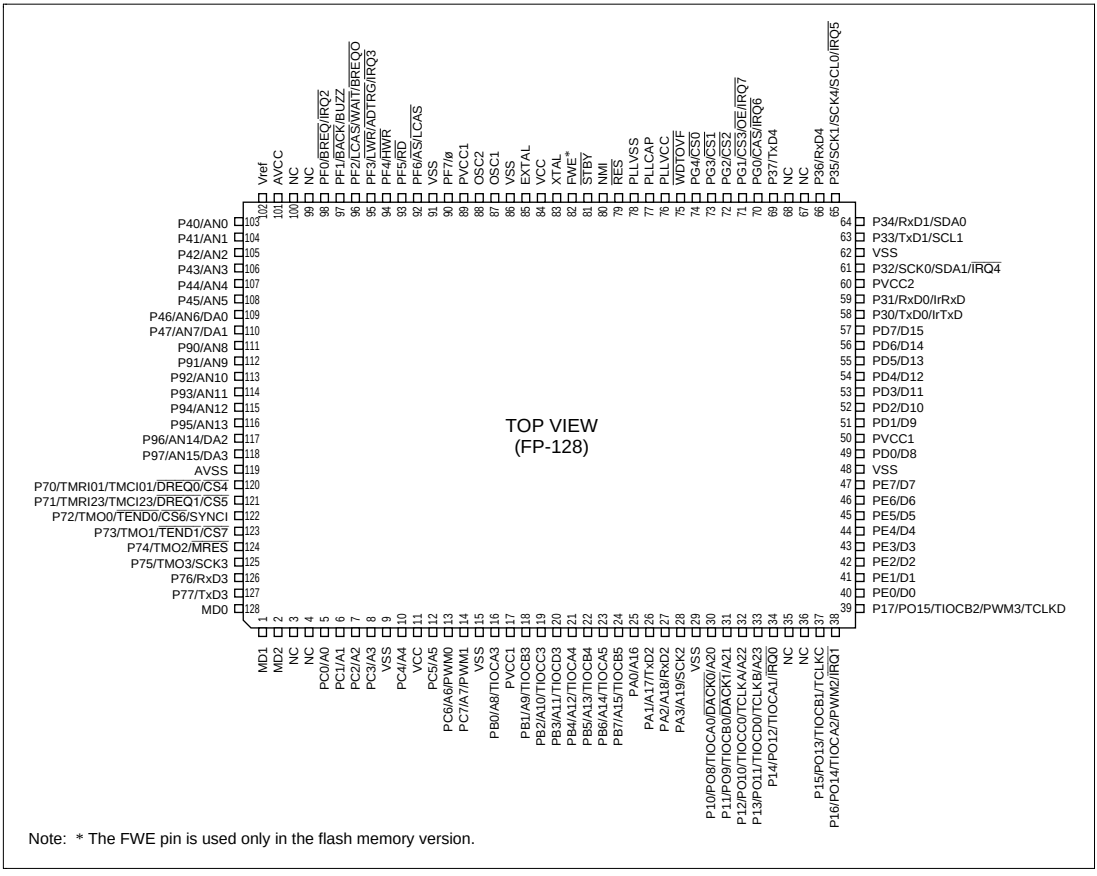
1.3.1 Pin Arrangement

Figure 1-2 shows the pin arrangement of the H8S/2633 Series.



Note: * The FWE pin is used only in the flash memory version.

Figure 1-2 Pin Arrangement (TFP-120: Top View)



Note: * The FWE pin is used only in the flash memory version.

Figure 1-3 Pin Arrangement (FP-128: Top View)

1.3.2 Pin Functions in Each Operating Mode

Table 1-2 shows the pin functions of the H8S/2633 Series in each of the operating modes.

Table 1-2 Pin Functions in Each Operating Mode

Pin No.		Pin Name			
TFP-120	**FP-128**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
1	5	A0	A0	PC0/A0	PC0
2	6	A1	A1	PC1/A1	PC1
3	7	A2	A2	PC2/A2	PC2
4	8	A3	A3	PC3/A3	PC3
5	9	VSS	VSS	VSS	VSS
6	10	A4	A4	PC4/A4	PC4
7	11	VCC	VCC	VCC	VCC
8	12	A5	A5	PC5/A5	PC5
9	13	A6	A6	PC6/A6/PWM0	PC6/PWM0
10	14	A7	A7	PC7/A7/PWM1	PC7/PWM1
11	15	VSS	VSS	VSS	VSS
12	16	A8	A8	PB0/A8/TIOCA3	PB0/TIOCA3
13	17	PVCC1	PVCC1	PVCC1	PVCC1
14	18	A9	A9	PB1/A9/TIOCB3	PB1/TIOCB3
15	19	A11	A11	PB3/A11/TIOCD3	PB3/TIOCD3
17	21	A12	A12	PB4/A12/TIOCA4	PB4/TIOCA4
18	22	A13	A13	PB5/A13/TIOCB4	PB5/TIOCB4
19	23	A14	A14	PB6/A14/TIOCA5	PB6/TIOCA5
20	24	A15	A15	PB7/A15/TIOCB5	PB7/TIOCB5
21	25	A16	A16	PA0/A16	PA0
22	26	A17	A17	PA1/A17/TxD2	PA1/TxD2
23	27	A18	A18	PA2/A18/RxD2	PA2/RxD2
24	28	A19	A19	PA3/A19/SCK2	PA3/SCK2
25	29	VSS	VSS	VSS	VSS
26	30	P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$/A20	P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$/A20	P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$/A20	P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$
27	31	P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$/A21	P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$/A21	P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$/A21	P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$

Pin No.		Pin Name			
TFP-120	**FP-128**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
28	32	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA
29	33	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB
30	34	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$
—	35	NC	NC	NC	NC
—	36	NC	NC	NC	NC
31	37	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC
32	38	P16/PO14/TIOCA2/ PWM2/$\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ PWM2/$\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ PWM2/$\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ PWM2/$\overline{\text{IRQ1}}$
33	39	P17/PO15/TIOCB2/ PWM3/TCLKD	P17/PO15/TIOCB2/ PWM3/TCLKD	P17/PO15/TIOCB2/ PWM3/TCLKD	P17/PO15/TIOCB2/ PWM3/TCLKD
34	40	D0	PE0/D0	PE0/D0	PE0
35	41	D1	PE1/D1	PE1/D1	PE1
36	42	D2	PE2/D2	PE2/D2	PE2
37	43	D3	PE3/D3	PE3/D3	PE3
38	44	D4	PE4/D4	PE4/D4	PE4
39	45	D5	PE5/D5	PE5/D5	PE5
40	46	D6	PE6/D6	PE6/D6	PE6
41	47	D7	PE7/D7	PE7/D7	PE7
42	48	VSS	VSS	VSS	VSS
43	49	D8	D8	D8	PD0
44	50	PVCC1	PVCC1	PVCC1	PVCC1
45	51	D9	D9	D9	PD1
46	52	D10	D10	D10	PD2
47	53	D11	D11	D11	PD3
48	54	D12	D12	D12	PD4
49	55	D13	D13	D13	PD5
50	56	D14	D14	D14	PD6
51	57	D15	D15	D15	PD7
52	58	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD

Pin No.		Pin Name			
TFP-120	**FP-128**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
53	59	P31/RxD0/IrRxD	P31/RxD0/IrRxD	P31/RxD0/IrRxD	P31/RxD0/IrRxD
54	60	PVCC2	PVCC2	PVCC2	PVCC2
55	61	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$
56	62	VSS	VSS	VSS	VSS
57	63	P33/TxD1/SCL1	P33/TxD1/SCL1	P33/TxD1/SCL1	P33/TxD1/SCL1
58	64	P34/RxD1/SDA0	P34/RxD1/SDA0	P34/RxD1/SDA0	P34/RxD1/SDA0
59	65	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$
60	66	P36/RxD4	P36/RxD4	P36/RxD4	P36/RxD4
—	67	NC	NC	NC	NC
—	68	NC	NC	NC	NC
61	69	P37/TxD4	P37/TxD4	P37/TxD4	P37/TxD4
62	70	PG0/$\overline{\text{CAS}}$/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{CAS}}$/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{CAS}}$/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{IRQ6}}$
63	71	PG1/$\overline{\text{CS3}}$/$\overline{\text{OE}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{OE}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{OE}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{IRQ7}}$
64	72	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2
65	73	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3
66	74	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4
67	75	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
68	76	PLLVCC	PLLVCC	PLLVCC	PLLVCC
69	77	PLLCAP	PLLCAP	PLLCAP	PLLCAP
70	78	PLLVSS	PLLVSS	PLLVSS	PLLVSS
71	79	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
72	80	NMI	NMI	NMI	NMI
73	81	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
74	82	FWE	FWE	FWE	FWE
75	83	XTAL	XTAL	XTAL	XTAL
76	84	VCC	VCC	VCC	VCC
77	85	EXTAL	EXTAL	EXTAL	EXTAL
78	86	VSS	VSS	VSS	VSS
79	87	OSC1	OSC1	OSC1	OSC1

Pin No.		Pin Name			
TFP-120	**FP-128**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
80	88	OSC2	OSC2	OSC2	OSC2
81	89	PVCC1	PVCC1	PVCC1	PVCC1
82	90	PF7/ø	PF7/ø	PF7/ø	PF7/ø
83	91	VSS	VSS	VSS	VSS
84	92	$\overline{AS}$/$\overline{LCAS}$	$\overline{AS}$/$\overline{LCAS}$	$\overline{AS}$/$\overline{LCAS}$	PF6
85	93	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF5
86	94	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	PF4
87	95	$\overline{LWR}$	PF3/$\overline{LWR}$/$\overline{ADTRG}$/ $\overline{IRQ3}$	PF3/$\overline{LWR}$/$\overline{ADTRG}$/ $\overline{IRQ3}$	PF3/$\overline{ADTRG}$/$\overline{IRQ3}$
88	96	PF2/$\overline{LCAS}$/$\overline{WAIT}$/ $\overline{BREQO}$	PF2/$\overline{LCAS}$/$\overline{WAIT}$/ $\overline{BREQO}$	PF2/$\overline{LCAS}$/$\overline{WAIT}$/ $\overline{BREQO}$	PF2
89	97	PF1/$\overline{BACK}$/BUZZ	PF1/$\overline{BACK}$/BUZZ	PF1/$\overline{BACK}$/BUZZ	PF1/BUZZ
90	98	PF0/$\overline{BREQ}$/$\overline{IRQ2}$	PF0/$\overline{BREQ}$/$\overline{IRQ2}$	PF0/$\overline{BREQ}$/$\overline{IRQ2}$	PF0/$\overline{IRQ2}$
—	99	NC	NC	NC	NC
—	100	NC	NC	NC	NC
91	101	AVCC	AVCC	AVCC	AVCC
92	102	Vref	Vref	Vref	Vref
93	103	P40/AN0	P40/AN0	P40/AN0	P40/AN0
94	104	P41/AN1	P41/AN1	P41/AN1	P41/AN1
95	105	P42/AN2	P42/AN2	P42/AN2	P42/AN2
96	106	P43/AN3	P43/AN3	P43/AN3	P43/AN3
97	107	P44/AN4	P44/AN4	P44/AN4	P44/AN4
98	108	P45/AN5	P45/AN5	P45/AN5	P45/AN5
99	109	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0
100	110	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1
101	111	P90/AN8	P90/AN8	P90/AN8	P90/AN8
102	112	P91/AN9	P91/AN9	P91/AN9	P91/AN9
103	113	P92/AN10	P92/AN10	P92/AN10	P92/AN10
104	114	P93/AN11	P93/AN11	P93/AN11	P93/AN11
105	115	P94/AN12	P94/AN12	P94/AN12	P94/AN12
106	116	P95/AN13	P95/AN13	P95/AN13	P95/AN13

Pin No.		Pin Name			
TFP-120	**FP-128**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
107	117	P96/AN14/DA2	P96/AN14/DA2	P96/AN14/DA2	P96/AN14/DA2
108	118	P97/AN15/DA3	P97/AN15/DA3	P97/AN15/DA3	P97/AN15/DA3
109	119	AVSS	AVSS	AVSS	AVSS
110	120	P70/TMRI01/TMCI01/ $\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/TMRI01/TMCI01/ $\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/TMRI01/TMCI01/ $\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/TMRI01/TMCI01/ $\overline{\text{DREQ0}}$
111	121	P71/TMRI23/TMCI23/ $\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23/ $\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23/ $\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23/ $\overline{\text{DREQ1}}$
112	122	P72/TMO0/$\overline{\text{TEND0}}$/ $\overline{\text{CS6}}$/SYNCI	P72/TMO0/$\overline{\text{TEND0}}$/ $\overline{\text{CS6}}$/SYNCI	P72/TMO0/$\overline{\text{TEND0}}$/ $\overline{\text{CS6}}$/SYNCI	P72/TMO0/$\overline{\text{TEND0}}$/ SYNCI
113	123	P73/TMO1/$\overline{\text{TEND1}}$/ $\overline{\text{CS7}}$	P73/TMO1/$\overline{\text{TEND1}}$/ $\overline{\text{CS7}}$	P73/TMO1/$\overline{\text{TEND1}}$/ $\overline{\text{CS7}}$	P73/TMO1/$\overline{\text{TEND1}}$
114	124	P74/TMO2/$\overline{\text{MRES}}$	P74/TMO2/$\overline{\text{MRES}}$	P74/TMO2/$\overline{\text{MRES}}$	P74/TMO2/$\overline{\text{MRES}}$
115	125	P75/TMO3/SCK3	P75/TMO3/SCK3	P75/TMO3/SCK3	P75/TMO3/SCK3
116	126	P76/RxD3	P76/RxD3	P76/RxD3	P76/RxD3
117	127	P77/TxD3	P77/TxD3	P77/TxD3	P77/TxD3
118	128	MD0	MD0	MD0	MD0
119	1	MD1	MD1	MD1	MD1
120	2	MD2	MD2	MD2	MD2
—	3	NC	NC	NC	NC
—	4	NC	NC	NC	NC

Notes: 1. NC pins should be connected to VSS or left open.
2. FWE is used only in the flash memory version. Leave open in the mask ROM version.

1.3.3 Pin Functions

Table 1-3 outlines the pin functions of the H8S/2633 Series.

Table 1-3 Pin Functions

Type	Symbol	I/O	Name and Function
Power	VCC	Input	Power supply: For connection to the power supply. All VCC pins should be connected to the system power supply.
	PVCC1, PVCC2	Input	Port power supply pin. Connect all pins to the same power supply.
	VSS	Input	Ground: For connection to ground (0 V). All VSS pins should be connected to the system power supply (0 V).
Clock	PLLVCC	Input	PLL power supply: Power supply for on-chip PLL oscillator.
	PLLVSS	Input	PLL ground: Ground for on-chip PLL oscillator.
	PLLCAP	Input	PLL capacitance: External capacitance pin for on-chip PLL oscillator.
	XTAL	Input	Connects to a crystal oscillator. See section 23, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	Input	Connects to a crystal oscillator. The EXTAL pin can also input an external clock. See section 23, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	OSC1	Input	Subclock: Connects to a 32.768 kHz crystal oscillator. See Chapter 23 Clock Oscillator for examples of connections to a crystal oscillator.
	OSC2	Input	Subclock: Connects to a 32.768 kHz crystal oscillator. See Chapter 23 Clock Oscillator for examples of connections to a crystal oscillator.
	ø	Output	System clock: Supplies the system clock to an external device.

Type	Symbol	I/O	Name and Function
Operating mode control	MD2 to MD0	Input	Mode pins: These pins set the operating mode. The relation between the settings of pins MD2 to MD0 and the operating mode is shown below. These pins should not be changed while the H8S/2633 Series is operating.

MD2	MD1	MD0	Operating Mode
0	0	0	—
		1	—
	1	0	—
		1	—
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

System control	$\overline{\text{RES}}$	Input	Reset input: When this pin is driven low, the chip is reset.
	$\overline{\text{MRES}}$	Input	Manual reset: When this pin is driven low, a transmission is made to manual reset mode.
	$\overline{\text{STBY}}$	Input	Standby: When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{\text{BREQ}}$	Input	Bus request: Used by an external bus master to issue a bus request to the H8S/2633 Series.
	$\overline{\text{BREQO}}$	Output	Bus request output: The external bus request signal used when an internal bus master accesses external space in the external bus-released state.
	$\overline{\text{BACK}}$	Output	Bus request acknowledge: Indicates that the bus has been released to an external bus master.
	FWE	Input	Flash write enable: Pin for flash memory use (in planning stage).

Type	Symbol	I/O	Name and Function
Interrupts	NMI	Input	Nonmaskable interrupt: Requests a nonmaskable interrupt. When this pin is not used, it should be fixed high.
	$\overline{IRQ7}$ to $\overline{IRQ0}$	Input	Interrupt request 7 to 0: These pins request a maskable interrupt.
Address bus	A23 to A0	Output	Address bus: These pins output an address.
Data bus	D15 to D0	I/O	Data bus: These pins constitute a bidirectional data bus.
Bus control	$\overline{CS7}$ to $\overline{CS0}$	Output	Chip select: Selection signal for areas 0 to 7.
	$\overline{AS}$	Output	Address strobe: When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{RD}$	Output	Read: When this pin is low, it indicates that the external address space can be read.
	$\overline{HWR}$	Output	High write/write enable/upper write enable: A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled. The 2CAS type DRAM write enable signal. The 2WE type DRAM upper write enable signal.
	$\overline{LWR}$	Output	Low write/lower column address strobe/lower write enable: A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled. The 2CAS type (LCASS = 1) DRAM lower column address strobe signal. The 2WE type DRAM lower write enable signal.
	$\overline{CAS}$	Output	Upper column address strobe/column address strobe: The 2CAS type DRAM upper column address strobe signal.
	$\overline{LCAS}$	Output	Lower column address strobe: The 2CAS type DRAM lower column address strobe signal.
	$\overline{OE}$	Output	Output enable: Output enable signal for DRAM space read access.
	$\overline{WAIT}$	Input	Wait: Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.

Type	Symbol	I/O	Name and Function
DMA controller (DMAC)	$\overline{\text{DREQ1}}$, $\overline{\text{DREQ0}}$	Input	DMA request 1,0: Requests DMAC activation.
	$\overline{\text{TEND1}}$, $\overline{\text{TEND0}}$	Output	DMA transfer completed 1,0: Indicates DMAC data transfer end.
	$\overline{\text{DACK1}}$, $\overline{\text{DACK0}}$	Output	DMA transfer acknowledge 1,0: DMAC single address transfer acknowledge pin.
16-bit timer-pulse unit (TPU)	TCLKD to TCLKA	Input	Clock input D to A: These pins input an external clock.
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	I/O	Input capture/ output compare match A0 to D0: The TGR0A to TGR0D input capture input or output compare output, or PWM output pins.
	TIOCA1, TIOCB1	I/O	Input capture/ output compare match A1 and B1: The TGR1A and TGR1B input capture input or output compare output, or PWM output pins.
	TIOCA2, TIOCB2	I/O	Input capture/ output compare match A2 and B2: The TGR2A and TGR2B input capture input or output compare output, or PWM output pins.
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	I/O	Input capture/ output compare match A3 to D3: The TGR3A to TGR3D input capture input or output compare output, or PWM output pins.
	TIOCA4, TIOCB4	I/O	Input capture/output compare match A4 and B4: The TGR4A and TGR4B input capture input or output compare output, or PWM output pins.
	TIOCA5, TIOCB5	I/O	Input capture/output compare match A5 and B5: The TGR5A and TGR5B input capture input or output compare output, or PWM output pins.
Programmable pulse generator (PPG)	PO15 to PO8	Output	Pulse output 15 to 8: Pulse output pins.
8-bit timer	TMO0 to TMO3	Output	Compare match output: The compare match output pins.
	TMCI01, TMCI23	Input	Counter external clock input: Input pins for the external clock input to the counter.
	TMRI01, TMRI23	Input	Counter external reset input: The counter reset input pins.

Type	Symbol	I/O	Name and Function
14-bit PWM timer (PWMX)	PWM0 to PWM3	Output	PWMX timer output: PWM D/A pulse output pins.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	Output	Watchdog timer overflows: The counter overflows signal output pin in watchdog timer mode.
	BUZZ	Output	BUZZ output: Output pins for the pulse divided by the watchdog timer.
Serial communication interface (SCI)/ Smart Card interface	TxD4, TxD3, TxD2, TxD1, TxD0	Output	Transmit data (channel 0, 1, 2): Data output pins.
	RxD4, RxD3, RxD2, RxD1, RxD0	Input	Receive data (channel 0, 1, 2): Data input pins.
	SCK4, SCK3, SCK2, SCK1 SCK0	I/O	Serial clock (channel 0, 1, 2): Clock I/O pins. SCK0 output type is NMOS push-pull.
IrDA-equipped SCI 1 channel (SCI0)	IrTxD IrRxD	Output/ Input	IrDA transmission data/receive data: Input/output pins for the data encoded for the IrDA.
I^2C bus interface (IIC) (optional)	SCL0 SCL1	I/O	I^2C clock input (channel 1, 0): I^2C clock input/output pins. These functions have a bus driving function. SCL0's output format is an NMOS open drain.
	SDA0 SDA1	I/O	I^2C data input/output (channel 1, 0): I^2C clock input/output pins. These functions have a bus driving function. SCL0's output format is an NMOS open drain.
A/D converter	AN15 to AN0	Input	Analog 15 to 0: Analog input pins.
	$\overline{\text{ADTRG}}$	Input	A/D conversion external trigger input: Pin for input of an external trigger to start A/D conversion.
D/A converter	DA3 to DA0	Output	Analog output: Analog output pins for D/A converter.
A/D converter, D/A converter	AVCC	Input	A/D converter and D/A converter power supply pin When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V).

Type	Symbol	I/O	Name and Function
A/D converter, D/A converter	AVSS	Input	Analog circuit ground and reference voltage A/D converter and D/A converter ground and reference voltage. Connect to system power supply (0 V).
	Vref	Input	A/D converter and D/A converter reference voltage input pin When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V).
I/O ports	P17 to P10	I/O	Port 1: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 1 data direction register (P1DDR).
	P37 to P30	I/O	Port 3: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 3 data direction register (P3DDR).
	P47 to P40	Input	Port 4: An 8-bit input port.
	P77 to P70	I/O	Port 7: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 7 data direction register (P7DDR).
	P97 to P90	Input	Port 9: An 8-bit input port.
	PA3 to PA0	I/O	Port A: A 4-bit I/O port. Input or output can be designated for each bit by means of the port A data direction register (PADDR).
	PB7 to PB0	I/O	Port B: An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PBDDR).
	PC7 to PC0	I/O	Port C: An 8-bit I/O port. Input or output can be designated for each bit by means of the port C data direction register (PCDDR).
	PD7 to PD0	I/O	Port D: An 8-bit I/O port. Input or output can be designated for each bit by means of the port D data direction register (PDDDR).
	PE7 to PE0	I/O	Port E: An 8-bit I/O port. Input or output can be designated for each bit by means of the port E data direction register (PEDDR).
	PF7 to PF0	I/O	Port F: An 8-bit I/O port. Input or output can be designated for each bit by means of the port F data direction register (PFDDR).
	PG4 to PG0	I/O	Port G: An 5-bit I/O port. Input or output can be designated for each bit by means of the port G data direction register (PGDDR).

Section 2 CPU

2.1 Overview

The H8S/2600 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

2.1.1 Features

The H8S/2600 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H object programs
- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

- High-speed operation
  - — All frequently-used instructions execute in one or two states
  - — Maximum clock rate : 25 MHz
  - — 8/16/32-bit register-register add/subtract : 40 ns
  - — 8 × 8-bit register-register multiply : 120 ns
  - — 16 ÷ 8-bit register-register divide : 480 ns
  - — 16 × 16-bit register-register multiply : 160 ns
  - — 32 ÷ 16-bit register-register divide : 800 ns

- Two CPU operating modes
  - — Normal mode*
  - — Advanced mode

Note: * Not available in the H8S/2633 Series.

- Power-down state
  - — Transition to power-down state by SLEEP instruction
  - — CPU clock speed selection

2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration

  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- Number of execution states

  The number of execution states of the MULXU and MULXS instructions is different in each CPU.

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.3 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements.

- More general registers and control registers
  - — Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - — Normal mode* supports the same 64-kbyte address space as the H8/300 CPU.
  - — Advanced mode supports a maximum 16-Mbyte address space.

Note: * Not available in the H8S/2633 Series.

- Enhanced addressing
  - — The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — Signed multiply and divide instructions have been added.
  - — A multiply-and-accumulate instruction has been added.
  - — Two-bit shift instructions have been added.
  - — Instructions for saving and restoring multiple registers have been added.
  - — A test and set instruction has been added.
- Higher speed
  - — Basic instructions execute twice as fast.

2.1.4 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements.

- Additional control register
  - — One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — A multiply-and-accumulate instruction has been added.

— Two-bit shift instructions have been added.
— Instructions for saving and restoring multiple registers have been added.
— A test and set instruction has been added.

- Higher speed
  — Basic instructions execute twice as fast.

2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode* supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes for program and data areas combined). The mode is selected by the mode pins of the microcontroller.

Note: * Not available in the H8S/2633 Series.

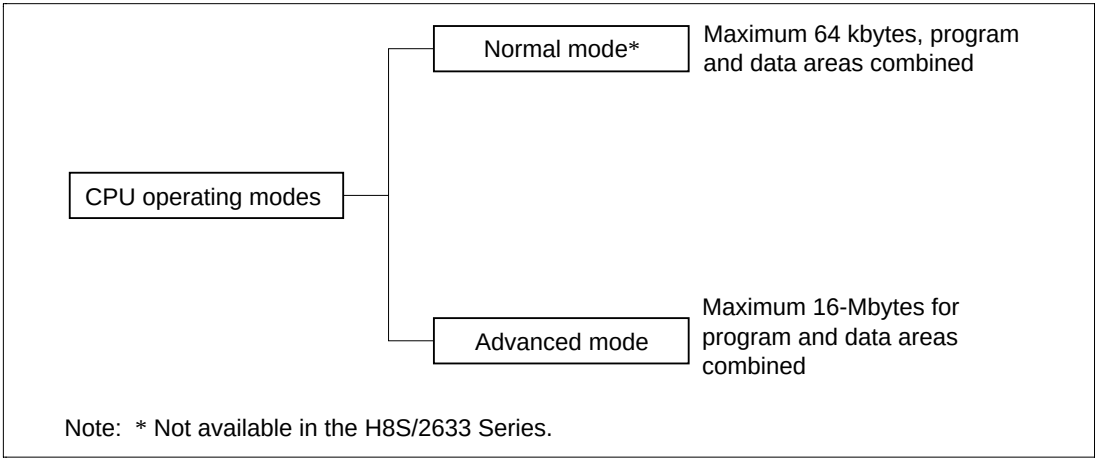


Figure 2-1 CPU Operating Modes

(1) Normal Mode (Not Available in the H8S/2633 Series)

The exception vector table and stack have the same structure as in the H8/300 CPU.

Address Space: A maximum address space of 64 kbytes can be accessed.

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.

Instruction Set: All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

Exception Vector Table and Memory Indirect Branch Addresses: In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits (figure 2-2). The exception vector table differs depending on the microcontroller. For details of the exception vector table, see section 4, Exception Handling.

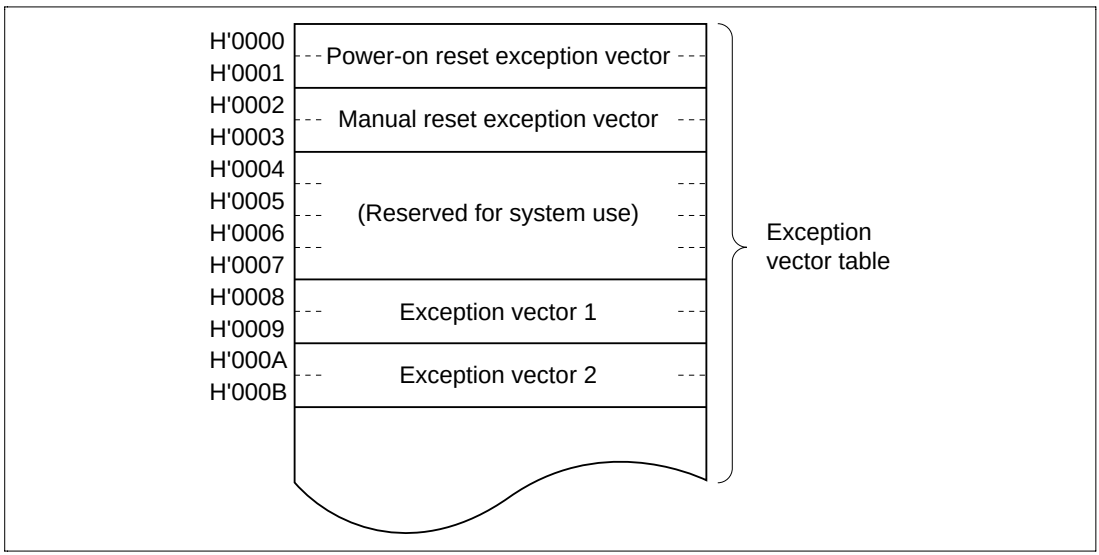
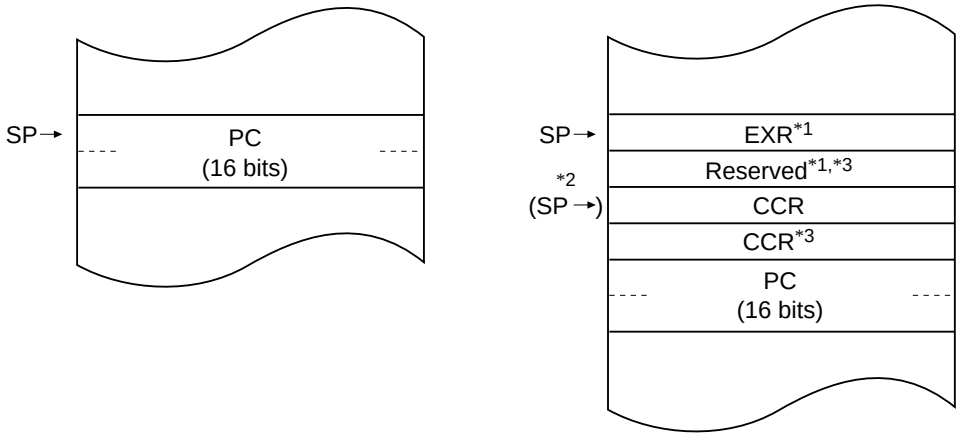


Figure 2-2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.

Stack Structure: When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2-3. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-3 Stack Structure in Normal Mode

(2) Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (architecturally a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximum of 4 Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.

Instruction Set: All instructions and addressing modes can be used.

Exception Vector Table and Memory Indirect Branch Addresses: In advanced mode the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2-4). For details of the exception vector table, see section 4, Exception Handling.

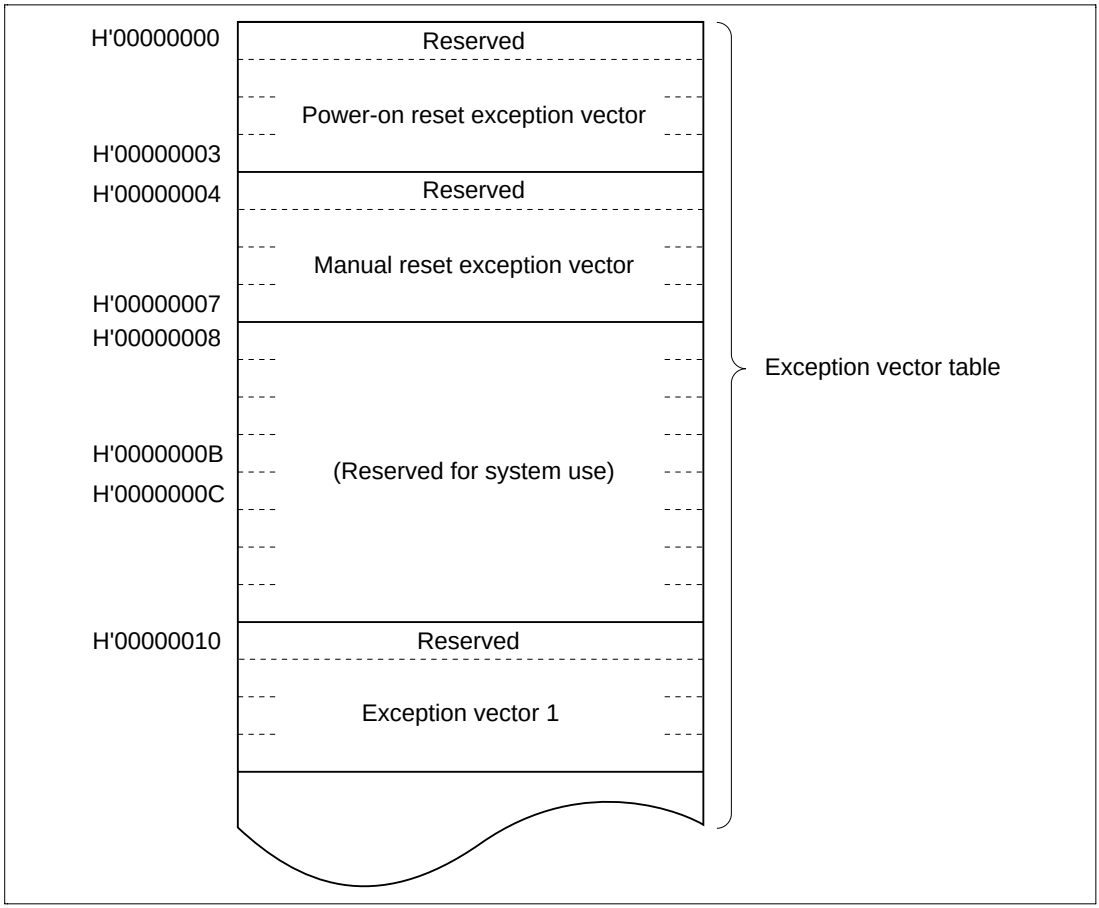
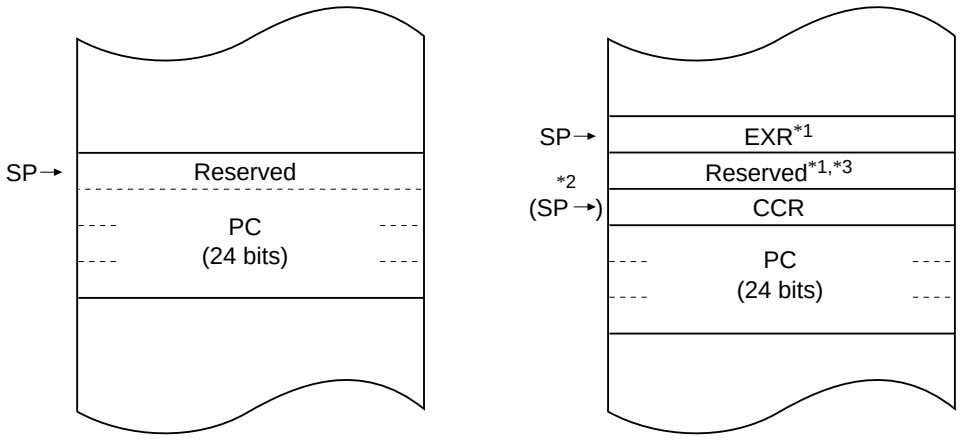


Figure 2-4 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also the exception vector table.

Stack Structure: In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2-5. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-5 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2-6 shows a memory map of the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode.

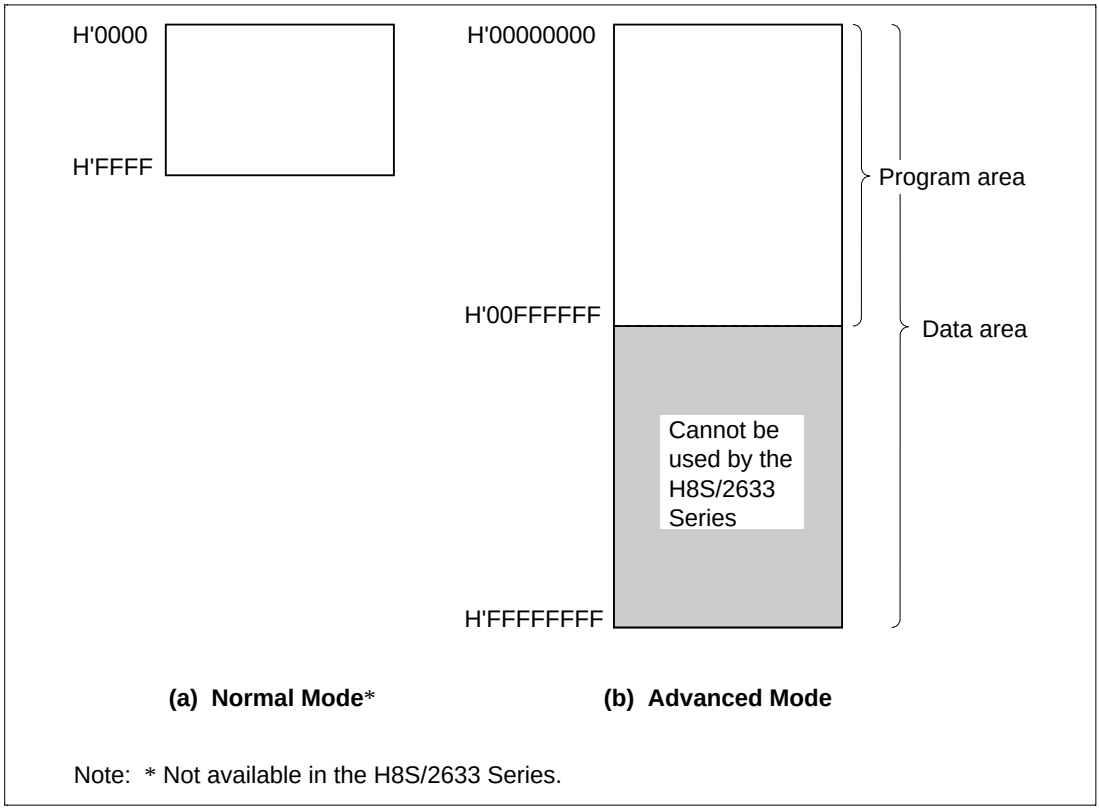


Figure 2-6 Memory Map

2.4 Register Configuration

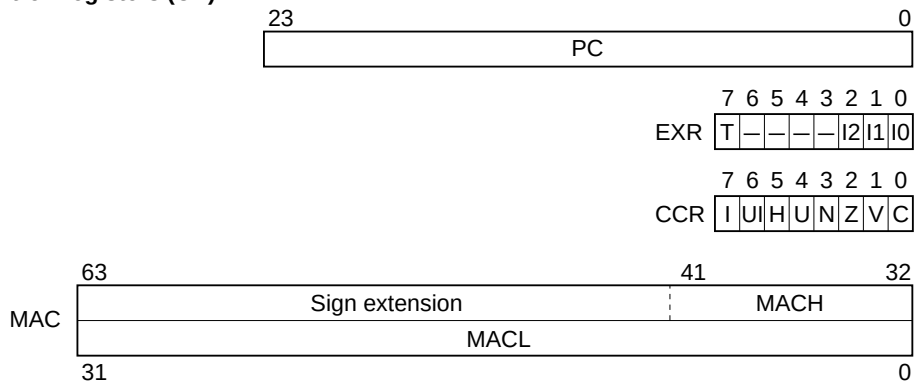
2.4.1 Overview

The CPU has the internal registers shown in figure 2-7. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

	15 — 07	07 — 0	
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



Legend

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit	MAC:	Multiply-accumulate register
UI:	User bit or interrupt mask bit*		

Note: * Cannot be used as an interrupt mask bit in the H8S/2633 Series.

Figure 2-7 CPU Registers

2.4.2 General Registers

The CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

Figure 2-8 illustrates the usage of the general registers. The usage of each register can be selected independently.

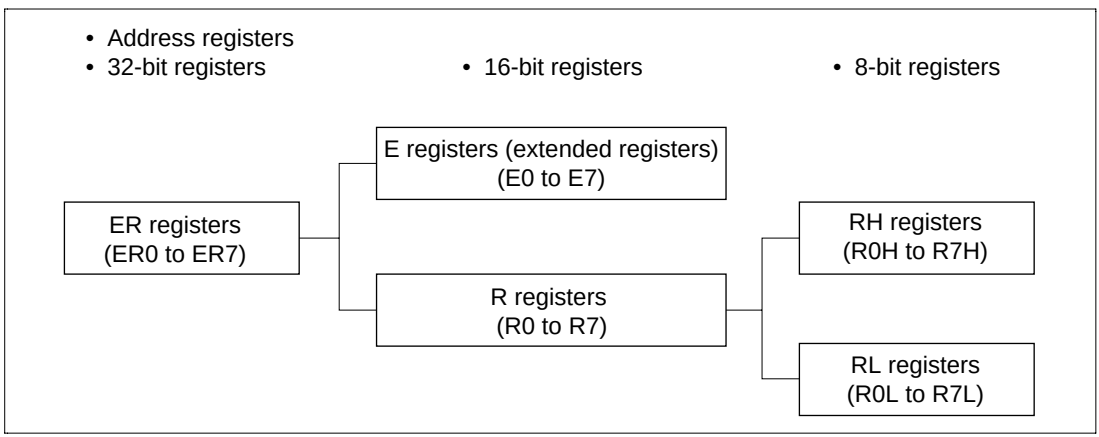


Figure 2-8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2-9 shows the stack.

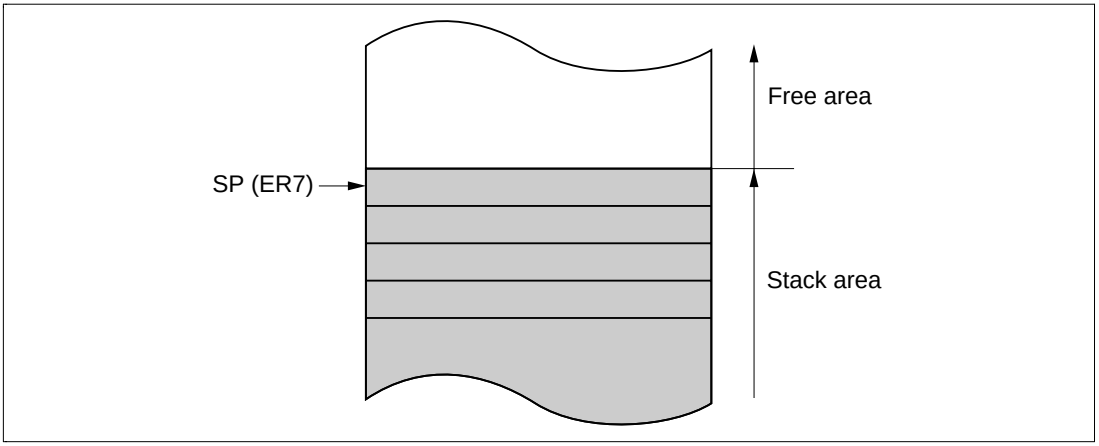


Figure 2-9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control register (EXR), 8-bit condition-code register (CCR), and 64-bit multiply-accumulate register (MAC).

(1) Program Counter (PC): This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0.)

(2) Extended Control Register (EXR): This 8-bit register contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are executed in sequence. When this bit is set to 1, a trace exception is generated each time an instruction is executed.

Bits 6 to 3—Reserved: They are always read as 1.

Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions. All interrupts, including NMI, are disabled for three states after one of these instructions is executed, except for STC.

(3) Condition-Code Register (CCR): This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI is accepted regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details, refer to section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of data.

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave some or all of the flag bits unchanged. For the action of each instruction on the flag bits, refer to Appendix A.1, List of Instructions.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

(4) Multiply-Accumulate Register (MAC): This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

2.4.4 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (ER7) is not initialized. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figure 2-10 shows the data formats in general registers.

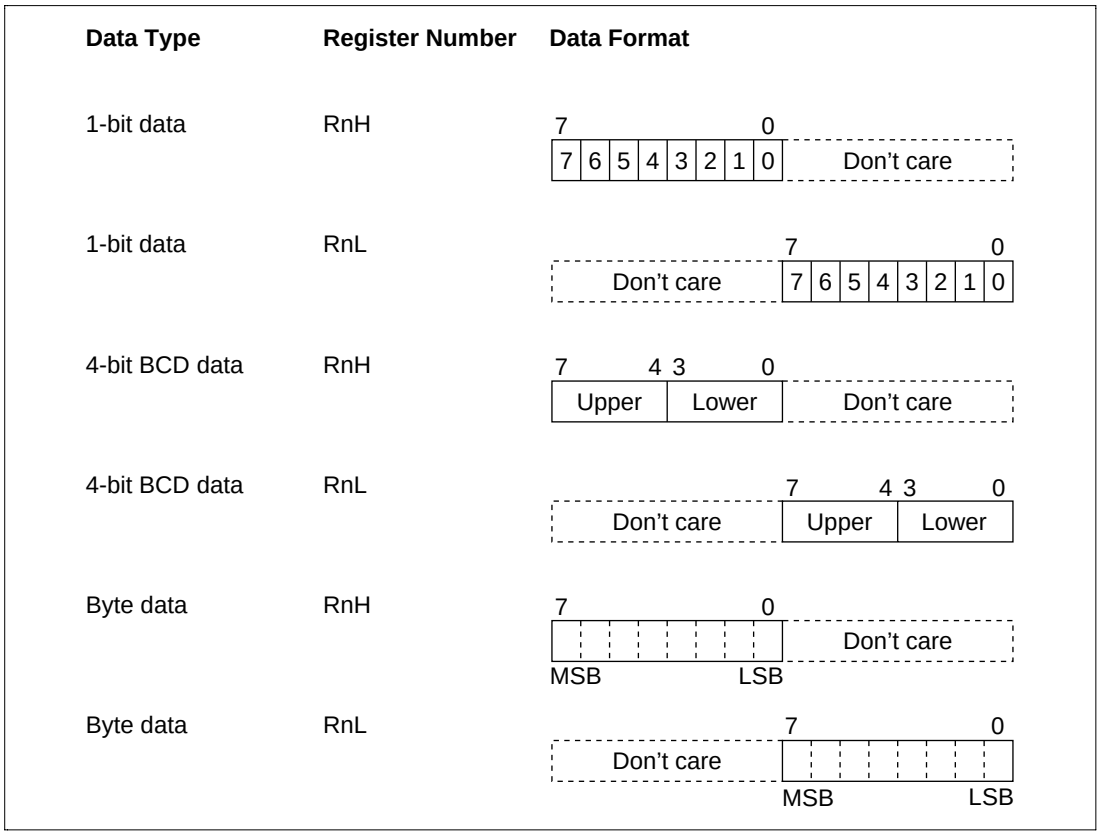
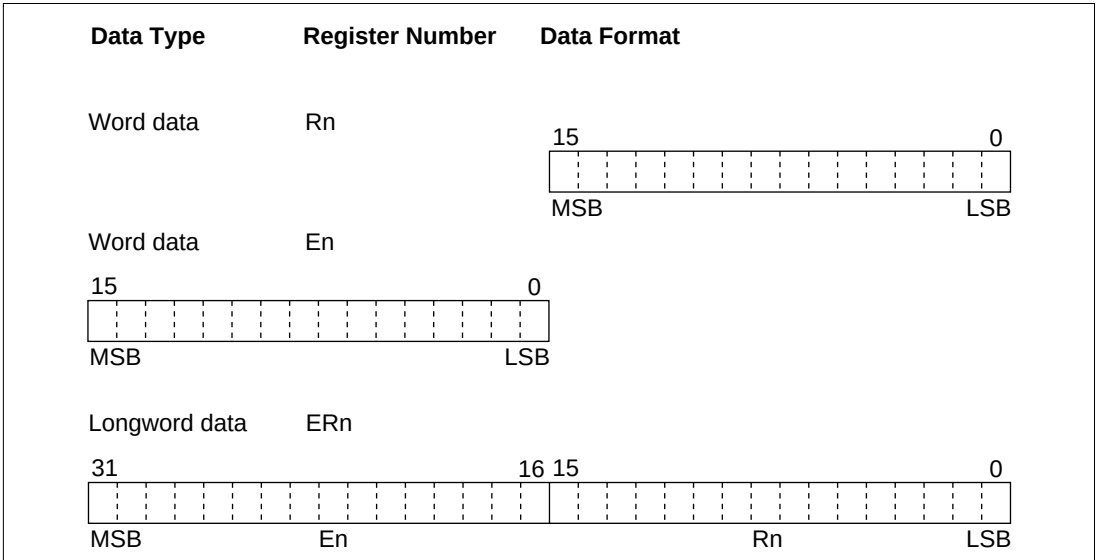


Figure 2-10 General Register Data Formats



Legend

ERn: General register ER
En: General register E
Rn: General register R
RnH: General register RH
RnL: General register RL
MSB: Most significant bit
LSB: Least significant bit

Figure 2-10 General Register Data Formats (cont)

2.5.2 Memory Data Formats

Figure 2-11 shows the data formats in memory. The CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

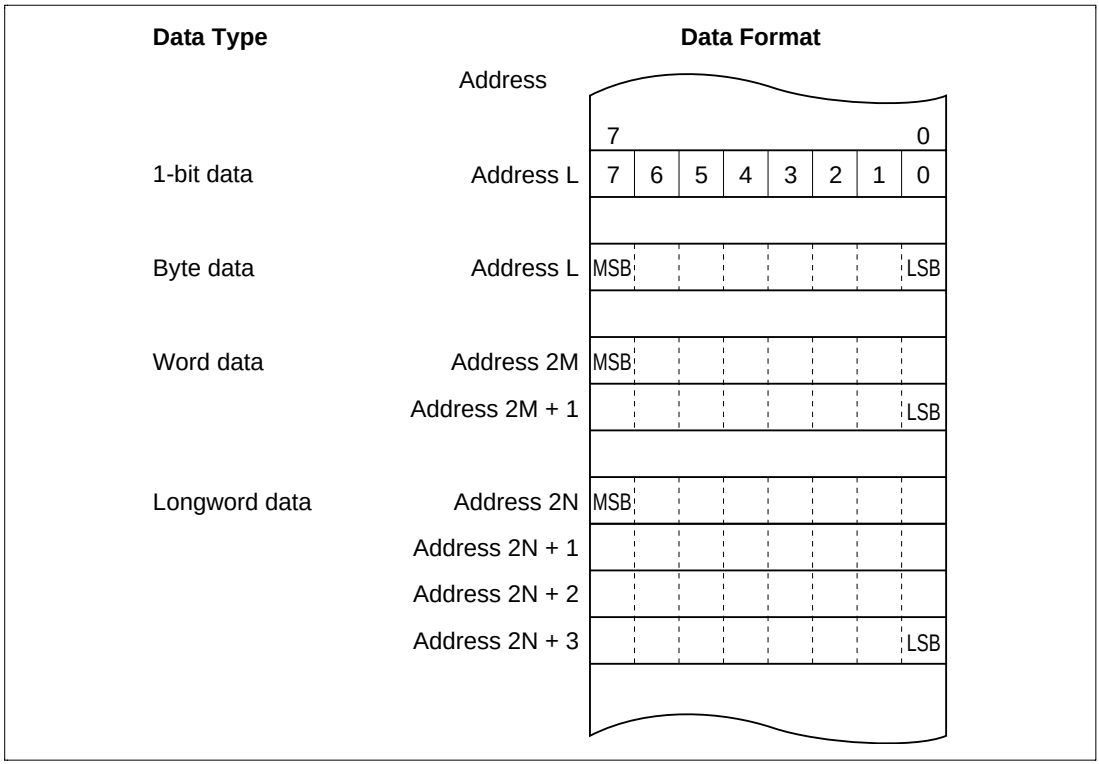


Figure 2-11 Memory Data Formats

When ER7 is used as an address register to access the stack, the operand size should be word size or longword size.

2.6 Instruction Set

2.6.1 Overview

The H8S/2600 CPU has 69 types of instructions. The instructions are classified by function in table 2-1.

Table 2-1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	BWL	5
	POP*[1], PUSH*[1]	WL	
	LDM, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	BWL	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	BWL	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	BW	
	EXTU, EXTS	WL	
	TAS*[4]	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	BWL	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1

Notes: B-byte size; W-word size; L-longword size.

*1 POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.

*2 Bcc is the general name for conditional branch instructions.

*3 Not available in the H8S/2633 Series.

*4 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

2.6.2 Instructions and Addressing Modes

Table 2-2 indicates the combinations of instructions and addressing modes that the H8S/2600 CPU can use.

Table 2-2 Combinations of Instructions and Addressing Modes

Function	Instruction	Addressing Modes															
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	—	WL
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—	—	L
	MOVEPE*1, MOVTPE*1	—	—	—	—	—	—	—	B	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—	—
	MULXU, DIVXU	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	MULXS, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—	—
	TAS*2	—	—	B	—	—	—	—	—	—	—	—	—	—	—
	MAC	—	—	—	—	—	○	—	—	—	—	—	—	—	—
	CLRMAC	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDMAC, STMAC	—	L	—	—	—	—	—	—	—	—	—	—	—	—

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	○	○	—	—
	JMP, JSR	—	—	—	—	—	—	—	—	○	—	—	—	○	—
	RTS	—	—	—	—	—	—	—	—	—	—	—	—	—	○
System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	—	W	—	—	—	—
	STC	—	B	W	W	W	W	—	W	—	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend

B: Byte

W: Word

L: Longword

Notes: *1 Not available in the H8S/2633 Series.

*2 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

2.6.3 Table of Instructions Classified by Function

Table 2-3 summarizes the instructions in each functional category. The notation used in table 2-3 is defined below.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Table 2-3 Instructions Classified by Function

Type	Instruction	Size*[1]	Function
Data transfer	MOV	B/W/L	(EAs) → Rd, Rs → (Ead) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
	MOVFPE	B	Cannot be used in the H8S/2633 Series.
	MOVTPE	B	Cannot be used in the H8S/2633 Series.
	POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
	PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
	LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
	STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.
Arithmetic operations	ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
	ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or on immediate data and data in a general register.
	INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
	ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
	DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.

Type	Instruction	Size*[1]	Function
Arithmetic operations	MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
	NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
	EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
	EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
	TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @Erd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
	MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating

Type	Instruction	Size*[1]	Function
Arithmetic operations	CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
	LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.
Logic operations	AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
	OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
	XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general register contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotation is possible.
Bit-manipulation instructions	BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
	BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
	BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.
Branch instructions	Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA(BT)	Always (true)	Always
BRN(BF)	Never (false)	Never
BHI	High	C ∨ Z = 0
BLS	Low or same	C ∨ Z = 1
BCC(BHS)	Carry clear (high or same)	C = 0
BCS(BLO)	Carry set (low)	C = 1
BNE	Not equal	Z = 0
BEQ	Equal	Z = 1
BVC	Overflow clear	V = 0
BVS	Overflow set	V = 1
BPL	Plus	N = 0
BMI	Minus	N = 1
BGE	Greater or equal	N ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z∨(N ⊕ V) = 0
BLE	Less or equal	Z∨(N ⊕ V) = 1

	JMP	—	Branches unconditionally to a specified address.
	BSR	—	Branches to a subroutine at a specified address.
	JSR	—	Branches to a subroutine at a specified address.
	RTS	—	Returns from a subroutine
System control instructions	TRAPA	—	Starts trap-instruction exception handling.
	RTE	—	Returns from an exception-handling routine.
	SLEEP	—	Causes a transition to a power-down state.

Type	Instruction	Size*[1]	Function
System control instructions	LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
	ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
	XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
	NOP	—	PC + 2 → PC Only increments the program counter.
Block data transfer instruction	EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
	EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next; Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6. R4L or R4: size of block (bytes) ER5: starting source address ER6: starting destination address Execution of the next instruction begins as soon as the transfer is completed.

Notes: *1 Size refers to the operand size.
B: Byte
W: Word
L: Longword
*2 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

2.6.4 Basic Instruction Formats

The H8S/2633 Series instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

(1) Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.

(2) Register Field: Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

(3) Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.

(4) Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2-12 shows examples of instruction formats.

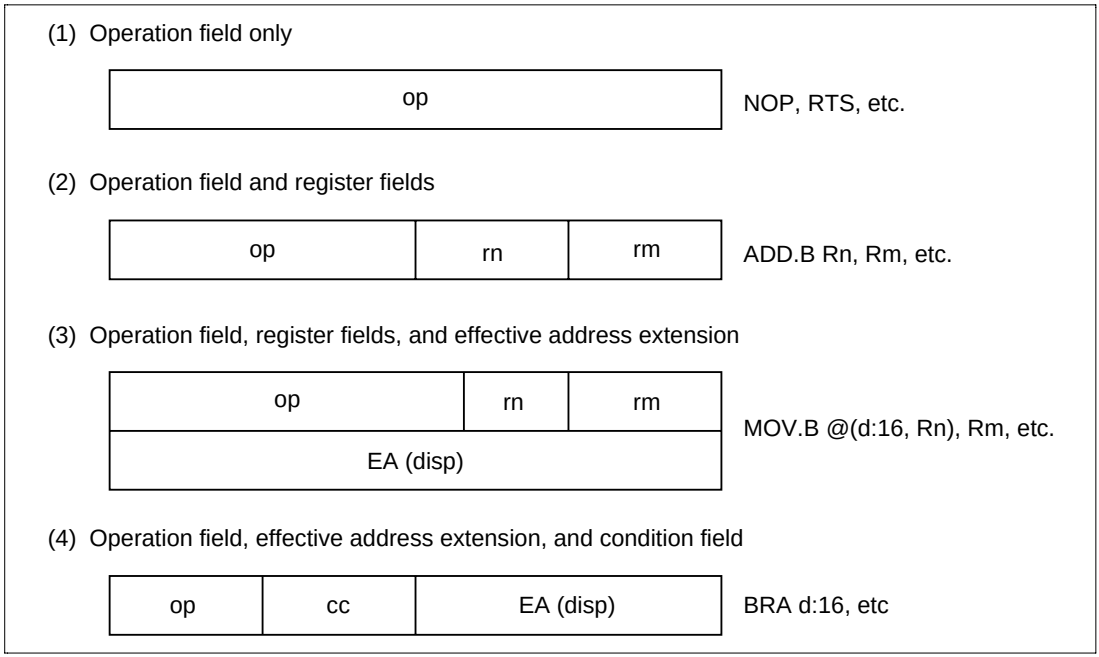


Figure 2-12 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2-4. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2-4 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

(1) Register Direct—Rn: The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

(2) Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn): A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

(4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn:

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

- Register indirect with pre-decrement—@-ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32).

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

Table 2-5 indicates the accessible absolute address ranges.

Table 2-5 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: * Not available in the H8S/2633 Series.

(6) Immediate—#xx:8, #xx:16, or #xx:32: The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC): This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode). In normal mode* the memory operand is a word operand and the branch address is 16 bits long. In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

Note: * Not available in the H8S/2633 Series.

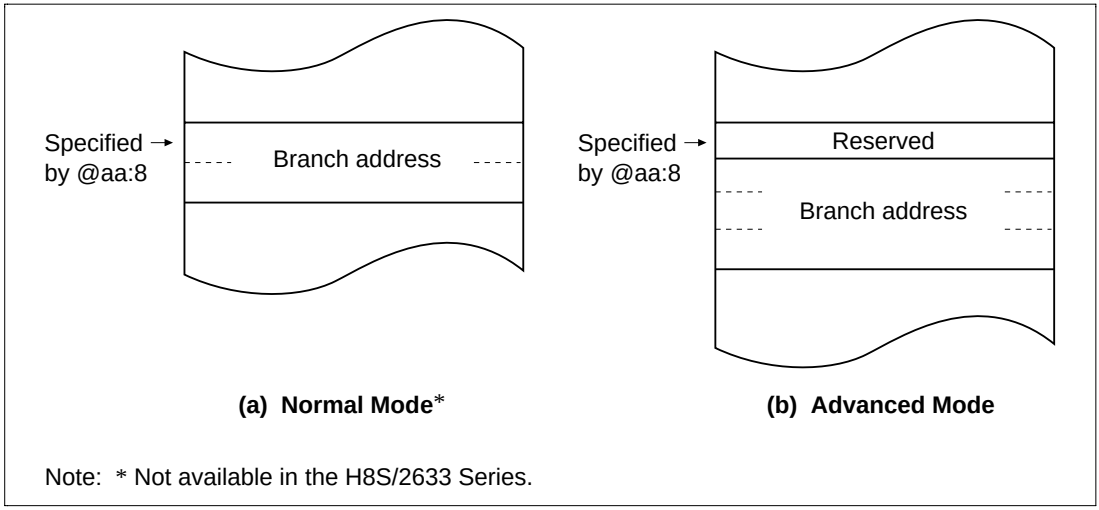


Figure 2-13 Branch Address Specification in Memory Indirect Mode

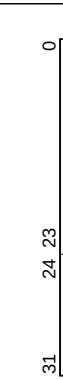
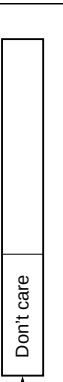
If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

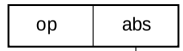
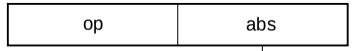
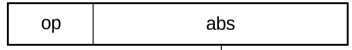
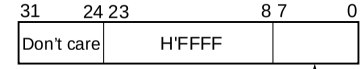
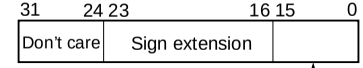
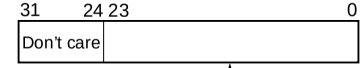
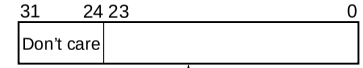
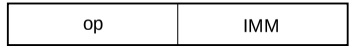
2.7.2 Effective Address Calculation

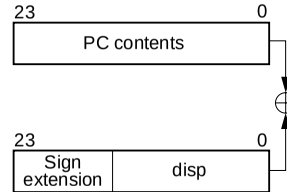
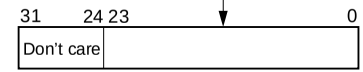
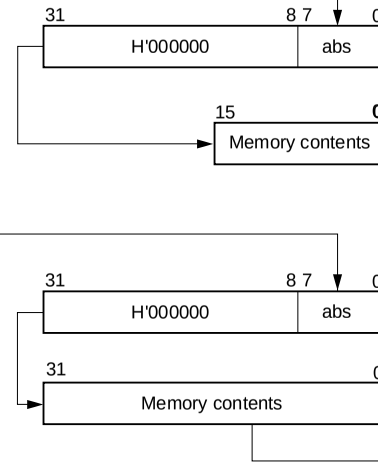
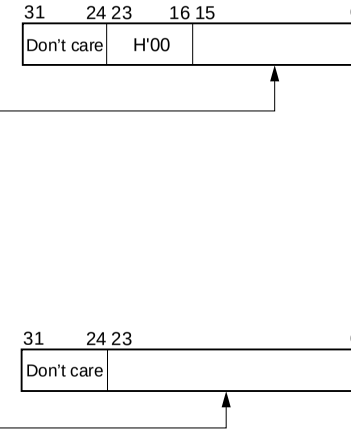
Table 2-6 indicates how effective addresses are calculated in each addressing mode. In normal mode* the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Note: * Not available in the H8S/2633 Series.

Table 2-6 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) op | rm | rm		Operand is general register contents.
2	Register indirect (@ERn) op | r		
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn) op | r | disp		
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+ op | r • Register indirect with pre-decrement @–ERn op | r		

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32 		   
6	Immediate #xx:8/#xx:16/#xx:32  		Operand is immediate data.

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
7	Program-counter relative @(d:8, PC)/@(d:16, PC)		
8	Memory indirect @@aa:8 • Normal mode* • Advanced mode		

Note: * Not available in the H8S/2633 Series.

2.8 Processing States

2.8.1 Overview

The CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2-14 shows a diagram of the processing states. Figure 2-15 indicates the state transitions.

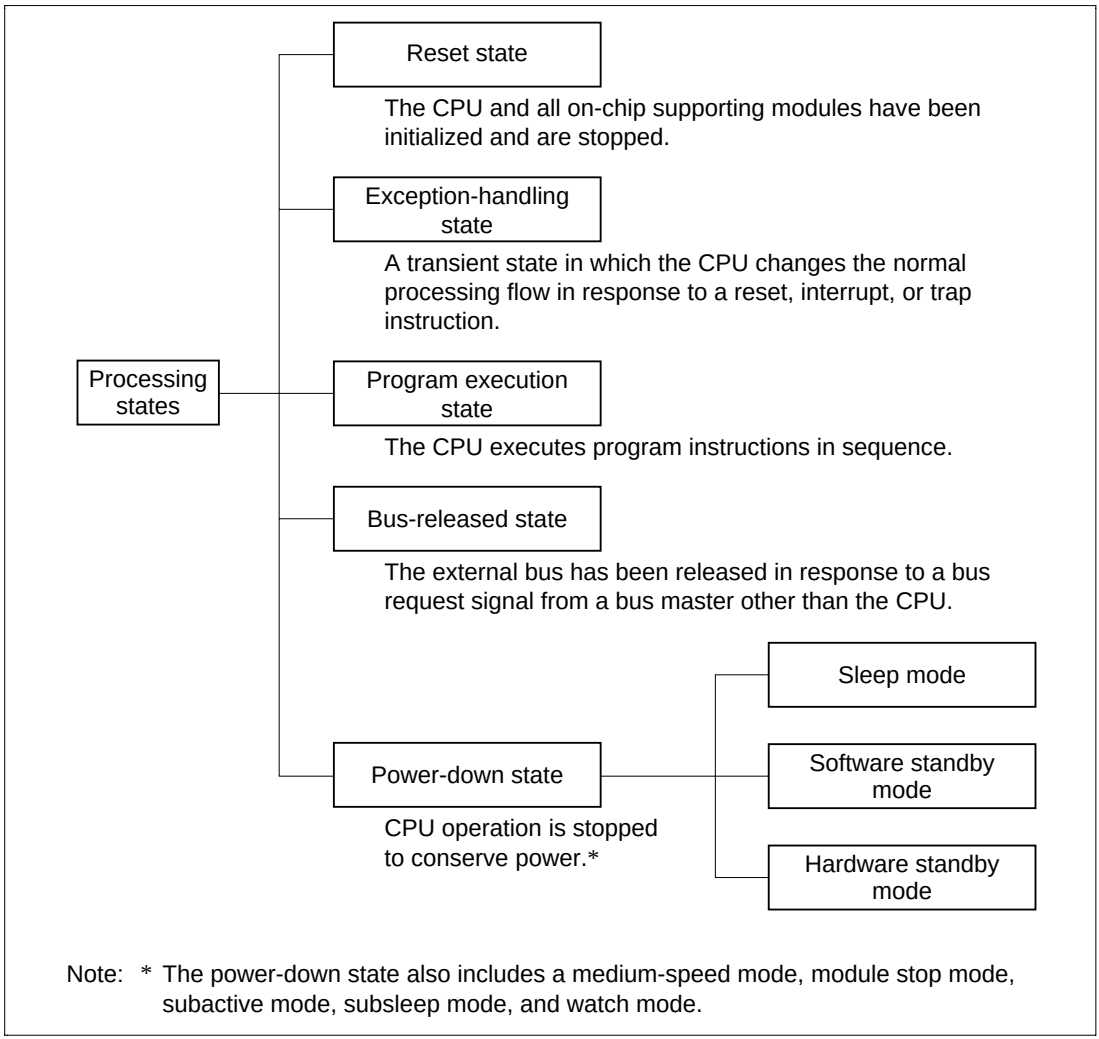
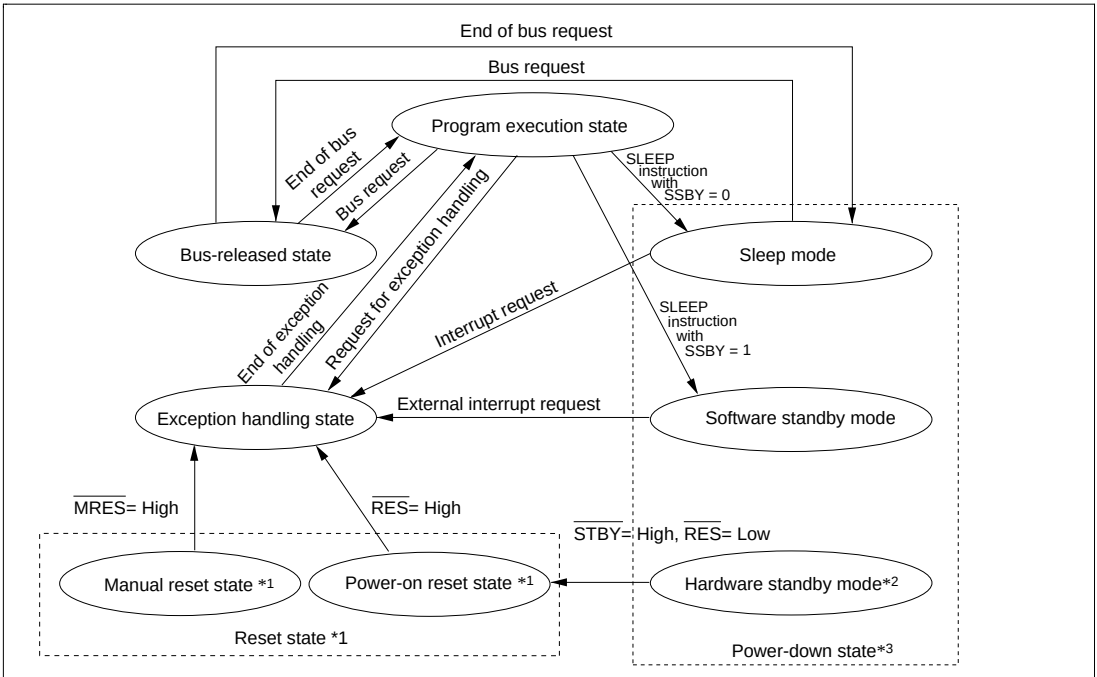


Figure 2-14 Processing States



Notes: *1 From any state except hardware standby mode, a transition to the power-on reset state occurs whenever $\overline{RES}$ goes low. From any state except hardware standby mode and power-on reset mode, a transition to the manual reset state occurs whenever $\overline{MRES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.

*2 From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

*3 Apart from these states, there are also the watch mode, subactive mode, and the subsleep mode. See Chapter 24 Power-Down States

Figure 2-15 State Transitions

2.8.2 Reset State

The CPU enters the reset state when the $\overline{RES}$ pin goes low, or when the $\overline{MRES}$ pin goes low while manual resets are enabled by the MRESE bit. In the reset state, currently executing processing is halted and all interrupts are disabled.

For details of MRESE bit setting, see section 3.2.2, System Control Register (SYSCR).

Reset exception handling starts when the $\overline{RES}$ or $\overline{MRES}$ pin* changes from low to high.

The reset state can also be entered in the event of watchdog timer overflow. For details see section 15, Watchdog Timer.

Note: * $\overline{MRES}$ pin in the case of a manual reset.

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to a reset, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address.

(1) Types of Exception Handling and Their Priority

Exception handling is performed for traces, resets, interrupts, and trap instructions. Table 2-7 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted, in the program execution state.

Exception handling and the stack structure depend on the interrupt control mode set in SYSCR.

Table 2-7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ↑	Reset	Synchronized with clock	Exception handling starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bit is set to 1, the trace starts at the end of the current instruction or current exception-handling sequence
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed*3
Low ↓			

Notes: *1 Traces are enabled only in interrupt control mode 2. Trace exception-handling is not executed at the end of the RTE instruction.

*2 Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.

*3 Trap instruction exception handling is always accepted, in the program execution state.

(2) Reset Exception Handling

After the $\overline{\text{RES}}$ pin has gone low and the reset state has been entered, when $\overline{\text{RES}}$ pin goes high again, reset exception handling starts. After the reset state has been entered by driving the $\overline{\text{MRES}}$ pin low while manual resets are enabled by the MRESE bit, reset exception handling starts when $\overline{\text{MRES}}$ pin is driven high again. The CPU enters the power-on reset state when the $\overline{\text{RES}}$ pin is low, and enters the manual reset state when the $\overline{\text{MRES}}$ pin is low. When reset exception handling starts the CPU fetches a start address (vector) from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

(3) Traces

Traces are enabled only in interrupt control mode 2. Trace mode is entered when the T bit of EXR is set to 1. When trace mode is established, trace exception handling starts at the end of each instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 and trace mode is cleared. Interrupt masks are not affected.

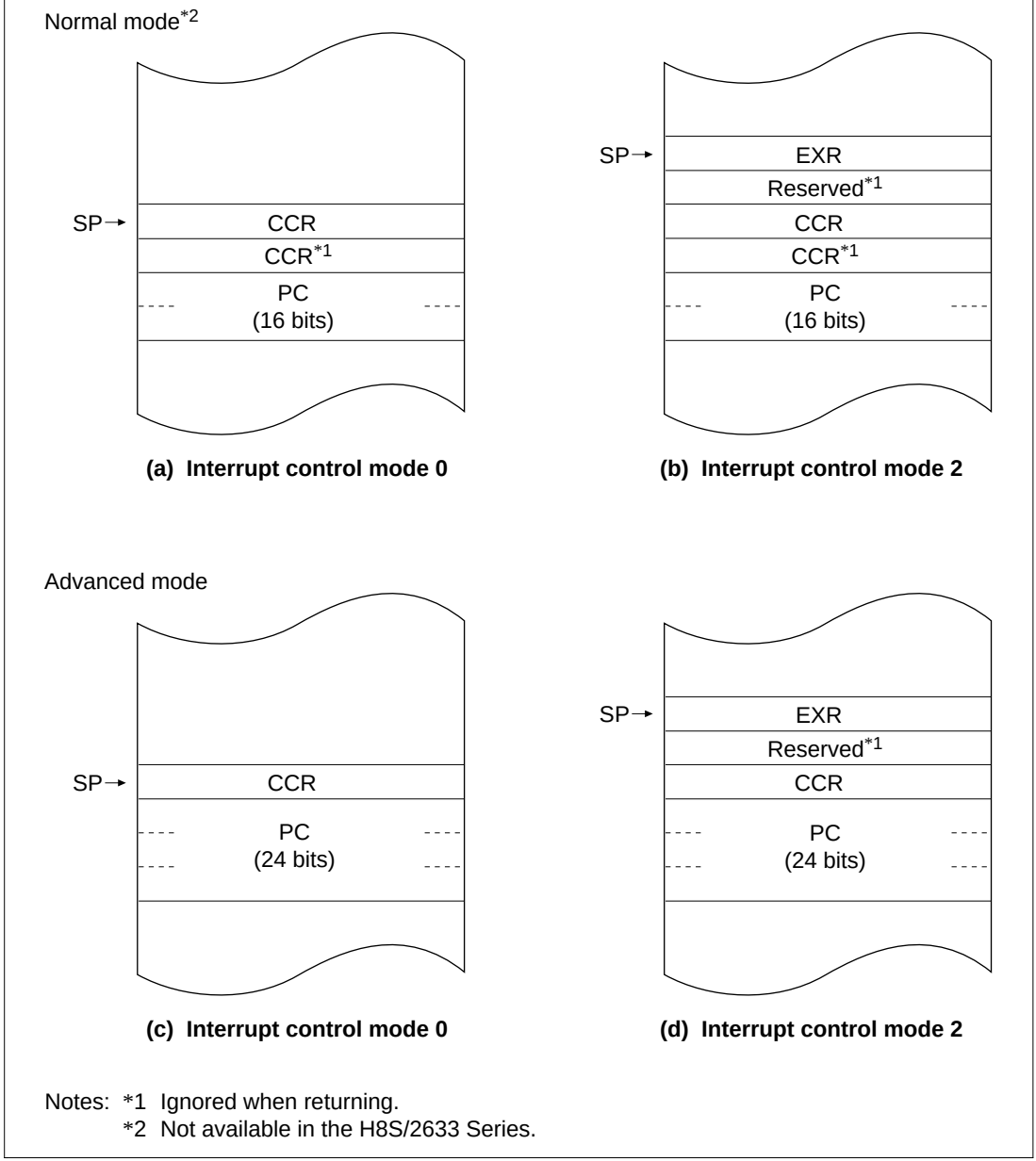
The T bit saved on the stack retains its value of 1, and when the RTE instruction is executed to return from the trace exception-handling routine, trace mode is entered again. Trace exception-handling is not executed at the end of the RTE instruction.

Trace mode is not entered in interrupt control mode 0, regardless of the state of the T bit.

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU references the stack pointer (ER7) and pushes the program counter and other control registers onto the stack. Next, the CPU alters the settings of the interrupt mask bits in the control registers. Then the CPU fetches a start address (vector) from the exception vector table and program execution starts from that start address.

Figure 2-16 shows the stack after exception handling ends.



Notes: *1 Ignored when returning.
*2 Not available in the H8S/2633 Series.

Figure 2-16 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.5 Bus-Released State

This is a state in which the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

Bus masters other than the CPU are DMA controller (DMAC) and data transfer controller (DTC).

For further details, refer to section 7, Bus Controller.

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are five modes in which the CPU stops operating: sleep mode, software standby mode, hardware standby mode, subsleep mode, and watch mode. There are also three other power-down modes: medium-speed mode, module stop mode, and subactive mode. In medium-speed mode the CPU and other bus masters operate on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. Subactive mode, subsleep mode, and watch mode are power-down states using subclock input. For details, refer to section 24, Power-Down State.

(1) Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the software standby bit (SSBY) in the standby control register (SBYCR) is cleared to 0. In sleep mode, CPU operations stop immediately after execution of the SLEEP instruction. The contents of CPU registers are retained.

(2) Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1. In software standby mode, the CPU and clock halt and all MCU operations stop. As long as a specified voltage is supplied, the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

(3) Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardware standby mode, the CPU and clock halt and all MCU operations stop. The on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

2.9 Basic Timing

2.9.1 Overview

The H8S/2600 CPU is driven by a system clock, denoted by the symbol ø. The period from one rising edge of ø to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip supporting modules, and the external address space.

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 2-17 shows the on-chip memory access cycle. Figure 2-18 shows the pin states.

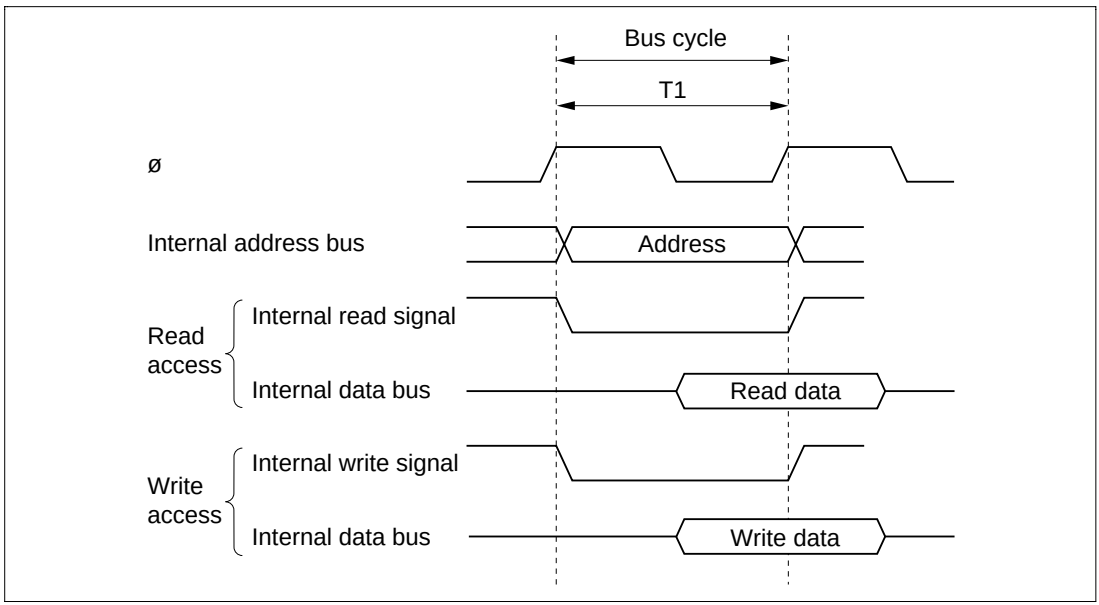


Figure 2-17 On-Chip Memory Access Cycle

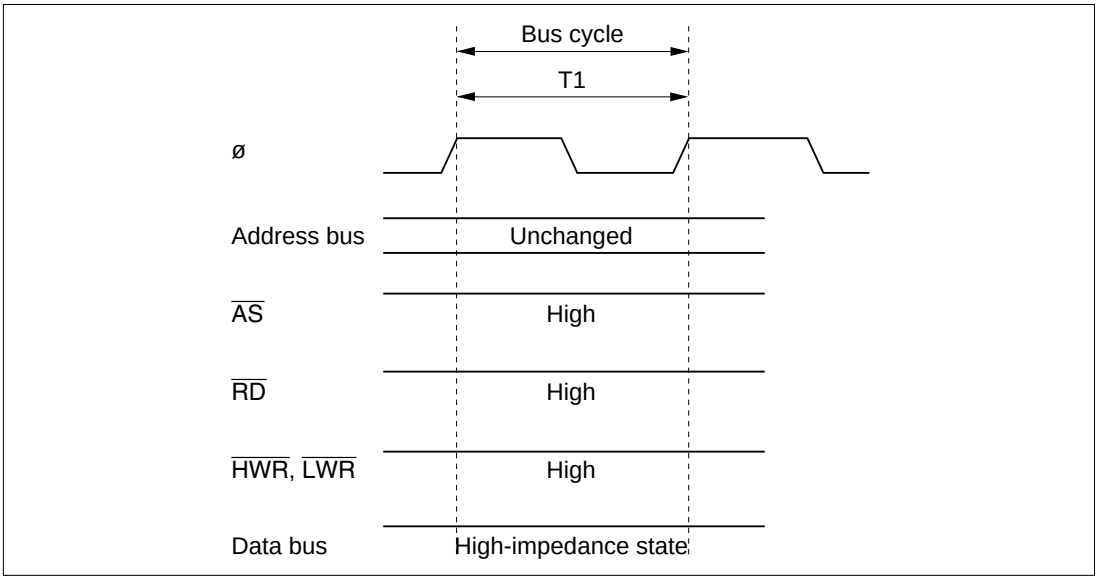


Figure 2-18 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 2-19 shows the access timing for the on-chip supporting modules. Figure 2-20 shows the pin states.

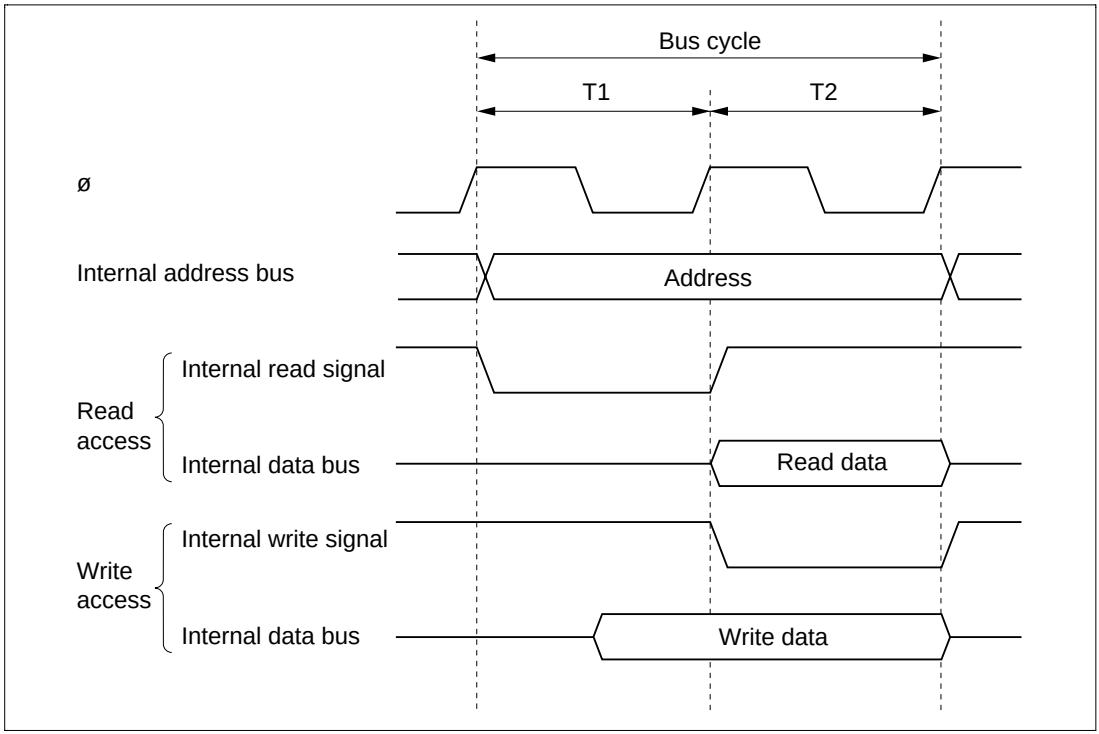


Figure 2-19 On-Chip Supporting Module Access Cycle

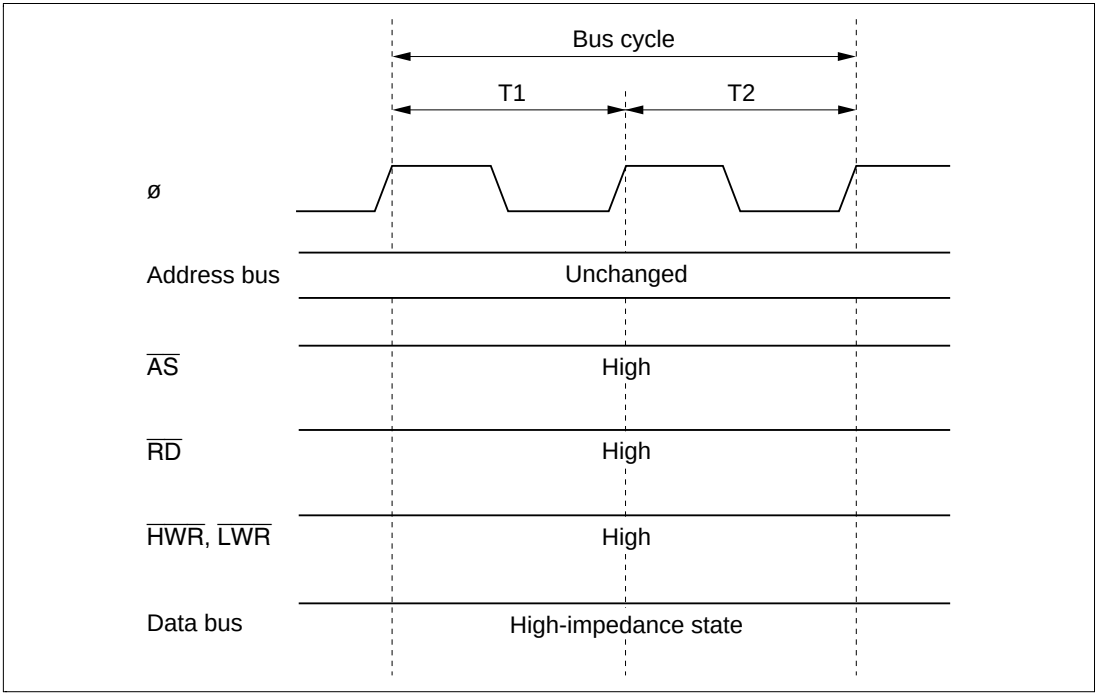


Figure 2-20 Pin States during On-Chip Supporting Module Access

2.9.4 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 7, Bus Controller.

2.10 Usage Note

2.10.1 TAS Instruction

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The TAS instruction is not generated by the Hitachi H8S and H8/300 series C/C++ compilers. If the TAS instruction is used as a user-defined intrinsic function, ensure that only register ER0, ER1, ER4, or ER5 is used.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8S/2633 Series has four operating modes (modes 4 to 7). These modes enable selection of the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus width setting, by setting the mode pins (MD2 to MD0).

Table 3-1 lists the MCU operating modes.

Table 3-1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus: Initial Width	External Data Bus: Max. Width
0*	0	0	0	—	—	—	—	
1*			1	—				
2*		1	0					
3*			1					
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits	16 bits
5			1				8 bits	16 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	

Note: * Not available in the H8S/2633 Series.

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2633 Series actually accesses a maximum of 16 Mbytes.

Modes 4 to 6 are externally expanded modes that allow access to external memory and peripheral devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set.

Note that the functions of each pin depend on the operating mode.

The H8S/2633 Series can be used only in modes 4 to 7. This means that the mode pins must be set to select one of these modes. Do not change the inputs at the mode pins during operation.

3.1.2 Register Configuration

The H8S/2633 Series has a mode control register (MDCR) that indicates the inputs at the mode pins (MD2 to MD0), and a system control register (SYSCR) that controls the operation of the H8S/2633 Series. Table 3-2 summarizes these registers.

Table 3-2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undetermined	H'FDE7
System control register	SYSCR	R/W	H'01	H'FDE5
Pin function control register	PFCR	R/W	H'0D/H'00	H'FDEB

Note: * Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	R/W	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit register that indicates the current operating mode of the H8S/2633 Series.

Bit 7—Reserved: Only 1 should be written to this bit.

Bits 6 to 3—Reserved: These bits always read as 0 and cannot be modified.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits-they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are cancelled by a power-on reset, but maintained by a manual reset.

3.2.2 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

SYSCR is an 8-bit readable-writable register that selects saturating or non-saturating calculation for the MAC instruction, selects the interrupt control mode, selects the detected edge for NMI, enables or disenables $\overline{\text{MRES}}$ pin input, and enables or disenables on-chip RAM.

SYSCR is initialized to H'01 by a power-on reset and in hardware standby mode. MACS, INTM1, INTM0, NMIEG, and RAME bits are initialized in manual reset mode, but the MRESE bit is not initialized. SYSCR is not initialized in software standby mode.

Bit 7—MAC Saturation (MACS): Selects either saturating or non-saturating calculation for the MAC instruction.

Bit 7		
MACS	**Description**	
0	Non-saturating calculation for MAC instruction	(Initial value)
1	Saturating calculation for MAC instruction	

Bit 6—Reserved: This bit always read as 0 and cannot be modified.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.4.1, Interrupt Control Modes and Interrupt Operation.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Control of interrupts by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Control of interrupts by I2 to I0 bits and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the valid edge of the NMI interrupt input.

Bit 3		
NMIEG	**Description**	
0	An interrupt is requested at the falling edge of NMI input	(Initial value)
1	An interrupt is requested at the rising edge of NMI input	

Bit 2—Manual Reset Selection Bit (MRESE): Enables or disenables manual reset input. It is possible to set the P74/TM02/$\overline{\text{MRES}}$ pin to the manual reset input ($\overline{\text{MRES}}$).

Table 3-3 shows the relationship between the $\overline{\text{MRES}}$ pin power-on reset and manual reset.

Bit 2		
MRESE	**Description**	
0	Disenables manual reset.	
	Possible to use P74/TM02/$\overline{\text{MRES}}$ pin as P74/TM02 input pin.	(Initial value)
1	Enables manual reset.	
	Possible to use P74/TM02/$\overline{\text{MRES}}$ pin as $\overline{\text{MRES}}$ input pin.	

Table 3-3 Relationship Between Power-On Reset and Manual Reset

Pin		
$\overline{\text{RES}}$	**$\overline{\text{MRES}}$**	**Reset Type**
0	*	Power-on reset (Initial state)
1	0	Manual reset
1	1	Operation state

*: Don't care

Bit 1—Reserved: This bit always read as 0 and cannot be modified.

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. It is not initialized in software standby mode.

Bit 0		
RAME	**Description**	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

Note: When the DTC is used, the RAME bit must be set to 1.

3.2.3 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		CSS07	CSS36	BUZZE	LCASS	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable-writable register that carries out CS selection control for PG4 and PG1 pins, LCAS selection control for PF2 and PF6 pins, and address output control during extension modes with ROM.

PFCR is initialized by H'0D/H'00 by a power-on reset or a hardware standby mode. The immediately previous state is maintained in manual reset or software standby mode.

Bit 7—$\overline{\text{CS0}}$/$\overline{\text{CS7}}$ Select (CSS07): Selects the CS output content for PG4 pin. In modes 4 to 6, the selected CS is output by setting the corresponding DDR to 1.

Bit 7

CSS07	Description	
0	Select $\overline{\text{CS0}}$.	(Initial value)
1	Select $\overline{\text{CS7}}$.	

Bit 6—$\overline{\text{CS3}}$/$\overline{\text{CS6}}$ Select (CSS36): Selects the CS output content for PG1 pin. In modes 4 to 6, the selected CS is output by setting the corresponding DDR to 1.

Bit 6

CSS36	Description	
0	Select $\overline{\text{CS3}}$.	(Initial value)
1	Select $\overline{\text{CS6}}$	

Bit 5—BUZZ Output Enable (BUZZE): Disenables/enables BUZZ output of PF1 pin. Input clock of WDT1 selected by PSS, CKS2 to CKS0 bits is output as a BUZZ signal.

Bit 5

BUZZE	Description	
0	Functions as PF1 input pin	(Initial value)
1	Functions as BUZZ output pin	

Bit 4—LCAS Output Pin Selection Bit (LCASS): Selects the LCAS signal output pin.

Bit 4

LCASS	Description	
0	Outputs LCAS signal from PF2	(Initial Value)
1	Outputs LCAS signal from PF6	

Bits 3 to 0—Address Output Enable 3 to 0 (AE3–AE0): These bits select enabling or disabling of address outputs A8 to A23 in ROMless expanded mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1.

Bit 3	Bit 2	Bit 1	Bit 0	
AE3	**AE2**	**AE1**	**AE0**	**Description**
0	0	0	0	A8–A23 address output disabled (Initial value*)
			1	A8 address output enabled; A9–A23 address output disabled
		1	0	A8, A9 address output enabled; A10–A23 address output disabled
			1	A8–A10 address output enabled; A11–A23 address output disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address output disabled
			1	A8–A12 address output enabled; A13–A23 address output disabled
		1	0	A8–A13 address output enabled; A14–A23 address output disabled
			1	A8–A14 address output enabled; A15–A23 address output disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address output disabled
			1	A8–A16 address output enabled; A17–A23 address output disabled
		1	0	A8–A17 address output enabled; A18–A23 address output disabled
			1	A8–A18 address output enabled; A19–A23 address output disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address output disabled
			1	A8–A20 address output enabled; A21–A23 address output disabled (Initial value*)
		1	0	A8–A21 address output enabled; A22, A23 address output disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.

In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.

Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

3.3 Operating Mode Descriptions

3.3.1 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Ports 1, A, B, and C, function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, note that if 8-bit access is designated by the bus controller for all areas, the bus mode switches to 8 bits.

3.3.2 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Ports 1, A, B, and C, function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.3 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled.

Ports 1, A, B, and C, function as input port pins immediately after a reset. Address output can be performed by setting the corresponding DDR (data direction register) bits to 1.

Port D function as a data bus, and part of port F carries data bus signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.4 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

3.4 Pin Functions in Each Operating Mode

The pin functions of ports A to F vary depending on the operating mode. Table 3-3 shows their functions in each operating mode.

Table 3-3 Pin Functions in Each Mode

Port		Mode 4	Mode 5	Mode 6	Mode 7
Port 1	$P1_0$	P/A*	P/A*	P*/A	P
	$P1_1$ to $P1_3$	P*/A	P*/A	P*/A	P
Port A	PA_4 to PA_0	A	A	P*/A	P
Port B		A	A	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF_7	P/C*	P/C*	P/C*	P*/C
	PF_6 to PF_4	C	C	C	P
	PF_3	P/C*	P*/C	P*/C	
	PF_2 to PF_0	P*/C	P*/C	P*/C	
Port G	PG_4	C	C	P*/C	P
	PG_3 to PG_0	P*/C	P*/C	P*/C	P

Legend

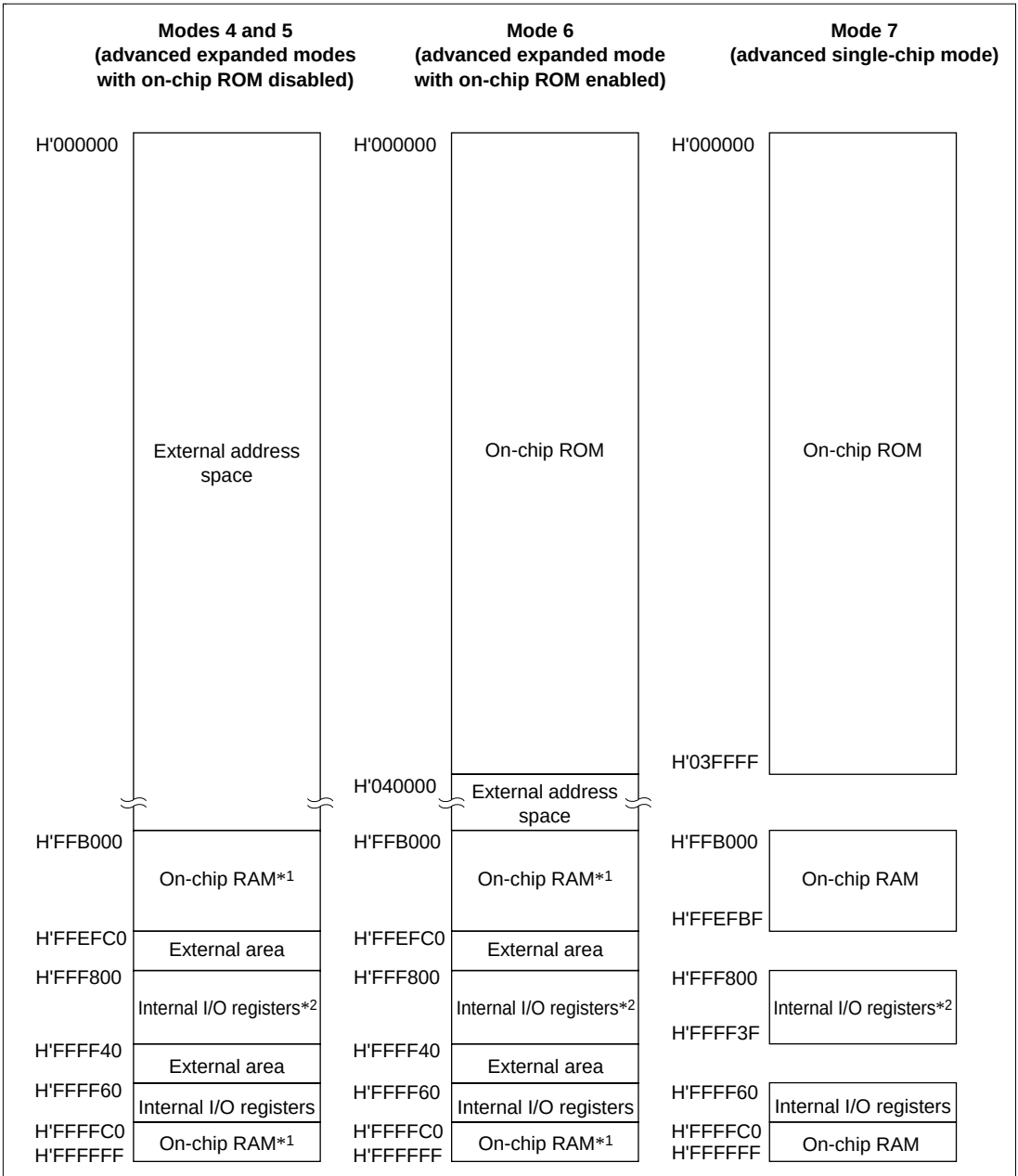
P: I/O port
A: Address bus output
D: Data bus I/O
C: Control signals, clock I/O
*: After reset

3.5 Address Map in Each Operating Mode

A address map of the H8S/2633 is shown in figure 3.1, and a address map of the H8S/2632 in figure 3-2.

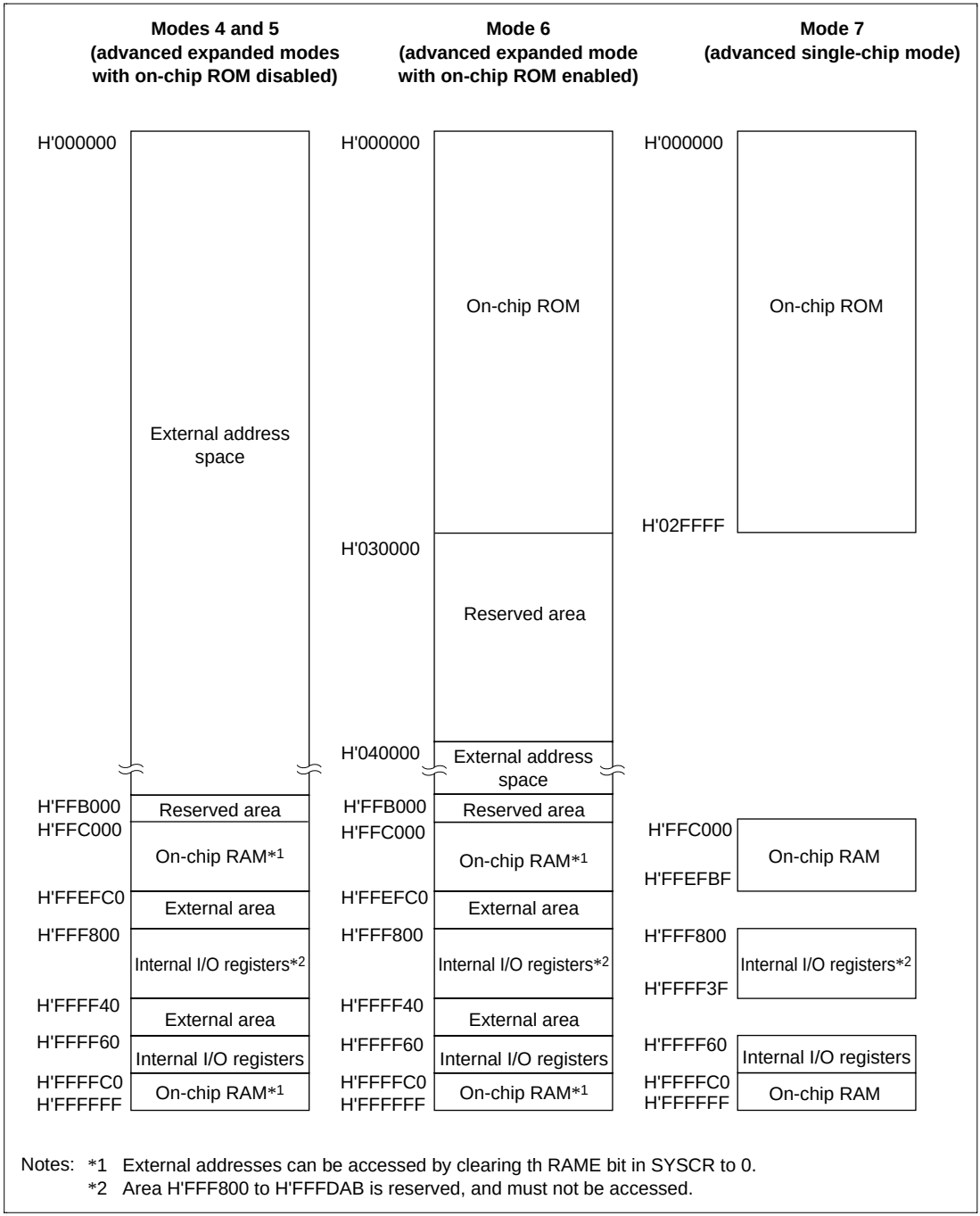
The address space is 16 kbytes in modes 4 to 7 (advanced modes).

The address space is divided into eight areas for modes 4 to 7. For details, see section 7, Bus Controller.



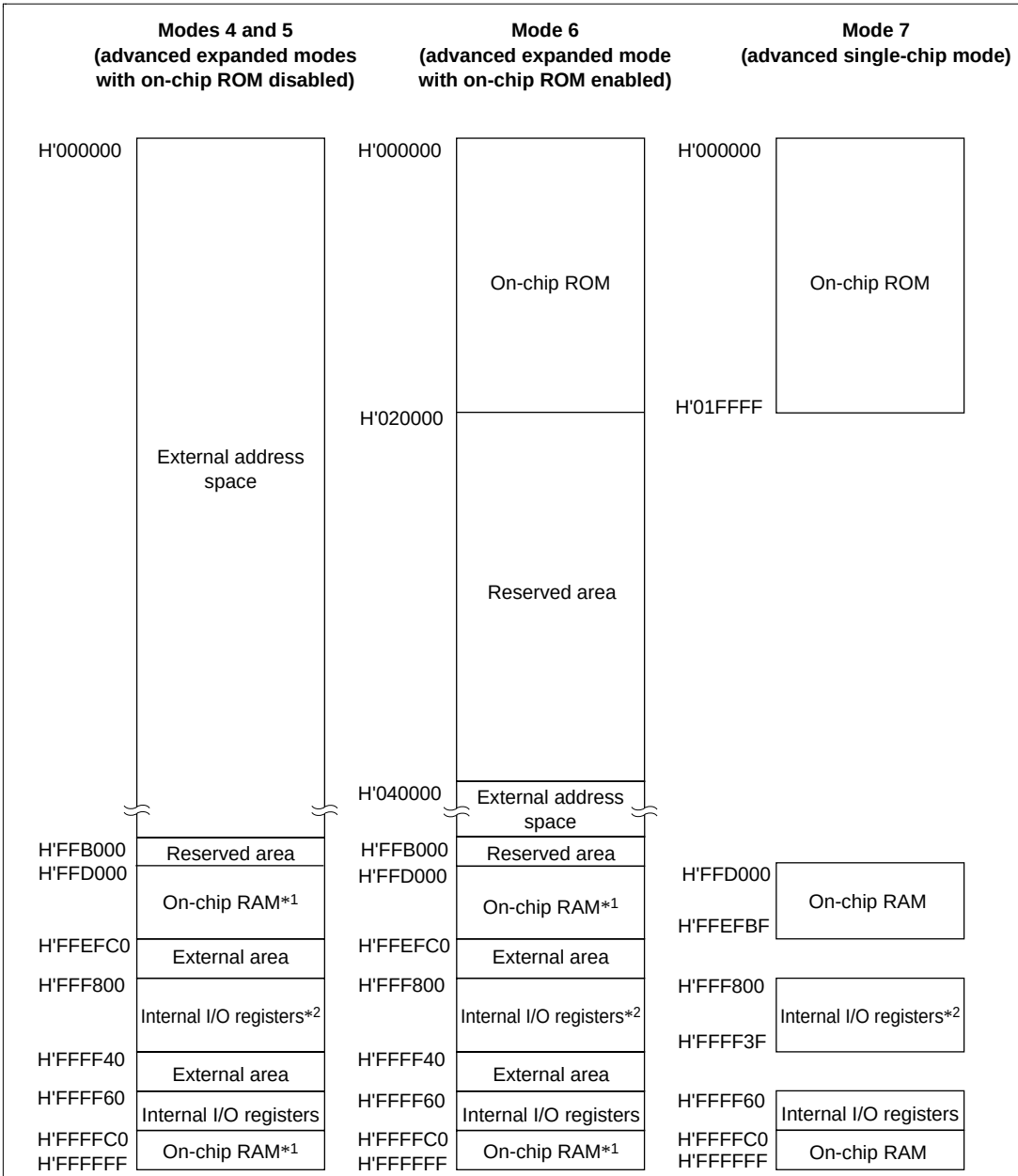
Notes: *1 External addresses can be accessed by clearing th RAME bit in SYSCR to 0.
*2 Area H'FFF800 to H'FFFDAB is reserved, and must not be accessed.

Figure 3-1 Memory Map in Each Operating Mode in the H8S/2633



Notes: *1 External addresses can be accessed by clearing th RAME bit in SYSCR to 0.
*2 Area H'FFF800 to H'FFFDAB is reserved, and must not be accessed.

Figure 3-2 Memory Map in Each Operating Mode in the H8S/2632



Notes: *1 External addresses can be accessed by clearing th RAME bit in SYSCR to 0.
*2 Area H'FFF800 to H'FFFDAB is reserved, and must not be accessed.

Figure 3-3 Memory Map in Each Operating Mode in the H8S/2631

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4-1 indicates, exception handling may be caused by a reset, direct transition, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4-1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Trap instruction exceptions are accepted at all times, in the program execution state.

Exception handling sources, the stack structure, and the operation of the CPU vary depending on the interrupt control mode set by the INTM0 and INTM1 bits of SYSCR.

Table 4-1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin, or when the watchdog overflows. The CPU enters the power-on reset state when the $\overline{\text{RES}}$ pin is low, and the manual reset state when the $\overline{\text{MRES}}$ pin is low.
	Trace*1	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1
	Direct transition	Starts when a direct transition occurs due to execution of a SLEEP instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued*2
Low	Trap instruction (TRAPA)*3	Started by execution of a trap instruction (TRAPA)

Notes: *1 Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.

*2 Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.

*3 Trap instruction exception handling requests are accepted at all times in program execution state.

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows:

1. The program counter (PC), condition code register (CCR), and extended register (EXR) are pushed onto the stack.
2. The interrupt mask bits are updated. The T bit is cleared to 0.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4-1. Different vector addresses are assigned to different exception sources.

Table 4-2 lists the exception sources and their vector addresses.

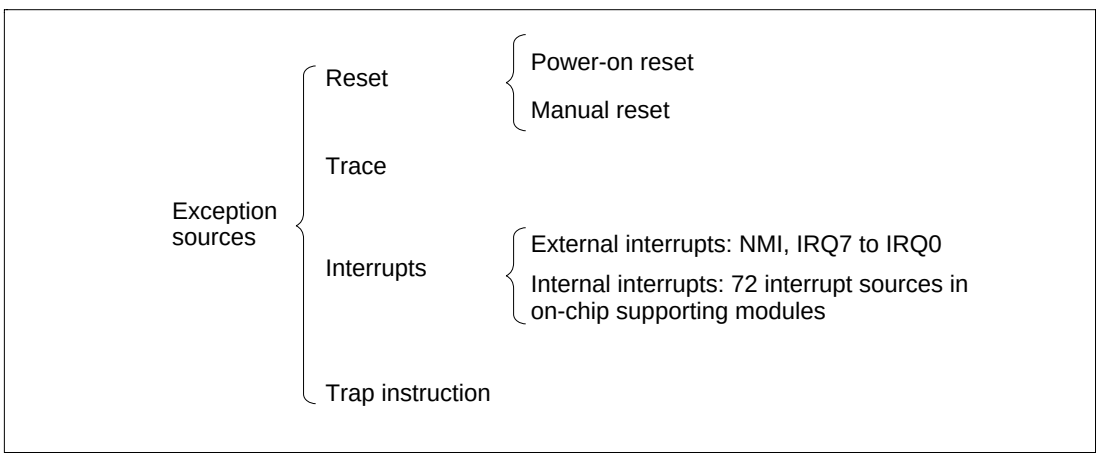


Figure 4-1 Exception Sources

Table 4-2 Exception Vector Table

Exception Source		Vector Number	Vector Address*[1] Advanced Mode
Power-on reset		0	H'0000 to H'0003
Manual reset*[3]		1	H'0004 to H'0007
Reserved for system use		2	H'0008 to H'000B
		3	H'000C to H'000F
		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Direct transition*[3]		6	H'0018 to H'001B
External interrupt	NMI	7	H'001C to H'001F
Trap instruction (4 sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved for system use		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
	IRQ6	22	H'0058 to H'005B
	IRQ7	23	H'005C to H'005F
Internal interrupt*[2]		24 | 127	H'0060 to H'0063 | H'01FC to H'01FF

Notes: *1 Lower 16 bits of the address.

*2 For details of internal interrupt vectors, see section 5.3.3, Interrupt Exception Handling Vector Table.

*3 See 24.11 Direct Transition for details on direct transition.

4.2 Reset

4.2.1 Overview

A reset has the highest exception handling priority. There are two kinds of reset: a power-on reset executed via the $\overline{\text{RES}}$ pin, and a manual reset executed via the $\overline{\text{MRES}}$ pin.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin* goes low, currently executing processing is halted and the chip enters the reset state. A reset initializes the internal state of the CPU and the registers of on-chip supporting modules. Immediately after a reset, interrupt control mode 0 is set.

Reset exception handling starts when the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin* changes from low to high.

The reset state can also be entered in the event of watchdog timer overflow. For details see section 15, Watchdog Timer.

Note: * $\overline{\text{MRES}}$ pin in the case of a manual reset.

4.2.2 Types of Reset

There are two types of reset: power-on reset and manual reset.

Table 4-3 shows the types of reset. When turning power on, do so as a power-on reset.

Both power-on reset and manual reset initialize the internal state of the CPU. In a power-on reset, all of the registers of the built-in vicinity modules are initialized, while in a manual reset, the registers of the built-in vicinity models except for bus controllers and I/O ports are initialized. The states of the bus controllers and I/O ports are maintained.

During a manual reset built-in vicinity modules are initialized, and ports used as input pins for built-in vicinity modules switch to the input ports controlled by DDR and DR.

If using manual reset, set the MRESE bit to 1 beforehand, thereby enabling manual resets.

See 3.2.2 System Control Register (SYSCR) for settings of the MRESE bit.

There are also power-on resets and manual resets as the two types of reset carried out by the watchdog timer.

Table 4-3 Types of Reset

Type	Conditions for Transition to Reset		Internal State	
	$\overline{\text{MRES}}$	$\overline{\text{RES}}$	CPU	Built-in vicinity module
Power-on reset	*	Low	Initialization	Initialization
Manual reset	Low	High	Initialization	Initialization except for bus controller and I/O port

*: Don't Care

4.2.3 Reset Sequence

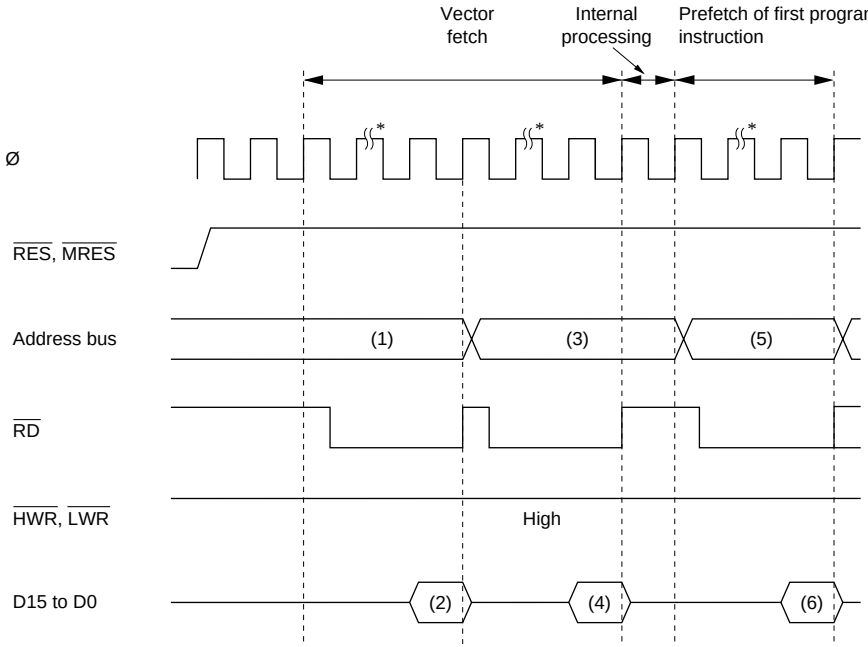
This LSI enters reset state when the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin goes low.

To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin or the $\overline{\text{MRES}}$ pin low for at least 20 ms at power-up. To reset during operation, hold the $\overline{\text{RES}}$ pin or the $\overline{\text{MRES}}$ pin low for at least 20 states.

When the $\overline{\text{RES}}$ pin or the $\overline{\text{MRES}}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows.

1. The internal state of the CPU and the registers of the on-chip supporting modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

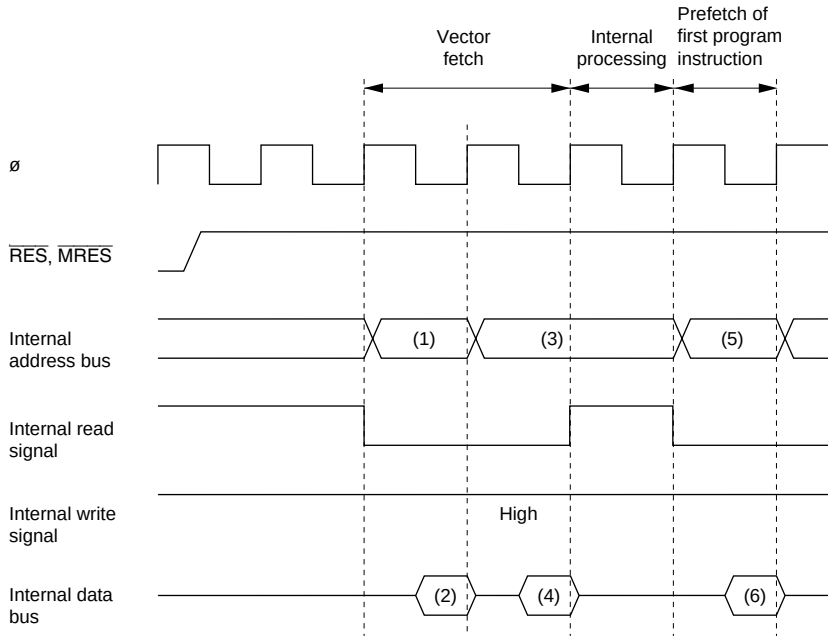
Figures 4-2 and 4-3 show examples of the reset sequence.



(1) (3) Reset exception handling vector address (when power-on reset, (1) = H'000000*, (3) = H'000002; when manual reset, (1)= H'000004, (3)= H'000006)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * 3 program wait states are inserted.

Figure 4-2 Reset Sequence (Modes 4 and 5)



(1) (3) Reset exception handling vector address (when power-on reset, (1) = H'000000, (3) = H'000002)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Figure 4-3 Reset Sequence (Modes 6 and 7)

4.2.4 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

4.2.5 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCRA to MSTPCRC are initialized to H'3F, H'FF, and H'FF, respectively, and all modules except the DMAC and DTC, enter module stop mode. Consequently, on-chip supporting module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

4.3 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details of interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction.

Trace mode is canceled by clearing the T bit in EXR to 0. It is not affected by interrupt masking.

Table 4-4 shows the state of CCR and EXR after execution of trace exception handling.

Interrupts are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes.

Trace exception handling is not carried out after execution of the RTE instruction.

Table 4-4 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

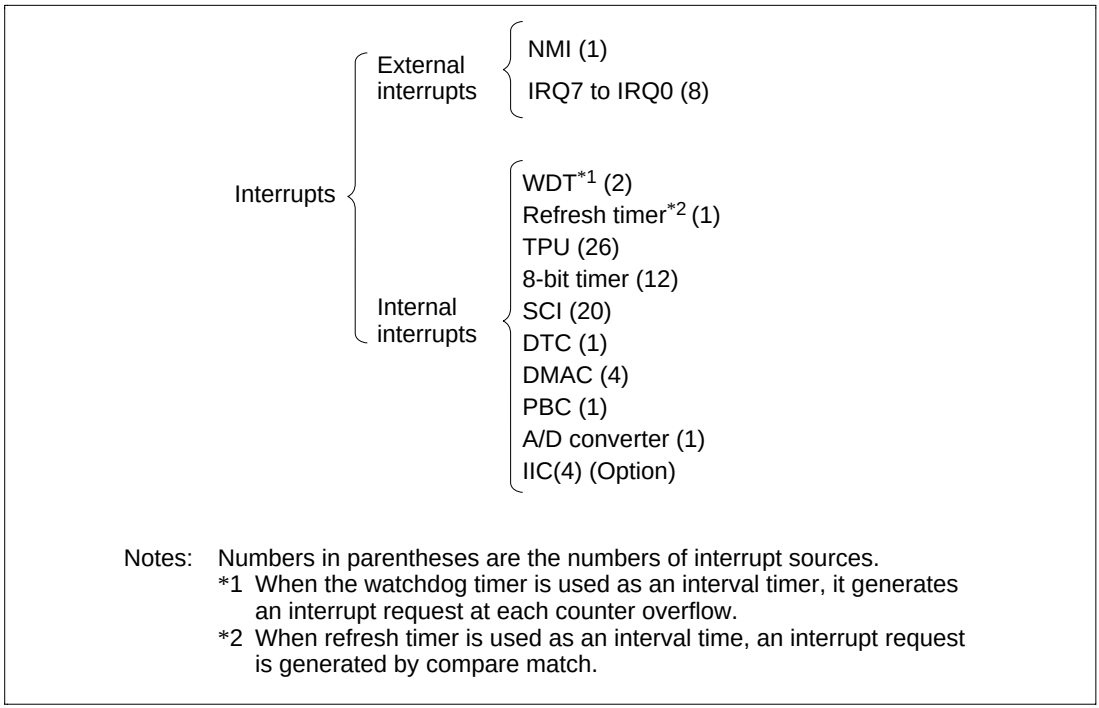
4.4 Interrupts

Interrupt exception handling can be requested by nine external sources (NMI, IRQ7 to IRQ0) and 72 internal sources in the on-chip supporting modules. Figure 4-4 classifies the interrupt sources and the number of interrupts of each type.

The on-chip supporting modules that can request interrupts include the watchdog timer (WDT), 16-bit timer-pulse unit (TPU), 8-bit timer, serial communication interface (SCI), data transfer controller (DTC), DMA controller (DMAC), PC break controller (PBC), A/D converter, and I^2C bus interface (IIC). Each interrupt source has a separate vector address.

NMI is the highest-priority interrupt. Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control.

For details of interrupts, see section 5, Interrupt Controller.



Notes: Numbers in parentheses are the numbers of interrupt sources.
*1 When the watchdog timer is used as an interval timer, it generates an interrupt request at each counter overflow.
*2 When refresh timer is used as an interval time, an interrupt request is generated by compare match.

Figure 4-4 Interrupt Sources and Number of Interrupts

4.5 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4-5 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4-5 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

4.6 Stack Status after Exception Handling

Figure 4-5 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

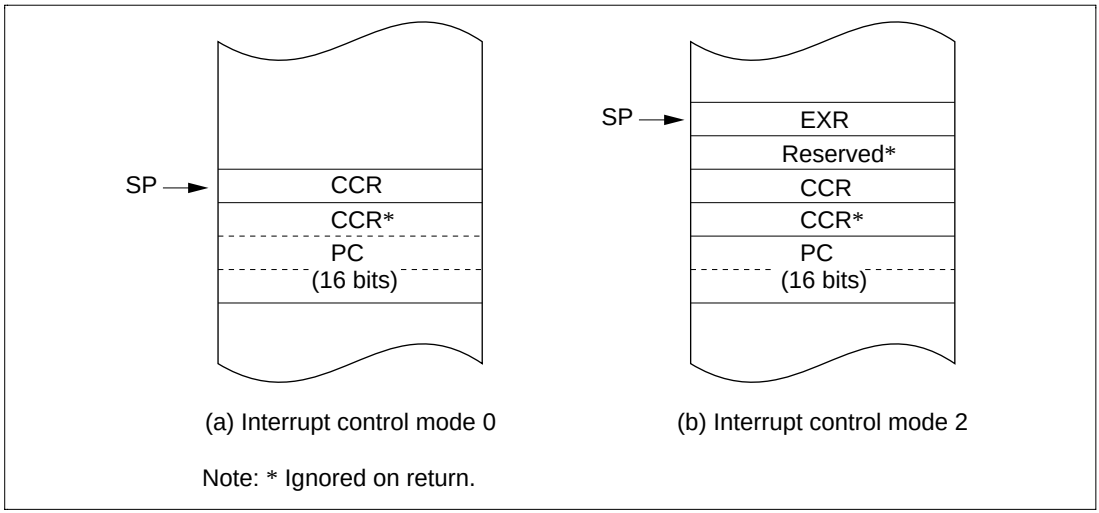


Figure 4-5 (1) Stack Status after Exception Handling (Normal Modes: Not Available in the H8S/2633 Series)

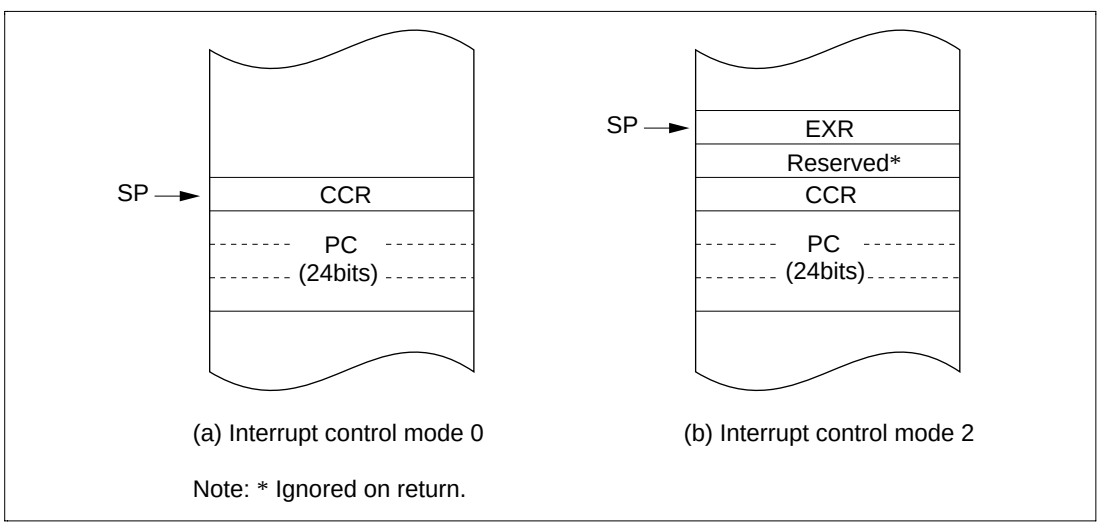


Figure 4-5 (2) Stack Status after Exception Handling (Advanced Modes)

4.7 Notes on Use of the Stack

When accessing word data or longword data, the H8S/2633 Series assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4-6 shows an example of what happens when the SP value is odd.

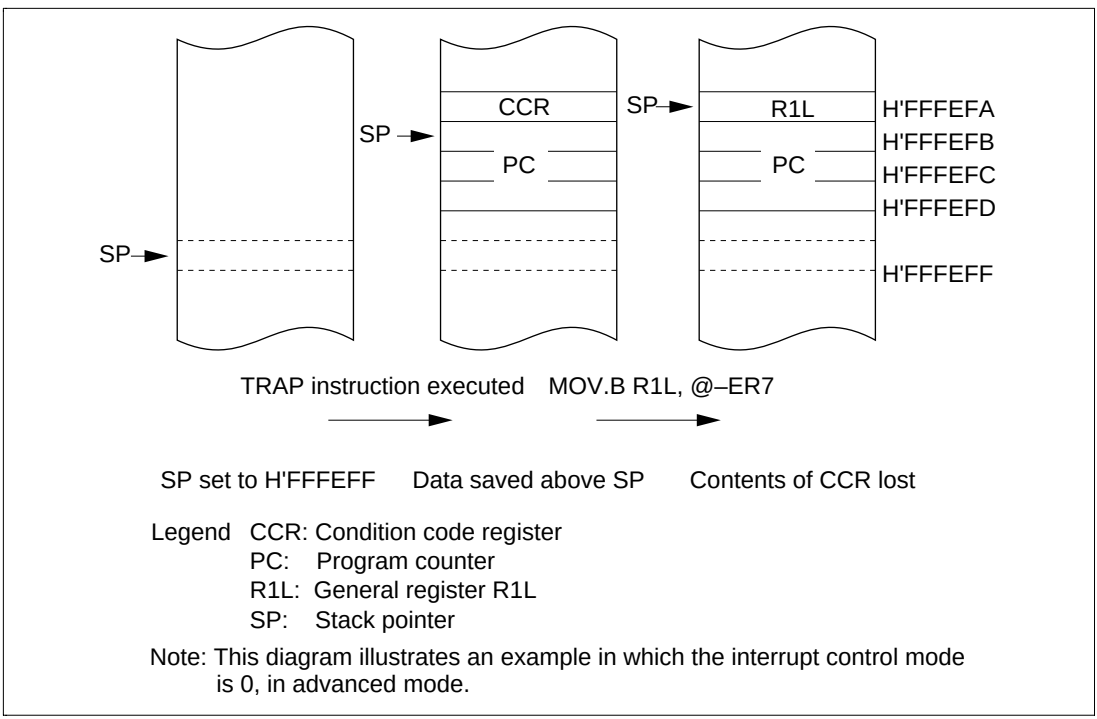


Figure 4-6 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

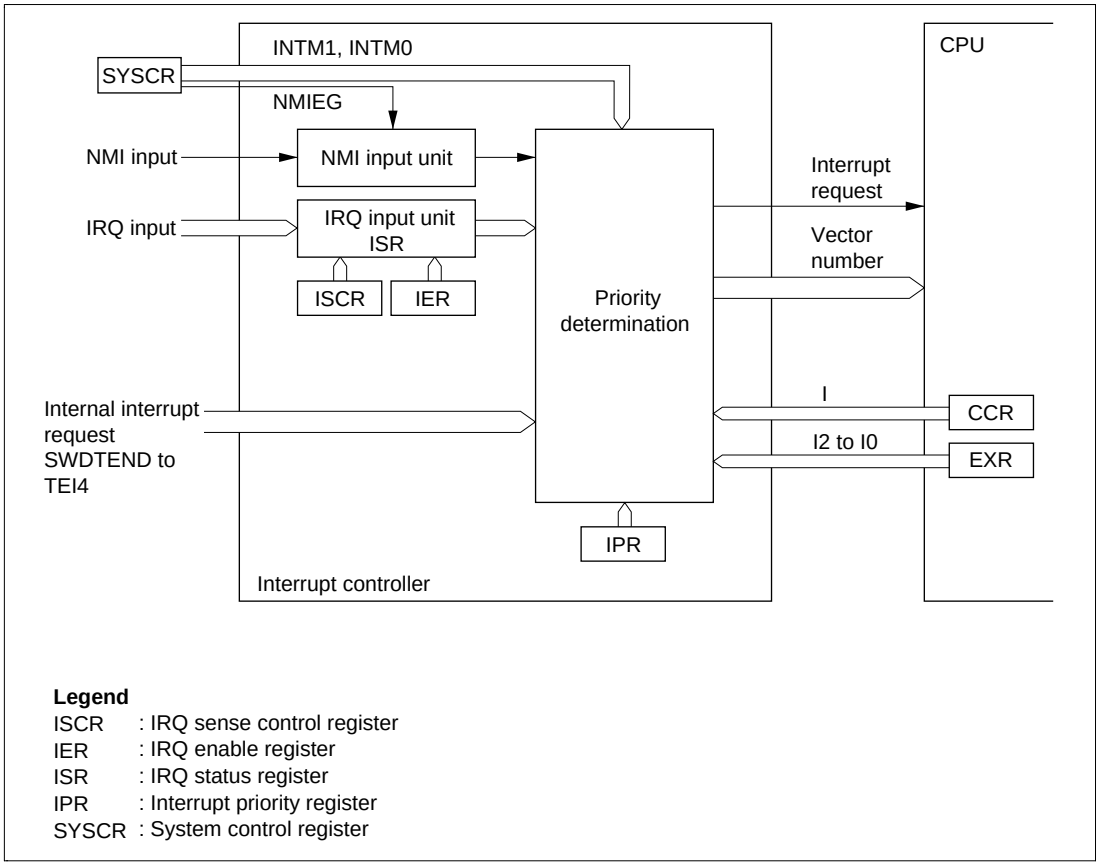
5.1.1 Features

The H8S/2633 Series controls interrupts by means of an interrupt controller. The interrupt controller has the following features:

- Two interrupt control modes
  - — Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with IPR
  - — An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI.
  - — NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses
  - — All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Nine external interrupts
  - — NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI.
  - — Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ7 to IRQ0.
- DTC and DMAC control
  - — DTC and DMAC activation is performed by means of interrupts.

5.1.2 Block Diagram

A block diagram of the interrupt controller is shown in Figure 5-1.



Legend

ISCR : IRQ sense control register
IER : IRQ enable register
ISR : IRQ status register
IPR : Interrupt priority register
SYSCR : System control register

Figure 5-1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5-1 summarizes the pins of the interrupt controller.

Table 5-1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 7 to 0	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected

5.1.4 Register Configuration

Table 5-2 summarizes the registers of the interrupt controller.

Table 5-2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
System control register	SYSCR	R/W	H'01	H'FDE5
IRQ sense control register H	ISCRH	R/W	H'00	H'FE12
IRQ sense control register L	ISCRL	R/W	H'00	H'FE13
IRQ enable register	IER	R/W	H'00	H'FE14
IRQ status register	ISR	R/(W)*[2]	H'00	H'FE15
Interrupt priority register A	IPRA	R/W	H'77	H'FEC0
Interrupt priority register B	IPRB	R/W	H'77	H'FEC1
Interrupt priority register C	IPRC	R/W	H'77	H'FEC2
Interrupt priority register D	IPRD	R/W	H'77	H'FEC3
Interrupt priority register E	IPRE	R/W	H'77	H'FEC4
Interrupt priority register F	IPRF	R/W	H'77	H'FEC5
Interrupt priority register G	IPRG	R/W	H'77	H'FEC6
Interrupt priority register H	IPRH	R/W	H'77	H'FEC7
Interrupt priority register I	IPRI	R/W	H'77	H'FEC8
Interrupt priority register J	IPRJ	R/W	H'77	H'FEC9
Interrupt priority register K	IPRK	R/W	H'77	H'FECA
Interrupt priority register L	IPRL	R/W	H'77	H'FECB
Interrupt priority register O	IPRO	R/W	H'77	H'FECE

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, and the detected edge for NMI.

Only bits 5 to 3 are described here; for details of the other bits, see section 3.2.2, System Control Register (SYSCR).

SYSCR is initialized to H'01 by a power-on reset, manual reset, and in hardware standby mode. SYSCR is not initialized in software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select one of two interrupt control modes for the interrupt controller.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Interrupts are controlled by bits I2 to I0, and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 3		
NMIEG	**Description**	
0	Interrupt request generated at falling edge of NMI input	(Initial value)
1	Interrupt request generated at rising edge of NMI input	

5.2.2 Interrupt Priority Registers A to L, O (IPRA to IPRL, IPRO)

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

The IPR registers are thirteen 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between IPR settings and interrupt sources is shown in table 5-3.

The IPR registers set a priority (level 7 to 0) for each interrupt source other than NMI.

The IPR registers are initialized to H'77 by a reset and in hardware standby mode.

Bits 7 and 3—Reserved: These bits are always read as 0 and cannot be modified.

Table 5-3 Correspondence between Interrupt Sources and IPR Settings

	Bits	
Register	**6 to 4**	**2 to 0**
IPRA	IRQ0	IRQ1
IPRB	IRQ2 IRQ3	IRQ4 IRQ5
IPRC	IRQ6 IRQ7	DTC
IPRD	Watchdog timer 0	Refresh timer
IPRE	PC break	A/D converter, watchdog timer 1
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRL	8-bit timer 2, 3	IIC (Option)
IPRO	SCI channel 3	SCI channel 4

As shown in table 5-3, multiple interrupts are assigned to one IPR. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corresponding interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest priority level, level 7, by setting H'7.

When interrupt requests are generated, the highest-priority interrupt according to the priority levels set in the IPR registers is selected. This interrupt level is then compared with the interrupt mask level set by the interrupt mask bits (I2 to I0) in the extend register (EXR) in the CPU, and if the priority level of the interrupt is higher than the set mask level, an interrupt request is issued to the CPU.

5.2.3 IRQ Enable Register (IER)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IER is an 8-bit readable/writable register that controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

IER is initialized to H'00 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 to 0—IRQ7 to IRQ0 Enable (IRQ7E to IRQ0E): These bits select whether IRQ7 to IRQ0 are enabled or disabled.

Bit n		
IRQnE	**Description**	
0	IRQn interrupts disabled	(Initial value)
1	IRQn interrupts enabled	

(n = 7 to 0)

5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ISCR registers are 16-bit readable/writable registers that select rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

The ISCR registers are initialized to H'0000 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 15 to 0: IRQ7 Sense Control A and B (IRQ7SCA, IRQ7SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

Bits 15 to 0

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input low level (initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

5.2.5 IRQ Status Register (ISR)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

ISR is an 8-bit readable/writable register that indicates the status of IRQ7 to IRQ0 interrupt requests.

ISR is initialized to H'00 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 to 0—IRQ7 to IRQ0 flags (IRQ7F to IRQ0F): These bits indicate the status of IRQ7 to IRQ0 interrupt requests.

Bit n **IRQnF**	**Description**
0	[Clearing conditions] (Initial value) • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input when both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 5 to 0)

5.3 Interrupt Sources

Interrupt sources comprise external interrupts (NMI and IRQ7 to IRQ0) and internal interrupts (72 sources).

5.3.1 External Interrupts

There are nine external interrupts: NMI and IRQ7 to IRQ0. Of these, NMI and IRQ7 to IRQ0 can be used to restore the H8S/2633 Series from software standby mode.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

The vector number for NMI interrupt exception handling is 7.

IRQ7 to IRQ0 Interrupts: Interrupts IRQ7 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ5 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 5-2.

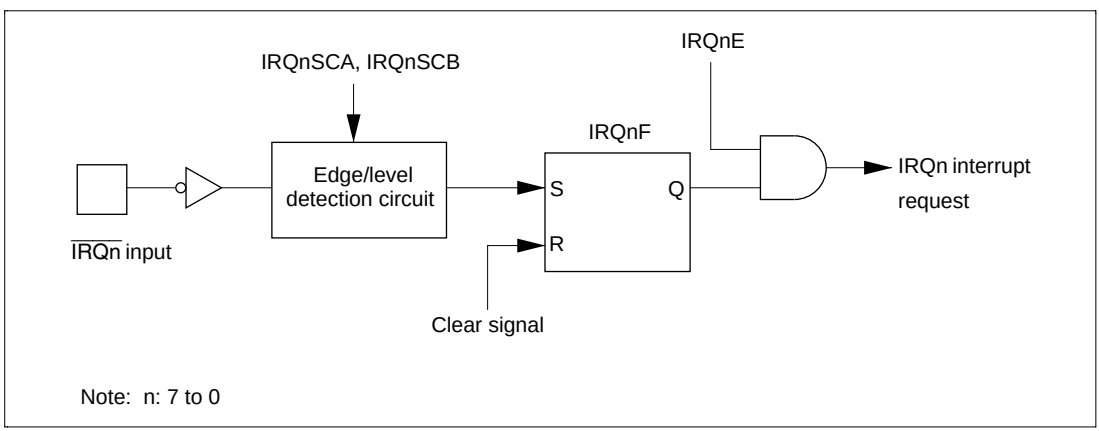


Figure 5-2 Block Diagram of Interrupts IRQ7 to IRQ0

Figure 5-3 shows the timing of setting IRQnF.

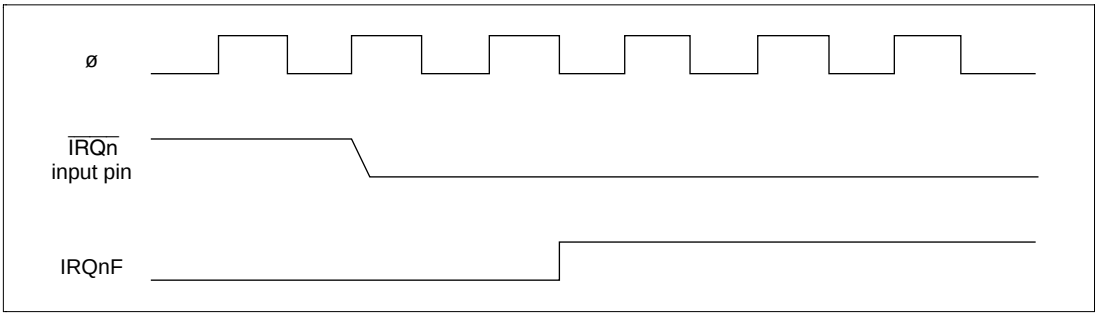


Figure 5-3 Timing of Setting IRQnF

The vector numbers for IRQ7 to IRQ0 interrupt exception handling are 23 to 16.

Detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 and use the pin as an I/O pin for another function.

5.3.2 Internal Interrupts

There are 72 sources for internal interrupts from on-chip supporting modules.

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DMAC and DTC can be activated by a TPU, 8-bit timer, SCI, or other interrupt request. When the DMAC and DTC are activated by an interrupt, the interrupt control mode and interrupt mask bits are not affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5-4 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. The situation when two or more modules are set to the same priority, and priorities within a module, are fixed as shown in table 5-4.

Table 5-4 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
NMI	External pin	7	H'001C		High
IRQ0		16	H'0040	IPRA6 to 4	
IRQ1		17	H'0044	IPRA2 to 0	
IRQ2 IRQ3		18 19	H'0048 H'004C	IPRB6 to 4	
IRQ4 IRQ5		20 21	H'0050 H'0054	IPRB2 to 0	
IRQ6 IRQ7		22 23	H'0058 H'005C	IPRC6 to 4	
SWDTEND (software activation interrupt end)	DTC	24	H'0060	IPRC2 to 0	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0064	IPRD6 to 4	
CMI	Refresh timer	26	H'0068	IPRD2 to 0	
PC break	PC break	27	H'006C	IPRE6 to 4	
ADI (A/D conversion end)	A/D	28	H'0070	IPRE2 to 0	
WOVI1 (interval timer)	Watchdog timer 1	29	H'0074		
Reserved	—	30 31	H'0078 H'007C		
TGI0A (TGR0A input capture/compare match)	TPU channel 0	32	H'0080	IPRF6 to 4	
TGI0B (TGR0B input capture/compare match)		33	H'0084		
TGI0C (TGR0C input capture/compare match)		34	H'0088		
TGI0D (TGR0D input capture/compare match)		35	H'008C		
TCI0V (overflow 0)		36	H'0090		
Reserved	—	37 38 39	H'0094 H'0098 H'009C		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
TGI1A (TGR1A input capture/compare match)	TPU channel 1	40	H'00A0	IPRF2 to 0	High
TGI1B (TGR1B input capture/compare match)		41	H'00A4		
TCI1V (overflow 1)		42	H'00A8		
TCI1U (underflow 1)		43	H'00AC		
TGI2A (TGR2A input capture/compare match)	TPU channel 2	44	H'00B0	IPRG6 to 4	
TGI2B (TGR2B input capture/compare match)		45	H'00B4		
TCI2V (overflow 2)		46	H'00B8		
TCI2U (underflow 2)		47	H'00BC		
TGI3A (TGR3A input capture/compare match)	TPU channel 3	48	H'00C0	IPRG2 to 0	
TGI3B (TGR3B input capture/compare match)		49	H'00C4		
TGI3C (TGR3C input capture/compare match)		50	H'00C8		
TGI3D (TGR3D input capture/compare match)		51	H'00CC		
TCI3V (overflow 3)		52	H'00D0		
Reserved	—	53	H'00D4		
		54	H'00D8		
		55	H'00DC		
TGI4A (TGR4A input capture/compare match)	TPU channel 4	56	H'00E0	IPRH6 to 4	
TGI4B (TGR4B input capture/compare match)		57	H'00E4		
TCI4V (overflow 4)		58	H'00E8		
TCI4U (underflow 4)		59	H'00EC		
TGI5A (TGR5A input capture/compare match)	TPU channel 5	60	H'00F0	IPRH2 to 0	
TGI5B (TGR5B input capture/compare match)		61	H'00F4		
TCI5V (overflow 5)		62	H'00F8		
TCI5U (underflow 5)		63	H'00FC		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
CMIA0 (compare match A0)	8-bit timer	64	H'0100	IPRI6 to 4	High
CMIB0 (compare match B0)	channel 0	65	H'0104		
OVI0 (overflow 0)		66	H'0108		
Reserved	—	67	H'010C		
CMIA1 (compare match A1)	8-bit timer	68	H'0110	IPRI2 to 0	
CMIB1 (compare match B1)	channel 1	69	H'0114		
OVI1 (overflow 1)		70	H'0118		
Reserved	—	71	H'011C		
DED0A (channel 0/channel 0A transfer end)	DMAC	72	H'0120	IPRJ6 to 4	
DEND0B (channel 0B transfer end)		73	H'0124		
DEND1A (channel 1/channel 1A transfer end)		74	H'0128		
DEND1B (channel 1B transfer end)		75	H'012C		
Reserved	—	76	H'0130		
		77	H'0134		
		78	H'0138		
		79	H'013C		
ERI0 (receive error 0)	SCI	80	H'0140	IPRJ2 to 0	
RXI0 (reception completed 0)	channel 0	81	H'0144		
TXI0 (transmit data empty 0)		82	H'0148		
TEI0 (transmission end 0)		83	H'014C		
ERI1 (receive error 1)	SCI	84	H'0150	IPRK6 to 4	
RXI1 (reception completed 1)	channel 1	85	H'0154		
TXI1 (transmit data empty 1)		86	H'0158		
TEI1 (transmission end 1)		87	H'015C		
ERI2 (receive error 2)	SCI	88	H'0160	IPRK2 to 0	
RXI2 (reception completed 2)	channel 2	89	H'0164		
TXI2 (transmit data empty 2)		90	H'0168		
TEI2 (transmission end 2)		91	H'016C		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
CMIA0 (compare match A2)	8 bit timer	92	H'0170	IPRL6 to 4	High
CMIB0 (compare match B2)	channel 2	93	H'0174		
OVI0 (overflow 2)		94	H'0178		
Reserved	—	95	H'017C		
CMIA1 (compare match A3)	8 bit timer	96	H'0180		
CMIB1 (compare match B3)	channel 3	97	H'0184		
OVI1 (overflow 3)		98	H'0188		
Reserved	—	99	H'018C		
IICI0 (1 byte transmission/reception completed)	IIC channel 0 (optional)	100	H'0190	IPRL2 to 0	
DDCSW1 (format switch)		101	H'0194		
IICI1 (1 byte transmission/reception completed)	IIC channel 1 (optional)	102	H'0198		
Reserved		103	H'019C		
Reserved	—	104	H'01A0	IPRM6 to 4	
		105	H'01A4		
		106	H'01A8		
		107	H'01AC		
Reserved	—	108	H'01B0	IPRM2 to 0	
		109	H'01B4		
		110	H'01B8		
		111	H'01BC		
Reserved	—	112	H'01C0	IPRN6 to 4	
		113	H'01C4		
		114	H'01C8		
		115	H'01CC		
Reserved	—	116	H'01D0	IPRN2 to 0	
		117	H'01D4		
		118	H'01D8		
		119	H'01DC		
ERI3 (reception error 3)	SCI	120	H'01E0	IPRO6 to 4	
RXI3 (reception completed 3)	channel 3	121	H'01E4		
TXI3 (transmission data empty 3)		122	H'01E8		
TEI3 (transmission end 3)		123	H'01EC		
ERI4 (reception error 4)	SCI	124	H'01F0	IPRO2 to 0	
RXI4 (reception completed 4)	channel 4	125	H'01F4		
TXI4 (transmission data empty 4)		126	H'01F8		
TEI4 (transmission end 4)		127	H'01FC		Low

Note: * Lower 16 bits of the start address.

5.4 Interrupt Operation

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the H8S/2633 Series differ depending on the interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware standby state. In the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5-5 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mode set by the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state indicated by the I and UI bits in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5-5 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
—		1	—	—	Setting prohibited

Figure 5-4 shows a block diagram of the priority decision circuit.

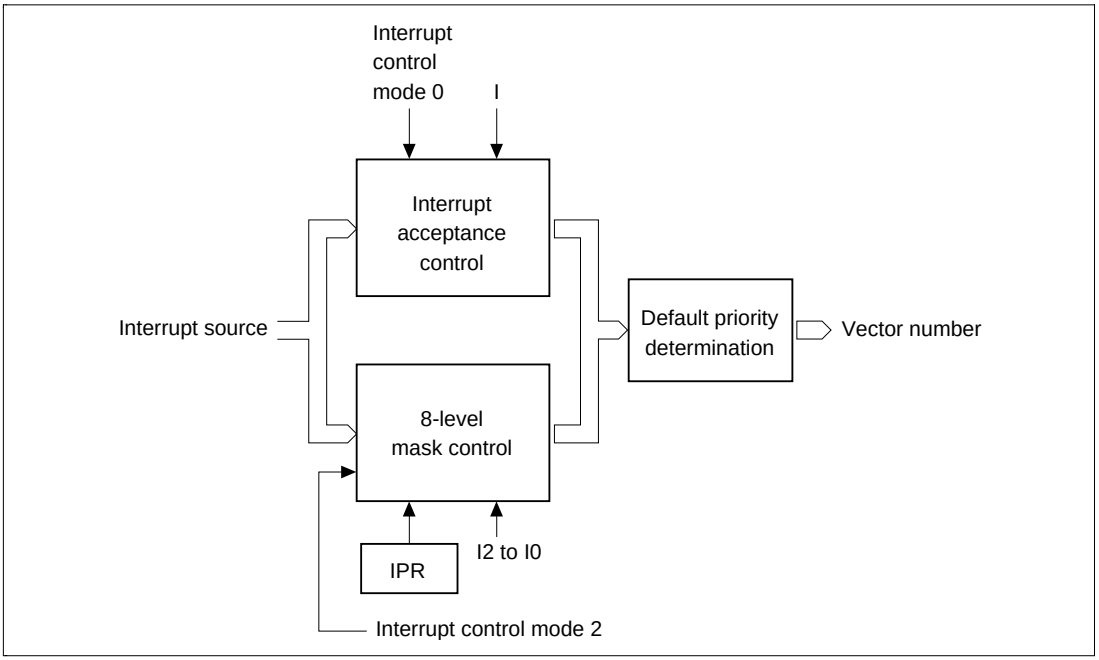


Figure 5-4 Block Diagram of Interrupt Control Operation

(1) Interrupt Acceptance Control

In interrupt control mode 0, interrupt acceptance is controlled by the I bit in CCR.

Table 5-6 shows the interrupts selected in each interrupt control mode.

Table 5-6 Interrupts Selected in Each Interrupt Control Mode (1)

	Interrupt Mask Bits	
Interrupt Control Mode	**I**	**Selected Interrupts**
0	0	All interrupts
	1	NMI interrupts
2	*	All interrupts

Legend
* : Don't care

(2) 8-Level Control

In interrupt control mode 2, 8-level mask level determination is performed for the selected interrupts in interrupt acceptance control according to the interrupt priority level (IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose priority level set in IPR is higher than the mask level.

Table 5-7 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest-priority-level (IPR) interrupt whose priority level is greater than the mask level (IPR > I2 to I0).

(3) Default Priority Determination

When an interrupt is selected by 8-level control, its priority is determined and a vector number is generated.

If the same value is set for IPR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5-8 shows operations and control signal functions in each interrupt control mode.

Table 5-8 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control		8-Level Control			Default Priority Determination	T (Trace)
	INTM1	INTM0		I		I2 to I0	IPR		
0	0	0	○	IM	X	—	—*2	○	—
2	1	0	X	—*1	○	IM	PR	○	T

Legend

○ : Interrupt operation control performed
X : No operation. (All interrupts enabled)
IM : Used as interrupt mask bit
PR : Sets priority.
— : Not used.

Notes: *1 Set to 1 when interrupt is accepted.
*2 Keep the initial setting.

5.4.2 Interrupt Control Mode 0

Enabling and disabling of IRQ interrupts and on-chip supporting module interrupts can be set by means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Figure 5-5 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.

[3] Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

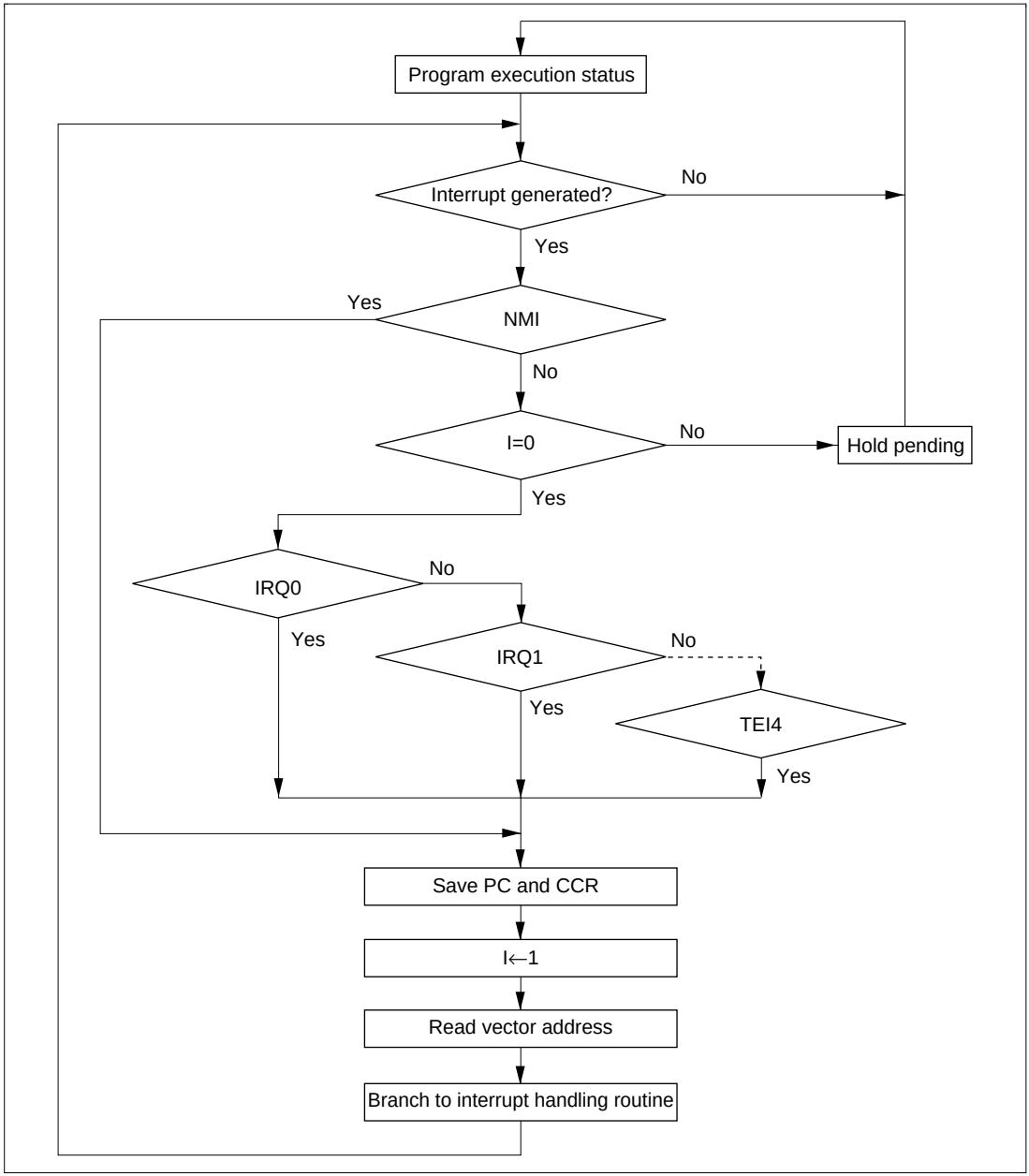


Figure 5-5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.4.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5-6 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5-4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

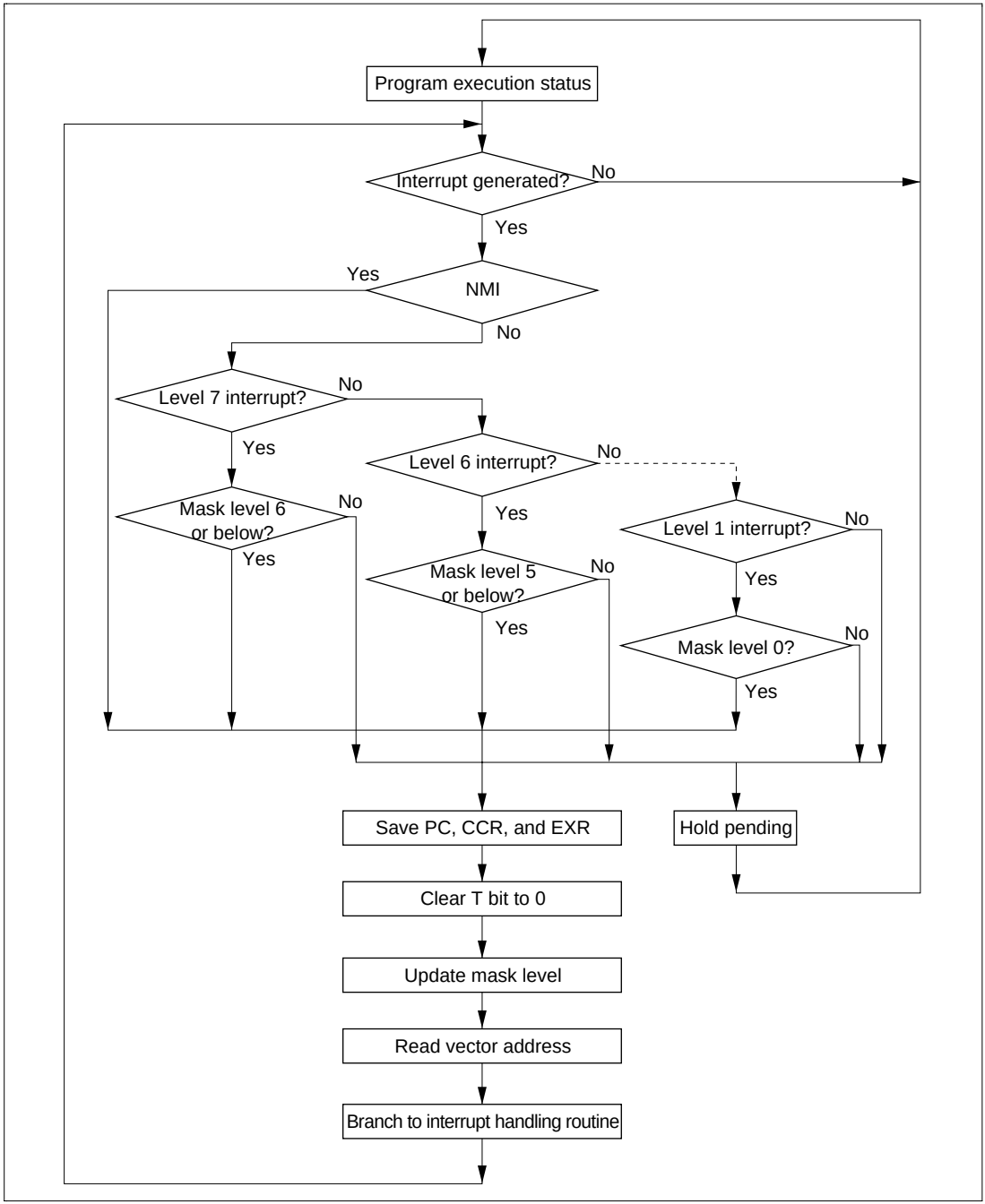


Figure 5-6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.4.4 Interrupt Exception Handling Sequence

Figure 5-7 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

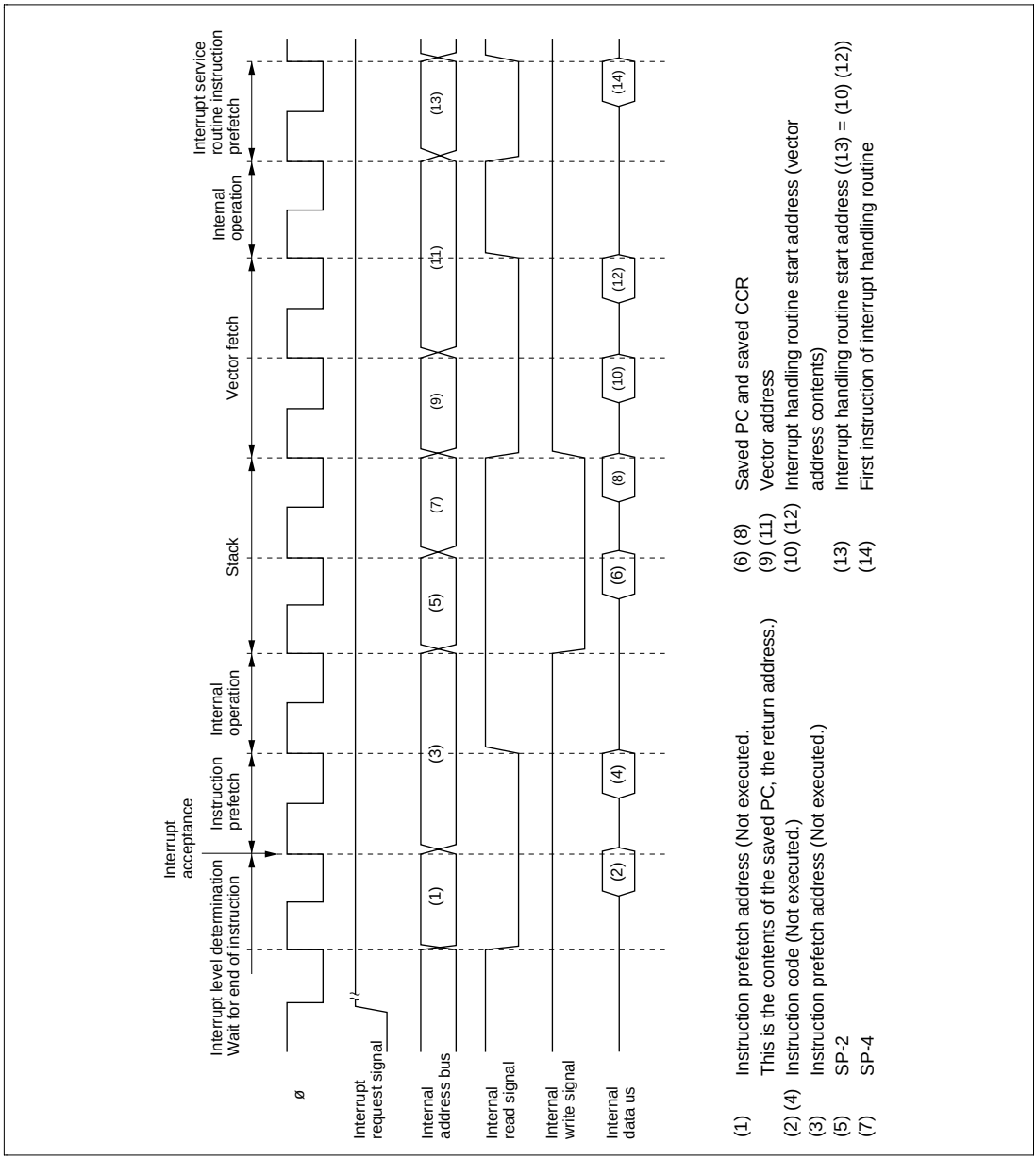


Figure 5-7 Interrupt Exception Handling

5.4.5 Interrupt Response Times

The H8S/2633 Series is capable of fast word transfer instruction to on-chip memory, and the program area is provided in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5-9 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5-9 are explained in table 5-10.

Table 5-9 Interrupt Response Times

		Normal Mode*5		Advanced Mode	
No.	**Execution Status**	**INTM1 = 0**	**INTM1 = 1**	**INTM1 = 0**	**INTM1 = 1**
1	Interrupt priority determination*1	3	3	3	3
2	Number of wait states until executing instruction ends*2	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$
3	PC, CCR, EXR stack save	$2\cdot S_K$	$3\cdot S_K$	$2\cdot S_K$	$3\cdot S_K$
4	Vector fetch	S_I	S_I	$2\cdot S_I$	$2\cdot S_I$
5	Instruction fetch*3	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$
6	Internal processing*4	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: *1 Two states in case of internal interrupt.
*2 Refers to MULXS and DIVXS instructions.
*3 Prefetch after interrupt acceptance and interrupt handling routine prefetch.
*4 Internal processing after interrupt acceptance and internal processing after vector fetch.
*5 Not available in the H8S/2633 Series.

Table 5-10 Number of States in Interrupt Handling Routine Execution Statuses

		Object of Access				
			External Device			
			8 Bit Bus		16 Bit Bus	
Symbol		Internal Memory	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	6+2m	2	3+m
Branch address read	S_J					
Stack manipulation	S_K					

Legend

m : Number of wait states in an external device access.

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

In other words, when an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared to 0.

Figure 5-8 shows an example in which the CMIEA bit in the TMR's TCR register is cleared to 0.

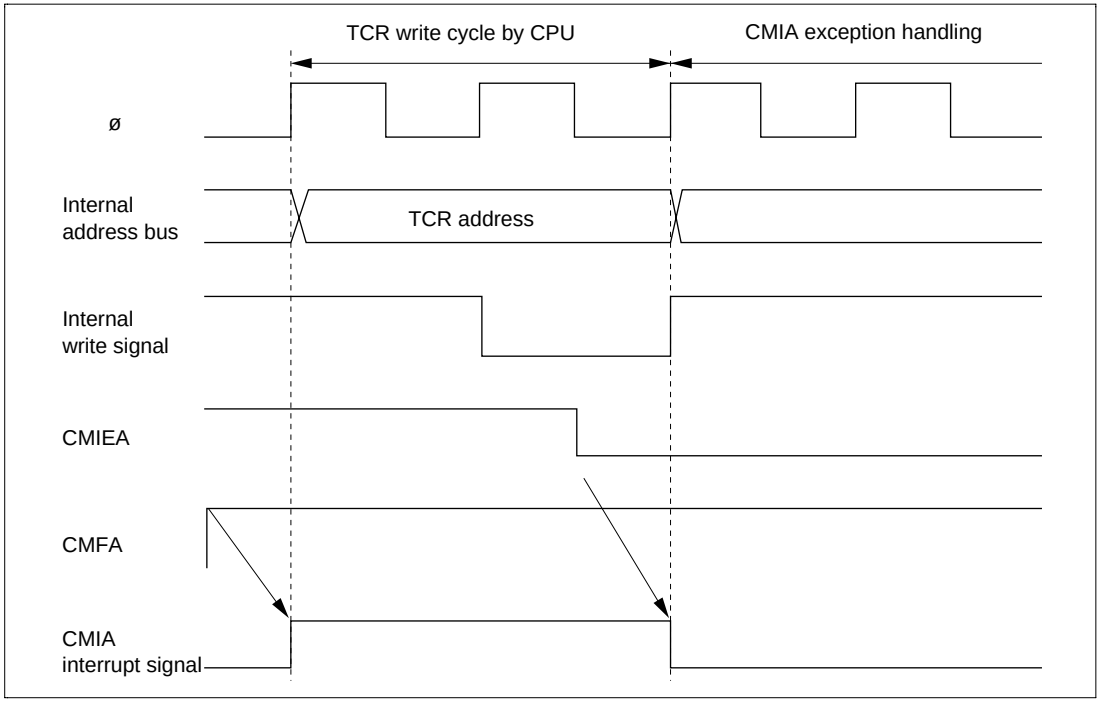


Figure 5-8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

5.5.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.5.3 Times when Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.5.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:   EEPMOV.W
      MOV.W     R4,R4
      BNE       L1
```

5.6 DTC and DMAC Activation by Interrupt

5.6.1 Overview

The DTC and DMAC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU
- Activation request to DTC
- Activation request to DMAC
- Selection of a number of the above

For details of interrupt requests that can be used with to activate the DTC and DMAC, see section 9, Data Transfer Controller and section 8, DMA Controller.

5.6.2 Block Diagram

Figure 5-9 shows a block diagram of the DTC interrupt controller.

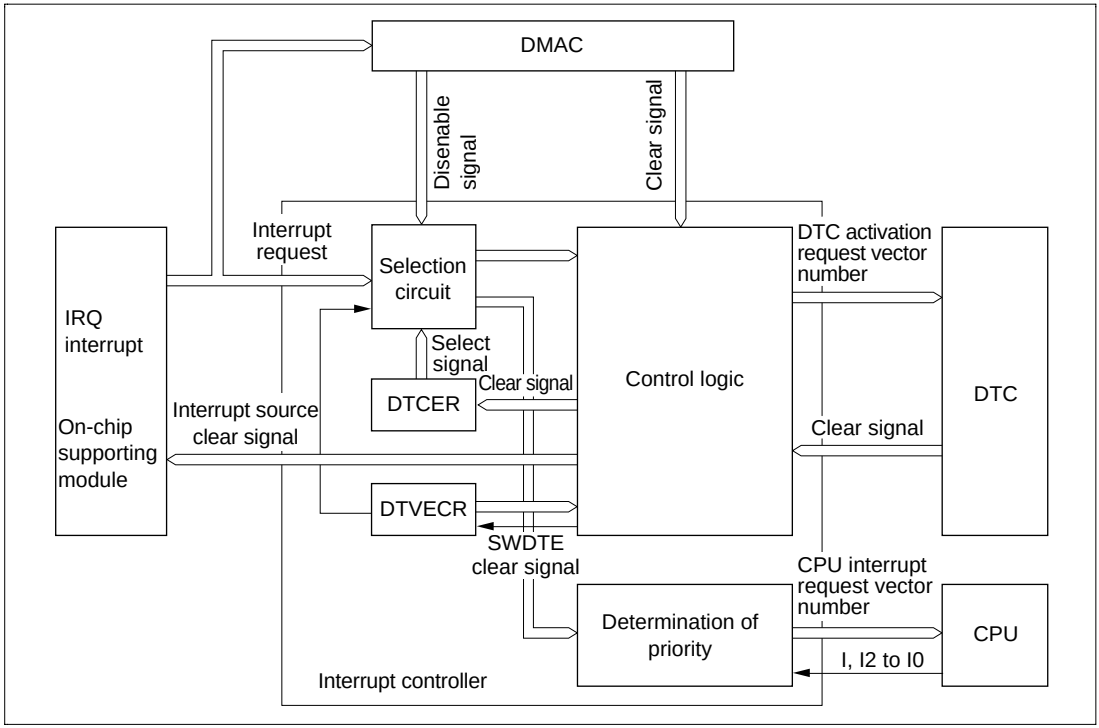


Figure 5-9 Interrupt Control for DTC and DMAC

5.6.3 Operation

The interrupt controller has three main functions in DTC and DMAC control.

(1) Selection of Interrupt Source: DMAC inputs activation factor directly to each channel. The activation factors for each channel of DMAC are selected by DTF3 to DTF0 bits of DMACR. The DTA bit of DMABCR can be used to select whether the selected activation factors are managed by DMAC. By setting the DTA bit to 1, the interrupt factor which were the activation factor for that DMAC do not act as the DTC activation factor or the CPU interrupt factor.

Interrupt factors other than the interrupts managed by the DMAC are selected as DTC activation request or CPU interrupt request by the DTCERA to DTCERF of DTC and the DTCE bit of DTCERI.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, and a CPU interrupt requested.

(2) Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See 8.6 Interrupts and 9.3.3 DTC Vector Table for the respective priority.

(3) Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as the DMAC activation factor and as the DTC activation factor or CPU interrupt factor, these operate independently. They operate in accordance with the respective operating states and bus priorities.

Table 5-11 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTA bit of DMAC's DMABCR, DTC's DTCERA to DTCERF, DTCERI's DTCE bits, and the DISEL bit of DTC's MRB.

Table 5-11 Interrupt Source Selection and Clearing Control

Settings					
DMAC	**DTC**		**Interrupt Source Selection/Clearing Control**		
DTA	**DTCE**	**DISEL**	**DMAC**	**DTC**	**CPU**
0	0	*	○	X	Δ
	1	0	○	Δ	X
		1	○	○	Δ
1	*	*	Δ	X	X

Legend

Δ : The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○ : The relevant interrupt is used. The interrupt source is not cleared.

X : The relevant bit cannot be used.

* : Don't care

(4) Notes on Use: SCI and A/D converter interrupt sources are cleared when the DMAC or DTC reads or writes to the prescribed register, and are not dependent upon the DTA, DTCE, and DISEL bits.

Section 6 PC Break Controller (PBC)

6.1 Overview

The PC break controller (PBC) provides functions that simplify program debugging. Using these functions, it is easy to create a self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator. Four break conditions can be set in the PBC: instruction fetch, data read, data write, and data read/write.

6.1.1 Features

The PC break controller has the following features:

- Two break channels (A and B)
- The following can be set as break compare conditions:
  - — 24 address bits
    Bit masking possible
  - — Bus cycle
    Instruction fetch
    Data access: data read, data write, data read/write
  - — Bus master
    Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition is as follows:
  - — Immediately before execution of the instruction fetched at the set address (instruction fetch)
  - — Immediately after execution of the instruction that accesses data at the set address (data access)
- Module stop mode can be set
  - — The initial setting is for PBC operation to be halted. Register access is enabled by clearing module stop mode.

6.1.2 Block Diagram

Figure 6-1 shows a block diagram of the PC break controller.

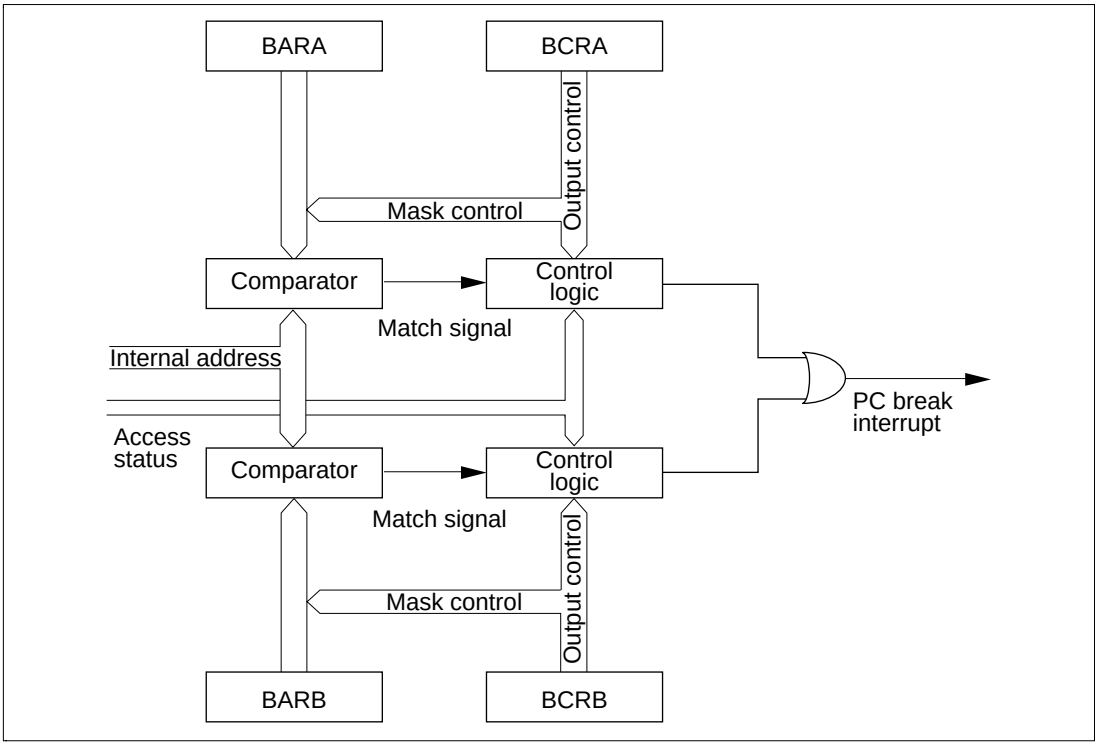


Figure 6-1 Block Diagram of PC Break Controller

6.1.3 Register Configuration

Table 6-1 shows the PC break controller registers.

Table 6-1 PC Break Controller Registers

Name	Abbreviation	R/W	Initial Value: Power-On Reset	Initial Value: Manual Reset	Address*[1]
Break address register A	BARA	R/W	H'XX000000	Retained	H'FE00
Break address register B	BARB	R/W	H'XX000000	Retained	H'FE04
Break control register A	BCRA	R/(W)*[2]	H'00	Retained	H'FE08
Break control register B	BCRB	R/(W)*[2]	H'00	Retained	H'FE09
Module stop control register C	MSTPCRC	R/W	H'FF	Retained	H'FDEA

Notes: *1. Lower 16 bits of the address.
*2. Only 0 can be written, for flag clearing.

6.2 Register Descriptions

6.2.1 Break Address Register A (BARA)

Bit	:	31	…	24	23	22	21	20	19	18	17	16	…	7	6	5	4	3	2	1	0
		—	…	—	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	…	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
Initial value	:	Undefined	…	Undefined	0	0	0	0	0	0	0	0	…	0	0	0	0	0	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

BARA is a 32-bit readable/writable register that specifies the channel A break address.

BAA23 to BAA0 are initialized to H'000000 by a power-on reset and in hardware standby mode.

Bits 31 to 24—Reserved: These bits return an undefined value if read, and cannot be modified.

Bits 23 to 0—Break Address A23 to A0 (BAA23–BAA0): These bits hold the channel A PC break address.

6.2.2 Break Address Register B (BARB)

BARB is the channel B break address register. The bit configuration is the same as for BARA.

6.2.3 Break Control Register A (BCRA)

Bit	:	7	6	5	4	3	2	1	0
		CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written, for flag clearing.

BCRA is an 8-bit readable/writable register that controls channel A PC breaks. BCRA (1) selects the break condition bus master, (2) specifies bits subject to address comparison masking, and (3) specifies whether the break condition is applied to an instruction fetch or a data access. It also contains a condition match flag.

BCRA is initialized to H'00 by a power-on reset and in hardware standby mode.

Bit 7—Condition Match Flag A (CMFA): Set to 1 when a break condition set for channel A is satisfied. This flag is not cleared to 0.

Bit 7

CMFA	Description	
0	[Clearing condition]	
	When 0 is written to CMFA after reading CMFA = 1	(Initial value)
1	[Setting condition]	
	When a condition set for channel A is satisfied	

Bit 6—CPU Cycle/DTC Cycle Select A (CDA): Selects the channel A break condition bus master.

Bit 6

CDA	Description	
0	PC break is performed when CPU is bus master	(Initial value)
1	PC break is performed when CPU or DTC is bus master	

Bits 5 to 3—Break Address Mask Register A2 to A0 (BAMRA2–BAMRA0): These bits specify which bits of the break address (BAA23–BAA0) set in BARA are to be masked.

Bit 5	Bit 4	Bit 3	
BAMRA2	**BAMRA1**	**BAMRA0**	**Description**
0	0	0	All BARA bits are unmasked and included in break conditions (Initial value)
		1	BAA0 (lowest bit) is masked, and not included in break conditions
	1	0	BAA1–0 (lower 2 bits) are masked, and not included in break conditions
		1	BAA2–0 (lower 3 bits) are masked, and not included in break conditions
1	0	0	BAA3–0 (lower 4 bits) are masked, and not included in break conditions
		1	BAA7–0 (lower 8 bits) are masked, and not included in break conditions
	1	0	BAA11–0 (lower 12 bits) are masked, and not included in break conditions
		1	BAA15–0 (lower 16 bits) are masked, and not included in break conditions

Bits 2 and 1—Break Condition Select A (CSELA1, CSELA0): These bits selection an instruction fetch, data read, data write, or data read/write cycle as the channel A break condition.

Bit 2	Bit 1		
CSELA1	**CSELA0**	**Description**	
0	0	Instruction fetch is used as break condition	(Initial value)
	1	Data read cycle is used as break condition	
1	0	Data write cycle is used as break condition	
	1	Data read/write cycle is used as break condition	

Bits 0—Break Interrupt Enable A (BIEA): Enables or disables channel A PC break interrupts.

Bit 0		
BIEA	**Description**	
0	PC break interrupts are disabled	(Initial value)
1	PC break interrupts are enabled	

6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for BCRA.

6.2.5 Module Stop Control Register C (MSTPCRC)

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	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

MSTPCRC is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPC4 bit is set to 1, PC break controller operation is stopped at the end of the bus cycle, and module stop mode is entered. Register read/write accesses are not possible in module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a power on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 4—Module Stop (MSTPC4): Specifies the PC break controller module stop mode.

Bit 4	
MSTPC4	**Description**
0	PC break controller module stop mode is cleared
1	PC break controller module stop mode is set (Initial value)

6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling is shown in sections 6.3.1 and 6.3.2, taking the example of channel A.

6.3.1 PC Break Interrupt Due to Instruction Fetch

(1) Initial settings

— Set the break address in BARA. For a PC break caused by an instruction fetch, set the address of the first instruction byte as the break address.

— Set the break conditions in BCRA.

BCRA bit 6 (CDA): With a PC break caused by an instruction fetch, the bus master must be the CPU. Set 0 to select the CPU.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2–1 (CSELA1–0): Set 00 to specify an instruction fetch as the break condition.

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— When the instruction at the set address is fetched, a PC break request is generated immediately before execution of the fetched instruction, and the condition match flag (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.2 PC Break Interrupt Due to Data Access

(1) Initial settings

— Set the break address in BARA. For a PC break caused by a data access, set the target ROM, RAM, I/O, or external address space address as the break address. Stack operations and branch address reads are included in data accesses.

— Set the break conditions in BCRA.

BCRA bit 6 (CDA): Select the bus master.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2–1 (CSELA1–0): Set 01, 10, or 11 to specify data access as the break condition.

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— After execution of the instruction that performs a data access on the set address, a PC break request is generated and the condition match flag (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.3 Notes on PC Break Interrupt Handling

(1) The PC break interrupt is shared by channels A and B. The channel from which the request was issued must be determined by the interrupt handler.

(2) The CMFA and CMFB flags are not cleared to 0, so 0 must be written to CMFA or CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another interrupt will be requested after interrupt handling ends.

(3) A PC break interrupt generated when the DTC is the bus master is accepted after the bus has been transferred to the CPU by the bus controller.

6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address after a SLEEP instruction is shown below.

(1) When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to sleep mode, or from subactive mode to subsleep mode:

After execution of the SLEEP instruction, a transition is not made to sleep mode or subsleep mode, and PC break interrupt handling is executed. After execution of PC break interrupt handling, the instruction at the address after the SLEEP instruction is executed (figure 6-2 (A)).

(2) When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to subactive mode:

After execution of the SLEEP instruction, a transition is made to subactive mode via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6-2 (B)).

(3) When the SLEEP instruction causes a transition from subactive mode to high-speed (medium-speed) mode:

After execution of the SLEEP instruction, and following the clock oscillation settling time, a transition is made to high-speed (medium-speed) mode via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6-2 (C)).

(4) When the SLEEP instruction causes a transition to software standby mode or watch mode:

After execution of the SLEEP instruction, a transition is made to the respective mode, and PC break interrupt handling is not executed. However, the CMFA or CMFB flag is set (figure 6-2 (D)).

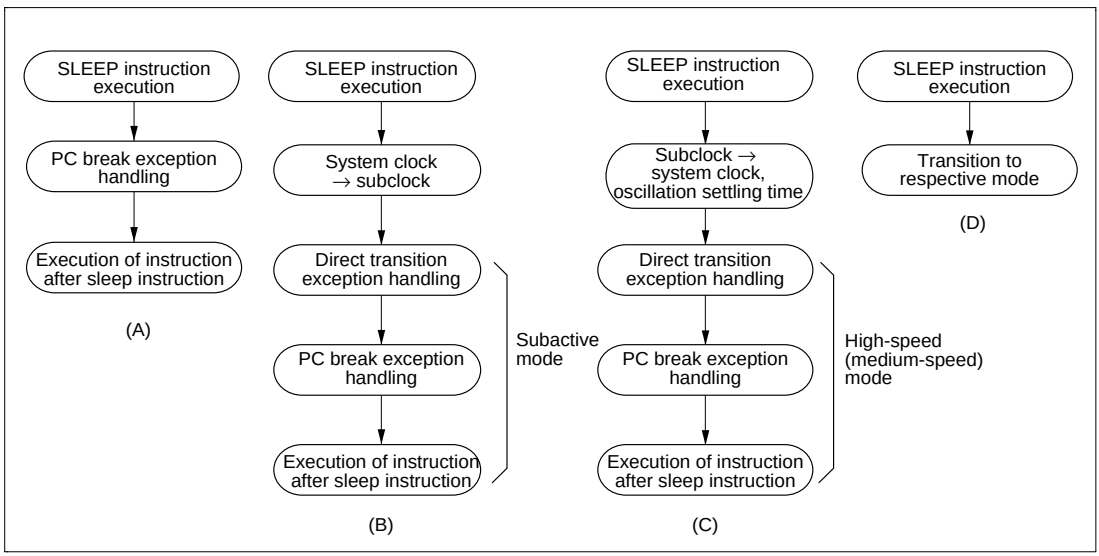


Figure 6-2 Operation in Power-Down Mode Transitions

6.3.5 PC Break Operation in Continuous Data Transfer

If a PC break interrupt is generated when the following operations are being performed, exception handling is executed on completion of the specified transfer.

(1) When a PC break interrupt is generated at the transfer address of an EEPMOV.B instruction:

PC break exception handling is executed after all data transfers have been completed and the EEPMOV.B instruction has ended.

(2) When a PC break interrupt is generated at a DTC transfer address:

PC break exception handling is executed after the DTC has completed the specified number of data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

6.3.6 When Instruction Execution is Delayed by One State

Caution is required in the following cases, as instruction execution is one state later than usual.

(1) When the PBC is enabled (i.e. when the break interrupt enable bit is set to 1), execution of a one-word branch instruction (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, or RTS) located in on-chip ROM or RAM is always delayed by one state.

(2) When break interruption by instruction fetch is set, the set address indicates on-chip ROM or RAM space, and that address is used for data access, the instruction that executes the data access is one state later than in normal operation.

(3) When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction has one of the addressing modes shown below, and that address indicates on-chip ROM or RAM, and that address is used for data access, the instruction will be one state later than in normal operation.
@ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8

(4) When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction is NOP or SLEEP, or has #xx,Rn as its addressing mode, and that instruction is located in on-chip ROM or RAM, the instruction will be one state later than in normal operation.

6.3.7 Additional Notes

(1) When a PC break is set for an instruction fetch at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction:

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

(2) When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the executing instruction. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XORC, the next instruction is always executed. For details, see section 5, Interrupt Controller.

(3) When a PC break is set for an instruction fetch at the address following a Bcc instruction:

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, but is not generated if the instruction at the next address is not executed.

(4) When a PC break is set for an instruction fetch at the branch destination address of a Bcc instruction:

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, but is not generated if the instruction at the branch destination is not executed.

Section 7 Bus Controller

7.1 Overview

The H8S/2633 Series has a built-in bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

The bus controller also has a bus arbitration function, and controls the operation of the internal bus masters: the CPU, DMA controller (DMAC), and data transfer controller (DTC).

7.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - Manages the external space as 8 areas of 2-Mbytes
  - Bus specifications can be set independently for each area
  - DRAM/Burst ROM interface can be set
- Basic bus interface
  - Chip selects ($\overline{CS0}$ to $\overline{CS7}$) can be output for areas 0 to 7.
  - 8-bit access or 16-bit access can be selected for each area
  - 2-state access or 3-state access can be selected for each area
  - Program wait states can be inserted for each area
- DRAM interface
  - DRAM interface can be set for areas 2 to 5 (in advanced mode);
  - Multiplexed output of row and column addresses (8/9/10-bit);
  - 2 CAS method;
  - Burst operation (in high-speed mode);
  - Insertion of T_p cycle to secure RAS precharge time;
  - Selection of CAS-before-RAS refresh and self refresh.
- Burst ROM interface
  - Burst ROM interface can be set for area 0
  - Choice of 1- or 2-state burst access

- Idle cycle insertion
  - An idle cycle can be inserted in case of an external read cycle between different areas
  - An idle cycle can be inserted in case of an external write cycle immediately after an external read cycle
- Write buffer functions
  - External write cycle and internal access can be executed in parallel
  - DMAC single-address mode and internal access can be executed in parallel.
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC and DTC
- Other features
  - Refresh counter (refresh timer) can be used as an interval timer.
  - External bus release function

7.1.2 Block Diagram

Figure 7-1 shows a block diagram of the bus controller.

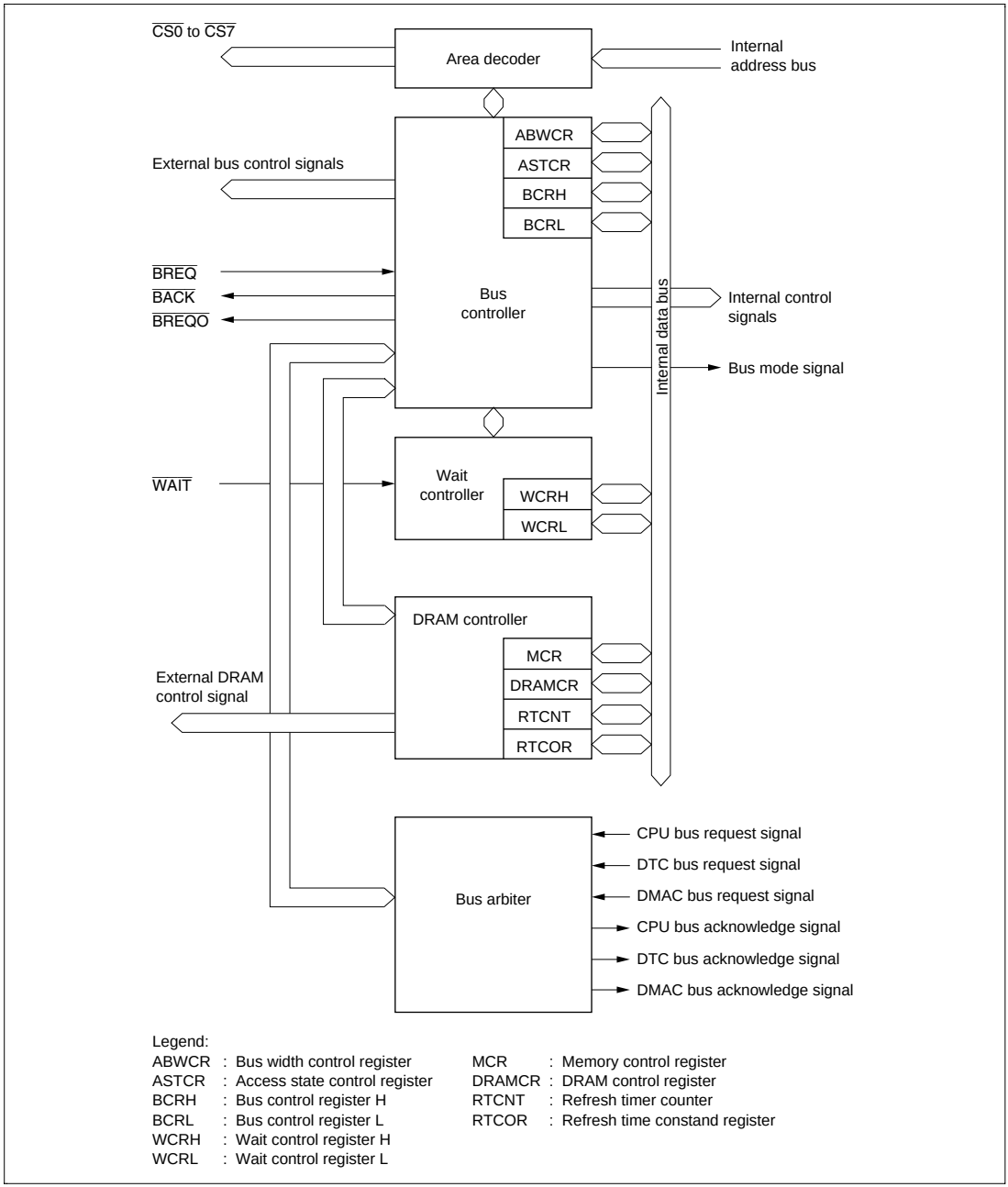


Figure 7-1 Block Diagram of Bus Controller

7.1.3 Pin Configuration

Table 7-1 summarizes the pins of the bus controller.

Table 7-1 Bus Controller Pins

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that external space is being read.
High write/ write enable	$\overline{HWR}$	Output	Strobe signal indicating that external space is to be written, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is to be written, and lower half (D7 to D0) of data bus is enabled.
Chip select 0	$\overline{CS0}$	Output	Strobe signal showing selection of area 0
Chip select 1	$\overline{CS1}$	Output	Strobe signal showing selection of area 1
Chip select 2/row address strobe 2	$\overline{CS2}$	Output	Strobe signal showing selection of area 2. When area 2 is allocated to DRAM space, this is the row address strobe signal for DRAM. When areas 2 to 5 are contiguous DRAM space, this is the row address strobe signal for DRAM.
Chip select 3/row address strobe 3	$\overline{CS3}/\overline{OE}$	Output	Strobe signal showing selection of area 3. When area 3 is allocated to DRAM space, this is the row address strobe signal for DRAM. When only area 2 is allocated to DRAM space, or when areas 2 to 5 are contiguous DRAM space, this is output enable signal.
Chip select 4/row address strobe 4	$\overline{CS4}$	Output	Strobe signal showing selection of area 4. When area 4 is allocated to DRAM space, this is the row address strobe signal for DRAM.
Chip select 5/row address strobe 5	$\overline{CS5}$	Output	Strobe signal showing selection of area 5. When area 5 is allocated to DRAM space, this is the row address strobe signal for DRAM.
Chip select 6	$\overline{CS6}$	Output	Strobe signal showing selection of area 6.
Chip select 7	$\overline{CS7}$	Output	Strobe signal showing selection of area 7.
Upper column address strobe	$\overline{CAS}$	Output	2 CAS method DRAM upper column address strobe signal
Lower column strobe	$\overline{LCAS}$	Output	DRAM lower column address strobe signal

Name	Symbol	I/O	Function
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external 3-state access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bus to external device.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released.
Bus request output	$\overline{BREQO}$	Output	External bus request signal used when internal bus master accesses external space when external bus is released.

7.1.4 Register Configuration

Table 7-2 summarizes the registers of the bus controller.

Table 7-2 Bus Controller Registers

			Initial Value		
Name	Abbreviation	R/W	Power-On Reset	Manual Reset	Address*[1]
Bus width control register	ABWCR	R/W	H'FF/H'00*[2]	Retained	H'FED0
Access state control register	ASTCR	R/W	H'FF	Retained	H'FED1
Wait control register H	WCRH	R/W	H'FF	Retained	H'FED2
Wait control register L	WCRL	R/W	H'FF	Retained	H'FED3
Bus control register H	BCRH	R/W	H'D0	Retained	H'FED4
Bus control register L	BCRL	R/W	H'08	Retained	H'FED5
Pin function control register	PFCR	R/W	H'0D/H'00	Retained	H'FDEB
Memory control register	MCR	R/W	H'00	Retained	H'FED6
DRAM control register	DRAMCR	R/W	H'00	Retained	H'FED7
Refresh timer counter	RTCNT	R/W	H'00	Retained	H'FED8
Refresh time constant register	RTCOR	R/W	H'FF	Retained	H'FED9

Notes: *1 Lower 16 bits of the address.
*2 Determined by the MCU operating mode.

7.2 Register Descriptions

7.2.1 Bus Width Control Register (ABWCR)

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7									
Initial value	:	1	1	1	1	1	1	1	1
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4									
Initial value	:	0	0	0	0	0	0	0	0
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABWCR is an 8-bit readable/writable register that designates each area for either 8-bit access or 16-bit access.

ABWCR sets the data bus width for the external memory space. The bus width for on-chip memory and internal I/O registers is fixed regardless of the settings in ABWCR.

In normal mode, the settings of bits ABW7 to ABW1 have no effect on operation.

After a power-on reset and in hardware standby mode, ABWCR is initialized to H'FF in modes 5 to 7, and to H'00 in mode 4. It is not initialized by a manual reset or in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select whether the corresponding area is to be designated for 8-bit access or 16-bit access.

Bit n	
ABWn	**Description**
0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

(n = 7 to 0)

7.2.2 Access State Control Register (ASTCR)

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	:	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

ASTCR is an 8-bit readable/writable register that designates each area as either a 2-state access space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of access states for on-chip memory and internal I/O registers is fixed regardless of the settings in ASTCR.

In normal mode, the settings of bits AST7 to AST1 have no effect on operation.

ASTCR is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is to be designated as a 2-state access space or a 3-state access space.

Wait state insertion is enabled or disabled at the same time.

Bit n ASTn	Description	
0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled	
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled	(Initial value)

(n = 7 to 0)

7.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

Program waits are not inserted in the case of on-chip memory or internal I/O registers.

WCRH and WCRL are initialized to H'FF by a power-on reset and in hardware standby mode. They are not initialized by a manual reset or in software standby mode.

(1) WCRH

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	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

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W61	**W60**	**Description**
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

(2) WCRL

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	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

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed (Initial value)

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed (Initial value)

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W11	**W10**	**Description**
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed (Initial value)

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed (Initial value)

7.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMTS0
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cycle insertion, and the memory interface for area 0.

BCRH is initialized to H'D0 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read cycles are performed in different areas.

Bit 7

ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas (Initial value)

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read and external write cycles are performed .

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles (Initial value)

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst ROM interface.

Bit 5

BRSTRM	Description
0	Area 0 is basic bus interface (Initial value)
1	Area 0 is burst ROM interface

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycles for the burst ROM interface.

Bit 4	
BRSTS1	**Description**
0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states (Initial value)

Bit 3—Burst Cycle Select 0 (BRSTS0): Selects the number of words that can be accessed in a burst ROM interface burst access.

Bit 3	
BRSTS0	**Description**
0	Max. 4 words in burst access (Initial value)
1	Max. 8 words in burst access

Bits 2 to 0—RAM Type Select (RMTS2 to RMTS0): In advanced mode, these bits select the memory interface for areas 2 to 5.

When DRAM space is selected, the appropriate area becomes the DRAM interface.

Bit 2	Bit 1	Bit 0	Description			
RMTS2	**RMTS1**	**RMTS0**	**Area 5**	**Area 4**	**Area 3**	**Area 2**
0	0	0	Normal space	Normal space	Normal space	Normal space
		1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space
1	1	1	Contiguous DRAM space	Contiguous DRAM space	Contiguous DRAM space	Contiguous DRAM space

Note: When all areas selected in DRAM are 8-bit space, the PF2 pin can be used as an I/O port and for $\overline{\text{BREQO}}$ and $\overline{\text{WAIT}}$. When contiguous RAM is selected set the appropriate bus width and number of access states (the number of programmable waits) to the same values for all of areas 2 to 5. Do not set other than the above combinations.

7.2.5 Bus Control Register L (BCRL)

Bit	:	7	6	5	4	3	2	1	0
		BRLE	BREQOE	—	OES	DDS	RCTS	WDBE	WAITE
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W

BCRL is an 8-bit readable/writable register that performs selection of the external bus-released state protocol, enabling or disabling of the write data buffer function, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

BCRL is initialized to H'08 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bit 7—Bus Release Enable (BRLE): Enables or disables external bus release.

Bit 7

BRLE	Description
0	External bus release is disabled. $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ and $\overline{\text{BREQO}}$ can be used as I/O ports. (Initial value)
1	External bus release is enabled.

Bit 6—BREQO Pin Enable (BREQOE): Outputs a signal that requests the external bus master to drop the bus request signal ($\overline{\text{BREQ}}$) in the external bus release state, when an internal bus master performs an external space access, or when a refresh request is generated.

Bit 6

BREQOE	Description	
0	$\overline{\text{BREQO}}$ output disabled. $\overline{\text{BREQO}}$ can be used as I/O port.	(Initial value)
1	$\overline{\text{BREQO}}$ output enabled.	

Bit 5—Reserved: This bit cannot be modified and is always read as 0.

Bit 4—OE Select (OES): Selects the $\overline{\text{CS3}}$ pin as the $\overline{\text{OE}}$ pin.

Bit 4

OES	Description	
0	Uses the $\overline{\text{CS3}}$ pin as the port or as $\overline{\text{CS3}}$ signal output	(Initial value)
1	When only area 2 is set for DRAM, or when areas 2 to 5 are set as contiguous DRAM space, the $\overline{\text{CS3}}$ pin is used as the $\overline{\text{OE}}$ pin.	

Bit 3—DACK Timing Select (DDS): When using the DRAM interface, this bit selects the DMAC single address transfer bus timing.

Bit 3

DDS	Description
0	When performing DMAC single address transfers to DRAM, always execute full access. The $\overline{DACK}$ signal is output as a low-level signal from the T_r or T_1 cycle.
1	Burst access is also possible when performing DMAC single address tranfers to DRAM. The $\overline{DACK}$ signal is output as a low-level signal from the T_{C1} or T_2 cycle. (Initial value)

Bit 2—Read CAS Timing Select (RCTS): Selects the $\overline{CAS}$ signal output timing.

Bit 2

RCTS	Description
0	$\overline{CAS}$ signal output timing is same when reading and writing. (Initial value)
1	When reading, $\overline{CAS}$ signal is asserted half cycle earlier than when writing.

Bit 1—Write Data Buffer Enable (WDBE): This bit selects whether or not to use the write buffer function in the external write cycle or the DMAC single address cycle.

Bit 1

WDBE	Description
0	Write data buffer function not used (Initial value)
1	Write data buffer function used

Bit 0—WAIT Pin Enable (WAITE): Selects enabling or disabling of wait input by the $\overline{WAIT}$ pin.

Bit 0

WAITE	Description
0	Wait input by $\overline{WAIT}$ pin disabled. $\overline{WAIT}$ pin can be used as I/O port. (Initial value)
1	Wait input by $\overline{WAIT}$ pin enabled

7.2.6 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		CSS07	CSS36	BUZZE	LCASS	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit read/write register that controls the CS selection of pins PG4 and PG1, controls LCAS selection of pins PF2 and PF6, and controls the address output in expanded mode with ROM.

PFCR is initialized to H'0D/H'00 by a power-on reset and in hardware standby mode. It retains its previous state by a manual reset or in software standby mode.

Bit 7—$\overline{\text{CS0}}$/$\overline{\text{CS7}}$ Select (CSS07): This bit selects the contents of CS output via the PG4 pin. In modes 4, 5, and 6, setting the corresponding DDR to 1 outputs the selected CS.

Bit 7

CSS07	Description	
0	Selects $\overline{\text{CS0}}$.	(Initial value)
1	Selects $\overline{\text{CS7}}$.	

Bit 6—$\overline{\text{CS3}}$/$\overline{\text{CS6}}$ Select (CSS36): This bit selects the contents of CS output via the PG1 pin. In modes 4, 5, and 6, setting the corresponding DDR to 1 outputs the selected CS.

Bit 6

CSS36	Description	
0	Selects $\overline{\text{CS3}}$.	(Initial value)
1	Selects $\overline{\text{CS6}}$.	

Bit 5—BUZZ Output Enable (BUZZE): This bit enables/disables BUZZ output via the PF1 pin. The WDT1 input clock, selected with PSS and CKS2 to CKS0, is output as the BUZZ signal. See Section 15.2.4, Pin Function Control Register (PFCR) for details of BUZZ output.

Bit 5		
BUZZE	**Description**	
0	Functions as PF1 input pin.	(Initial value)
1	Functions as BUZZ output pin.	

Bit 4—LCAS Output Pin Select Bit (LCASS): Selects output pin for LCAS signal.

Bit 4		
LCASS	**Description**	
0	Outputs LCAS signal from PF2.	(Initial value)
1	Outputs LCAS signal from PF6.	

Bits 3 to 0—Address Output Enable 3 to 0 (AE3–AE0): These bits select enabling or disabling of address outputs A8 to A23 in ROMless expanded mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1.

Bit 3	Bit 2	Bit 1	Bit 0	
AE3	**AE2**	**AE1**	**AE0**	**Description**
0	0	0	0	A8–A23 address output disabled (Initial value*)
			1	A8 address output enabled; A9–A23 address output disabled
		1	0	A8, A9 address output enabled; A10–A23 address output disabled
			1	A8–A10 address output enabled; A11–A23 address output disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address output disabled
			1	A8–A12 address output enabled; A13–A23 address output disabled
		1	0	A8–A13 address output enabled; A14–A23 address output disabled
			1	A8–A14 address output enabled; A15–A23 address output disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address output disabled
			1	A8–A16 address output enabled; A17–A23 address output disabled
		1	0	A8–A17 address output enabled; A18–A23 address output disabled
			1	A8–A18 address output enabled; A19–A23 address output disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address output disabled
			1	A8–A20 address output enabled; A21–A23 address output disabled (Initial value*)
		1	0	A8–A21 address output enabled; A22, A23 address output disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

7.2.7 Memory Control Register (MCR)

Bit	:	7	6	5	4	3	2	1	0
		TPC	BE	RCDM	CW2	MXC1	MXC0	RLW1	RLW0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The MCR is an 8-bit read/write register that, when areas 2 to 5 are set as the DRAM interface, controls the DRAM strobe method, number of precharge cycles, access mode, address multiplex shift amount, and number of wait states to be inserted when a refresh is performed.

The MCR is initialized to H'00 at a power-on reset and in hardware standby mode. It is not initialized at a manual reset or in software standby mode.

Bit 7—TP Cycle Control (TPC): When accessing areas 2 to 5, allocated to DRAM, this bit selects whether the precharge cycle (T_P) is 1 state or 2 states.

Bit 7

TPC	Description	
0	Insert 1 precharge cycle.	(Initial value)
1	Insert 2 precharge cycles.	

Bit 6—Burst Access Enable (BE): This bit enables/disables burst access of areas 2 to 5, allocated as DRAM space. DRAM space burst access is in high-speed page mode. When using EDO type in this case, either select OE output or RAS up mode.

Bit 6

BE	Description	
0	Burst disabled (always full access).	(Initial value)
1	Access DRAM space in high-speed page mode.	

Bit 5—RAS Down Mode (RCDM): When areas 2 to 5 are allocated to DRAM space, this bit selects whether the $\overline{\text{RAS}}$ signal level remains Low while waiting for the next DRAM access (RAS down mode) or the $\overline{\text{RAS}}$ signal level returns to High (RAS up mode), when DRAM access is discontinued.

Bit 5

RCDM	Description	
0	DRAM interface: selects RAS up mode.	(Initial value)
1	DRAM interface: selects RAS down mode.	

Bit 4—Reserved (CW2): Only write 0 to this bit.

Bits 3 and 2—Multiplex shift counts 1 and 0 (MXC1 and MXC0): These bits select the shift amount to the low side of the row address of the multiplexed row/column address in DRAM interface mode. They also select the row address to be compared in burst operation of the DRAM interface.

Bit 3	Bit 2	
MXC1	**MXC0**	**Description**
0	0	8-bit shift (Initial value) (1) 8-bit access space: target row addresses for comparison are A23 to A8 (2) 16-bit access space: target row addresses for comparison are A23 to A9
	1	9-bit shift (1) 8-bit access space: target row addresses for comparison are A23 to A9 (2) 16-bit access space: target row addresses for comparison are A23 to A10
1	0	10-bit shift (1) 8-bit access space: target row addresses for comparison are A23 to A10 (2) 16-bit access space: target row addresses for comparison are A23 to A11
	1	—

Bits 1 and 0—Refresh Cycle Wait Control 1 and 0 (RLW1 and RLW0): These bits select the number of wait states to be inserted in the CAS-before-RAS refresh cycle of the DRAM interface. The selected number of wait states is applied to all areas set as DRAM space. Wait input via the $\overline{\text{WAIT}}$ pin is disabled.

Bit 1	Bit 0	
RLW1	**RLW0**	**Description**
0	0	Do not insert wait state. (Initial value)
	1	Insert 1 wait state
1	0	Insert 2 wait states
	1	Insert 3 wait states

7.2.8 DRAM Control Register (DRAMCR)

Bit	:	7	6	5	4	3	2	1	0
		RFSHE	CBRM	RMODE	CMF	CMIE	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DRAMCR is an 8-bit read/write register that selects DRAM refresh mode, the refresh counter clock, and sets the refresh timer control.

The DRAMCR is initialized to H'00 at a power-on reset and in hardware standby mode. It is not initialized at a manual reset or in software standby mode.

Bit 7—Refresh Control (RFSHE): This bit selects whether or not to perform refresh control. When not performing refresh control, the refresh timer can be used as an interval timer.

Bit 7

RFSHE	Description	
0	Do not perform refresh control.	(Initial value)
1	Perform refresh control.	

Bit 6—CBR Refresh Mode (CBRM): This bit selects whether CBR refresh is performed in parallel with other external access, or only CBR refresh is performed.

Bit 6

CBRM	Description	
0	Enables external access during CAS-before-RAS refresh.	(Initial value)
1	Disables external access during CAS-before-RAS refresh.	

Bit 5—Refresh Mode (RMODE): This bit selects whether or not to perform a self refresh in software standby mode when performing refresh control (RFSHE=1).

Bit 5

RMODE	Description	
0	Do not perform self-refresh in software standby mode.	(Initial value)
1	Perform self-refresh in software standby mode.	

Bit 4—Compare Match Flag (CMF): This status flag shows a match between RTCNT and RTCOR values.

When performing refresh control (RFSHE=1), write 1 to CMF when writing to the DRAMCR.

Bit 4		
CMF	**Description**	
0	[Clearing]	
	When CMF=1, read the CMF flag, then clear the CMF flag to 0.	(Initial value)
1	[Setting]	
	CMF is set when RTCNT=RTCOR.	

Bit 3—Compare Match Interrupt Enable (CMIE): This bit enables/disables the CMF flag interrupt request (CMI) when the DRAMCR CMF flag is set to 1.

CMIE is always 0 when performing a self-refresh.

Bit 3		
CMIE	**Description**	
0	Disables CMF flag interrupt requests (CMI).	(Initial value)
1	Enables CMF flag interrupt requests (CMI).	

Bits 2 to 0—Refresh Counter Clock Select (CKS2 to CKS0): These bits select from the seven internal clocks derived by dividing the system clock (ø) to be input to RTCNT. The RTCNT count up starts when CKS2 to CKS0 are set to select the input clock.

Bit 2	Bit 1	Bit 0		
CKS2	**CKS1**	**CKS0**	**Description**	
0	0	0	Stops count.	(Initial value)
		1	Counts on ø/2	
	1	0	Counts on ø/8	
		1	Counts on ø/32	
1	0	0	Counts on ø/128	
		1	Counts on ø/512	
	1	0	Counts on ø/2048	
		1	Counts on ø/4096	

7.2.9 Refresh Timer Counter (RTCNT)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCNT is an 8-bit read/write up-counter.

RTCNT counts up using the internal clock selected by the DRAMCR CKS2 to CKS0 bits.

When RTCNT matches the value in RTCOR (compare match), the DRAMCR CMF flag is set to 1 and RTCNT is cleared to H'00. If, at this point, DRAMCR RFSHE is set to 1, the refresh cycle starts. When the DRAMCR CMIE bit is set to 1, a compare match interrupt (CMI) is also generated.

RTCNT is initialized to H'00 at a power-on reset and in hardware standby mode. It is not initialized at a manual reset or in software standby mode.

7.2.10 Refresh Time Constant Register (RTCOR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

RTCOR is an 8-bit read/write register that sets theRTCNT compare match cycle.

The values of RTCOR and RTCNT are constantly compared and, when both value match, the DRAMCR CMF flag is set to 1 and RTCNT is cleared to H'00.

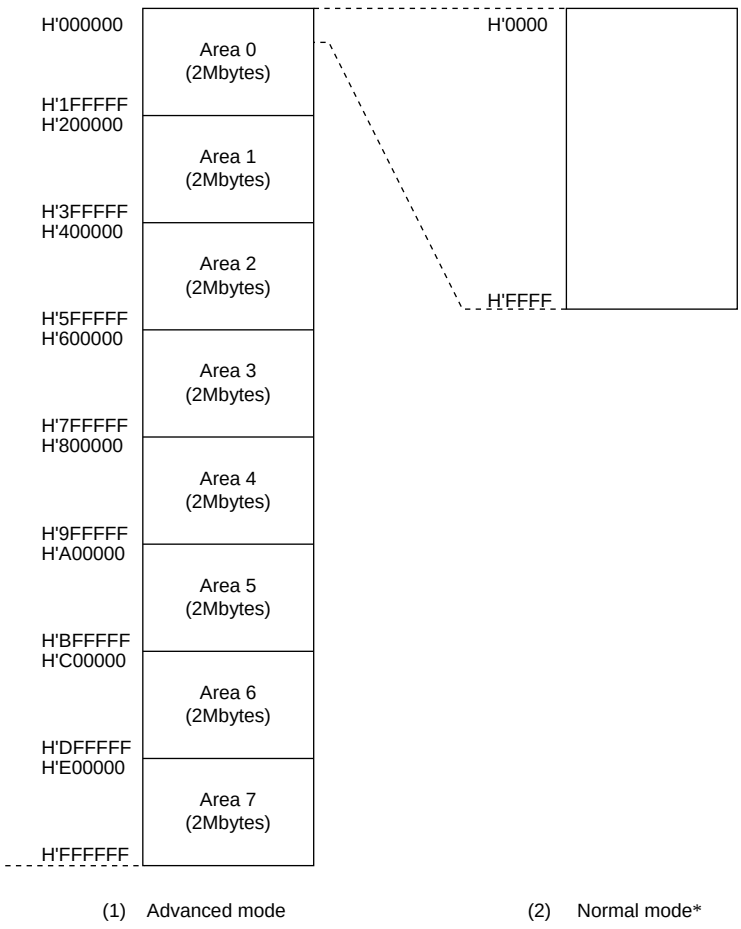
RTCOR is initialized to H'FF at a power-on reset and in hardware standby mode. It is not initialized at a manual reset or in software standby mode.

7.3 Overview of Bus Control

7.3.1 Area Partitioning

In advanced mode, the bus controller partitions the 16 Mbytes address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external space in area units. A chip select signal ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area. In normal mode*, it controls a 64-kbyte address space comprising part of area 0. Figure 7-2 shows an outline of the memory map.

Note: * Not available in the H8S/2633 Series.



Note: * Not available in the H8S/2633 Series.

Figure 7-2 Overview of Area Partitioning

7.3.2 Bus Specifications

The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

(1) Bus Width: A bus width of 8 or 16 bits can be selected with ADWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bus mode is always set.

(2) Number of Access States: Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space.

With the DRAM interface or the burst ROM interface, the number of access states may be determined without regard to ASTCR.

When 2-state access space is designated, wait insertion is disabled.

(3) Number of Program Wait States: When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 7-3 shows the bus specifications for each basic bus interface area.

Table 7-3 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

7.3.3 Memory Interfaces

The H8S/2633 Series memory interfaces comprise a basic bus interface that allows direct connection or ROM, SRAM, and so on, DRAM interface with direct DRAM connection and a burst ROM interface that allows direct connection of burst ROM. The memory interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, and areas set for DRAM interface are DRAM spaces an area for which the burst ROM interface is designated functions as burst ROM space.

7.3.4 Interface Specifications for Each Area

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode. The bus specifications described here cover basic items only, and the sections on each memory interface (7.4, 7.5 and 7.7) should be referred to for further details.

Area 0: Area 0 includes on-chip ROM, and in ROM-disabled expansion mode, all of area 0 is external space. In ROM-enabled expansion mode, the space excluding on-chip ROM is external space.

A $\overline{\text{CS0}}$ signal can be output when accessing area 0 external space.

Either basic bus interface or burst ROM interface can be selected for area 0.

Areas 1 and 6: In external expansion mode, all of areas 1 and 6 is external space.

$\overline{\text{CS1}}$ and $\overline{\text{CS6}}$ pin signals can be output when accessing the area 1 and 6 external space.

Only the basic bus interface can be used for areas 1 and 6.

Areas 2 to 5: In external expansion mode, all of areas 2 to 5 is external space.

$\overline{\text{CS2}}$ to $\overline{\text{CS5}}$ signals can be output when accessing area 2 to 5 external space.

The standard bus interface or DRAM interface can be selected for areas 2 to 5. In DRAM interface mode, signals $\overline{\text{CS2}}$ to $\overline{\text{CS5}}$ are used as $\overline{\text{RAS}}$ signals.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In external expansion mode, the space excluding the on-chip RAM and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

A $\overline{\text{CS7}}$ signal can be output when accessing area 7 external space.

Only the basic bus interface can be used for the area 7.

7.3.5 Chip Select Signals

This LSI allows chip select signals ($\overline{CS0}$ to $\overline{CS7}$) to be output for each of areas 0 to 7. The level of these signals is set Low when accessing the external space of the respective area.

Figure 7-3 shows example $\overline{CSn}$ (where n=0 to 7) signal output timing.

The output of the $\overline{CSn}$ signal can be enabled or disabled by the data direction register (DDR) of the port of the corresponding $\overline{CSn}$ pin.

In ROM-disabled expanded mode, the $\overline{CS0}$ pin is set for output after a power-on reset. The $\overline{CS1}$ to $\overline{CS7}$ pins are set for input after a power-on reset, so the corresponding DDR must be set to 1 to allow the output of $\overline{CS1}$ to $\overline{CS7}$ signals.

In ROM-disabled expanded mode, all of pins $\overline{CS0}$ to $\overline{CS7}$ are set for input after a power-on reset, so the corresponding DDR must be set to 1 to allow the output of $\overline{CS0}$ to $\overline{CS7}$ signals.

See Section 10, I/O Ports for details.

When areas 2 to 5 are set as DRAM space, $\overline{CS2}$ to $\overline{CS5}$ outputs are used as $\overline{RAS}$ signals.

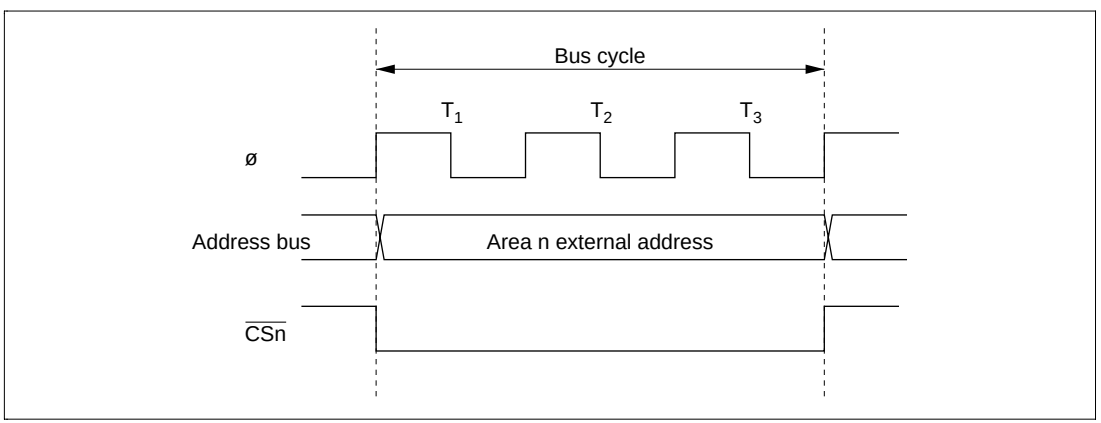


Figure 7-3 $\overline{CSn}$ Signal Output Timing (where n=0 to 7)

7.4 Basic Bus Interface

7.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 7-3).

7.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 7-4 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two byte accesses, and a longword transfer instruction, as four byte accesses.

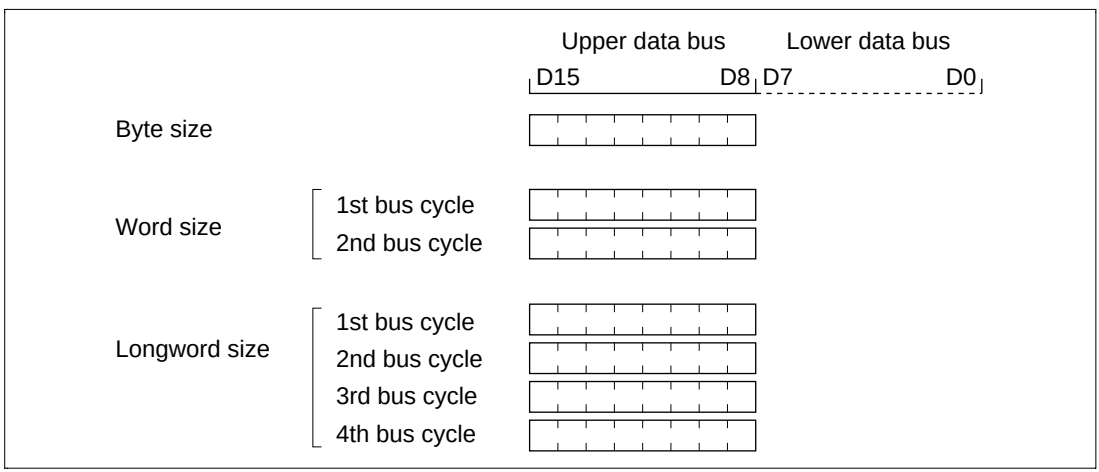


Figure 7-4 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 7-5 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword transfer instruction is executed as two word transfer instructions.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

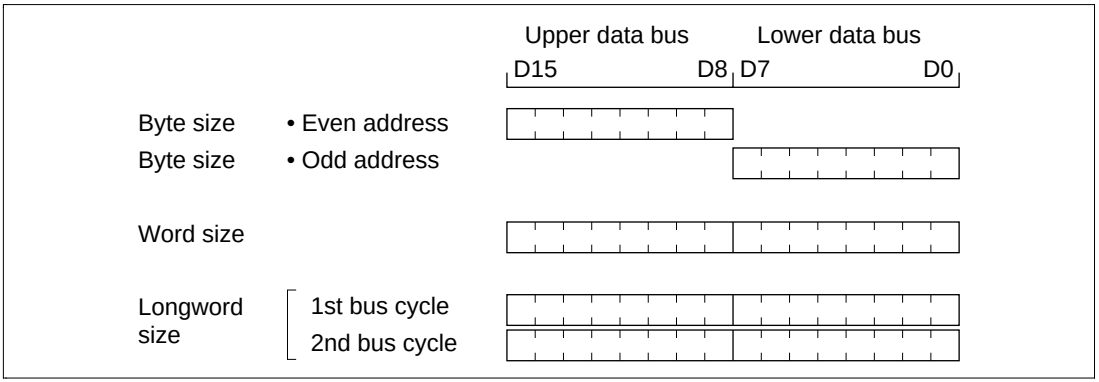


Figure 7-5 Access Sizes and Data Alignment Control (16-Bit Access Space)

7.4.3 Valid Strobes

Table 7-4 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves of the data bus.

In a write, the $\overline{HWR}$ signal is valid for the upper half of the data bus, and the $\overline{LWR}$ signal for the lower half.

Table 7-4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower data bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Note: Hi-Z: High impedance.
Invalid: Input state; input value is ignored.

7.4.4 Basic Timing

8-Bit 2-State Access Space: Figure 7-6 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

The $\overline{\text{LWR}}$ pin is fixed high. Wait states cannot be inserted.

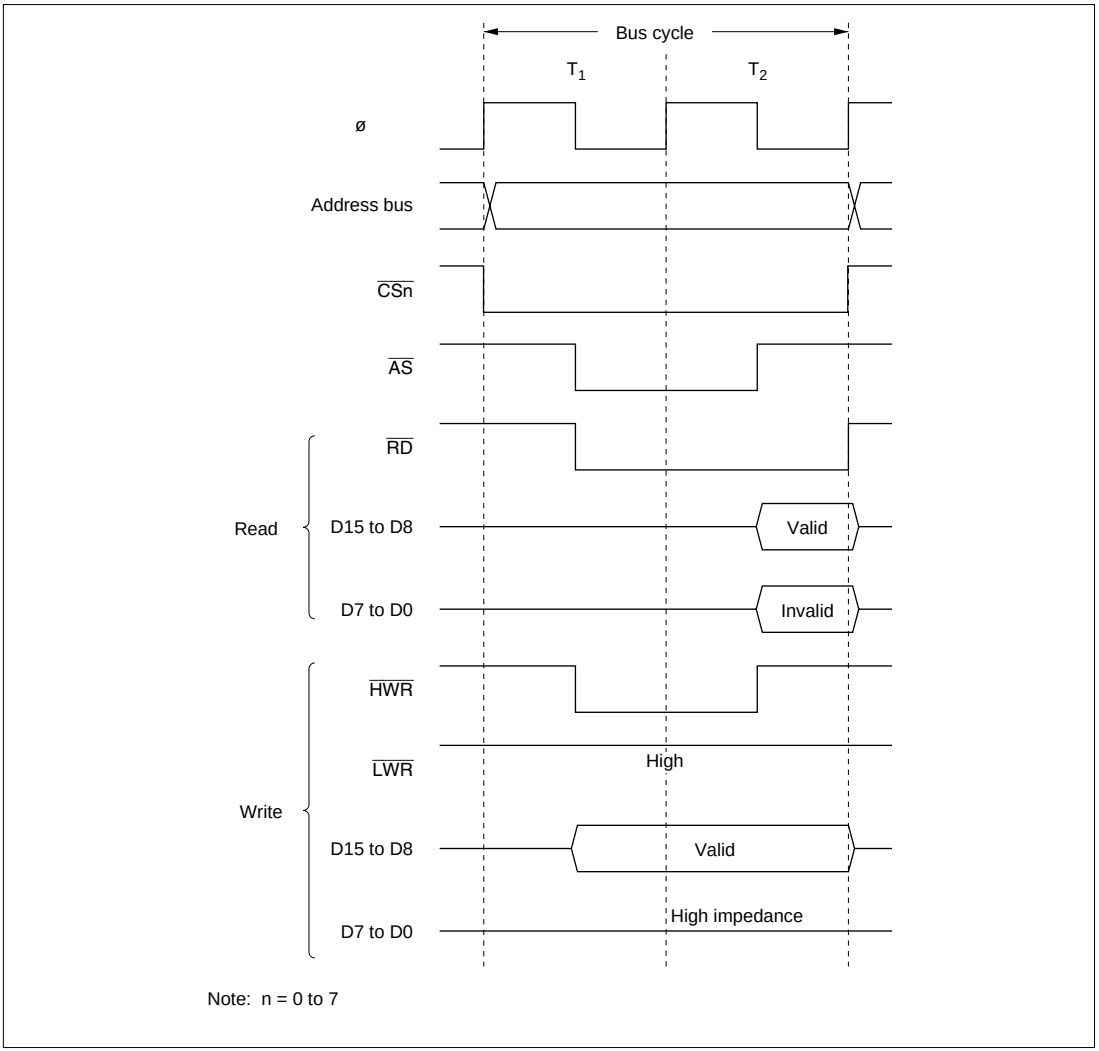


Figure 7-6 Bus Timing for 8-Bit 2-State Access Space

8-Bit 3-State Access Space: Figure 7-7 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

The $\overline{\text{LWR}}$ pin is fixed high. Wait states can be inserted.

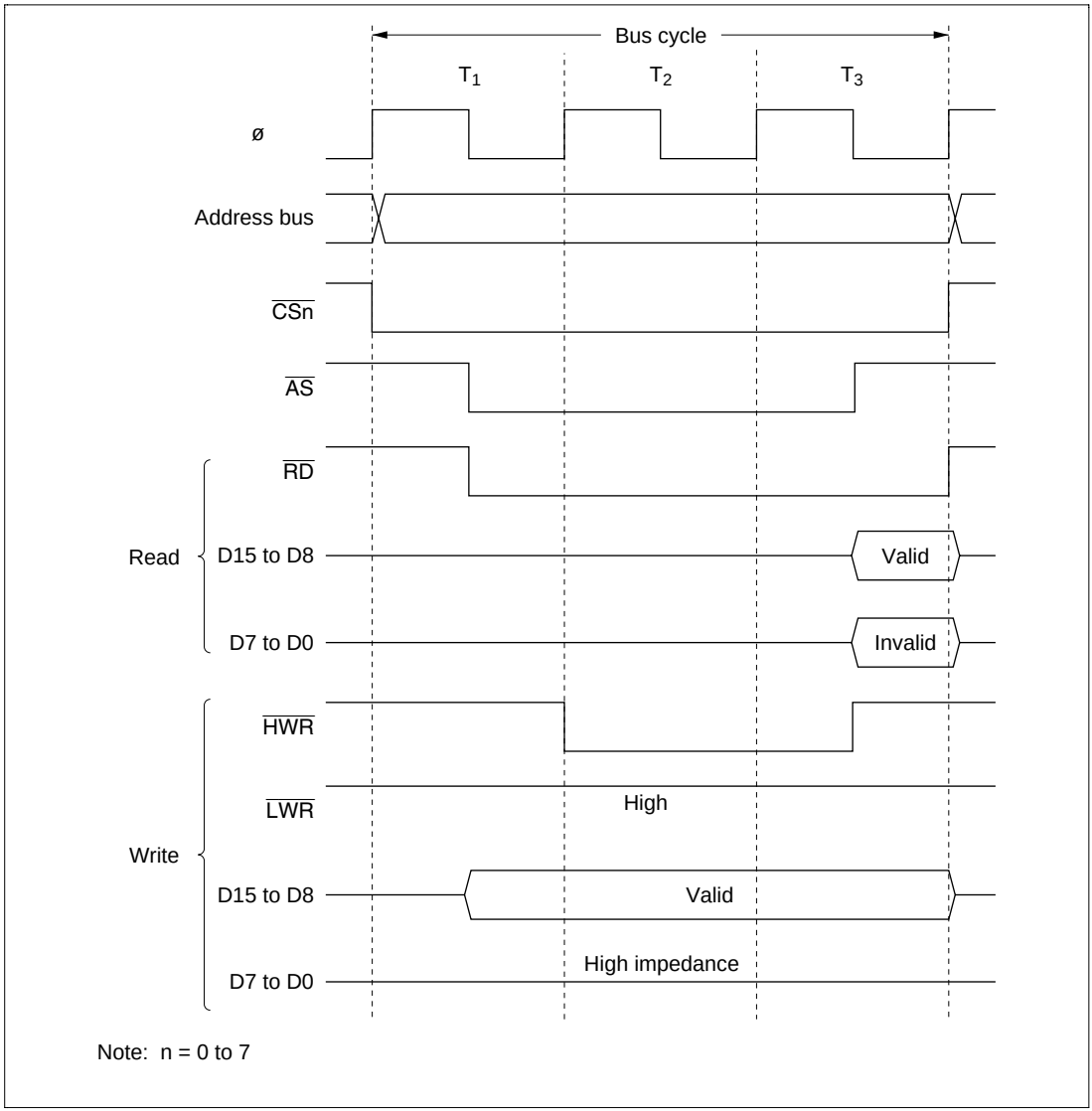


Figure 7-7 Bus Timing for 8-Bit 3-State Access Space

16-Bit 2-State Access Space: Figures 7-8 to 7-10 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states cannot be inserted.

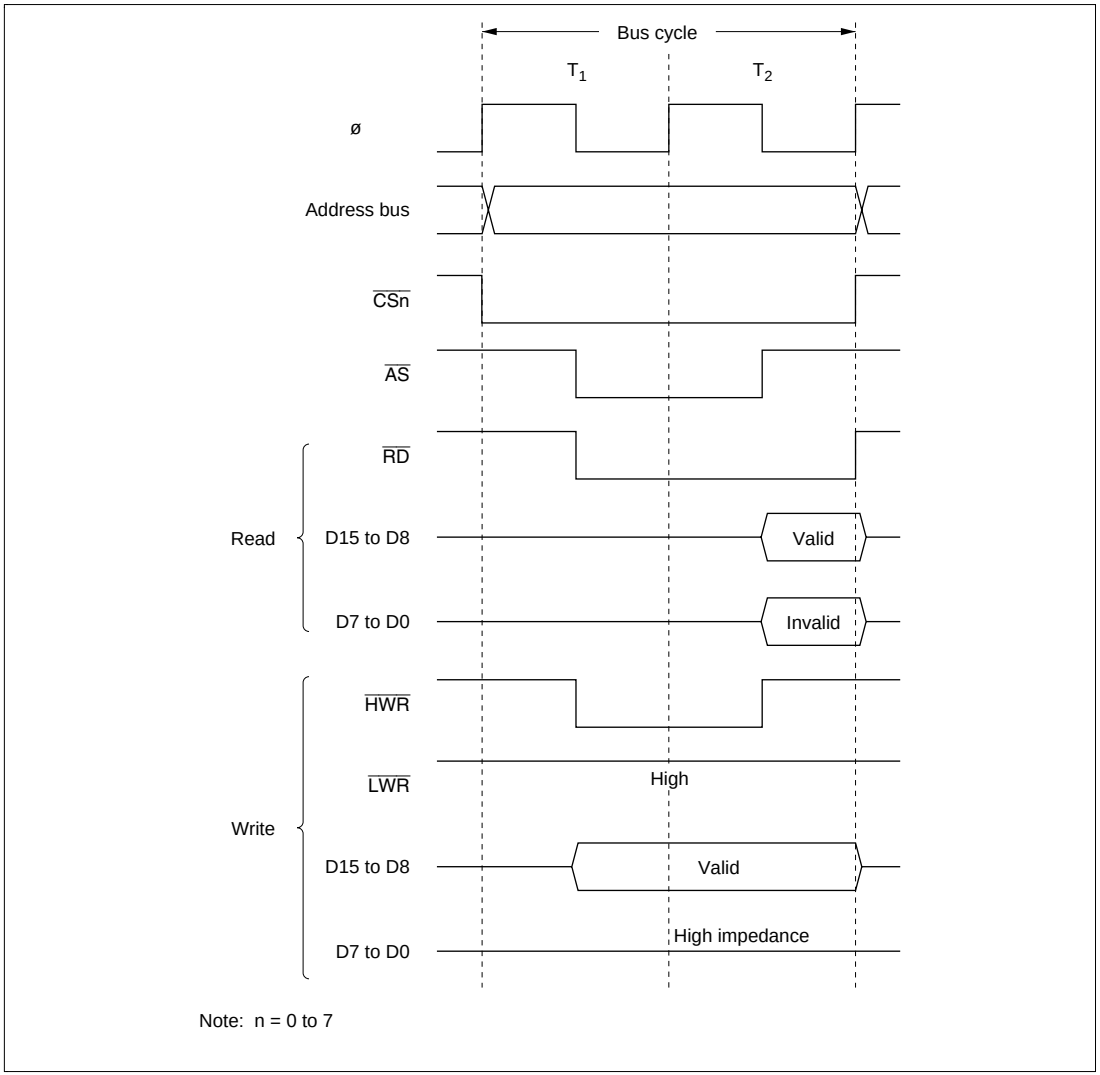


Figure 7-8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

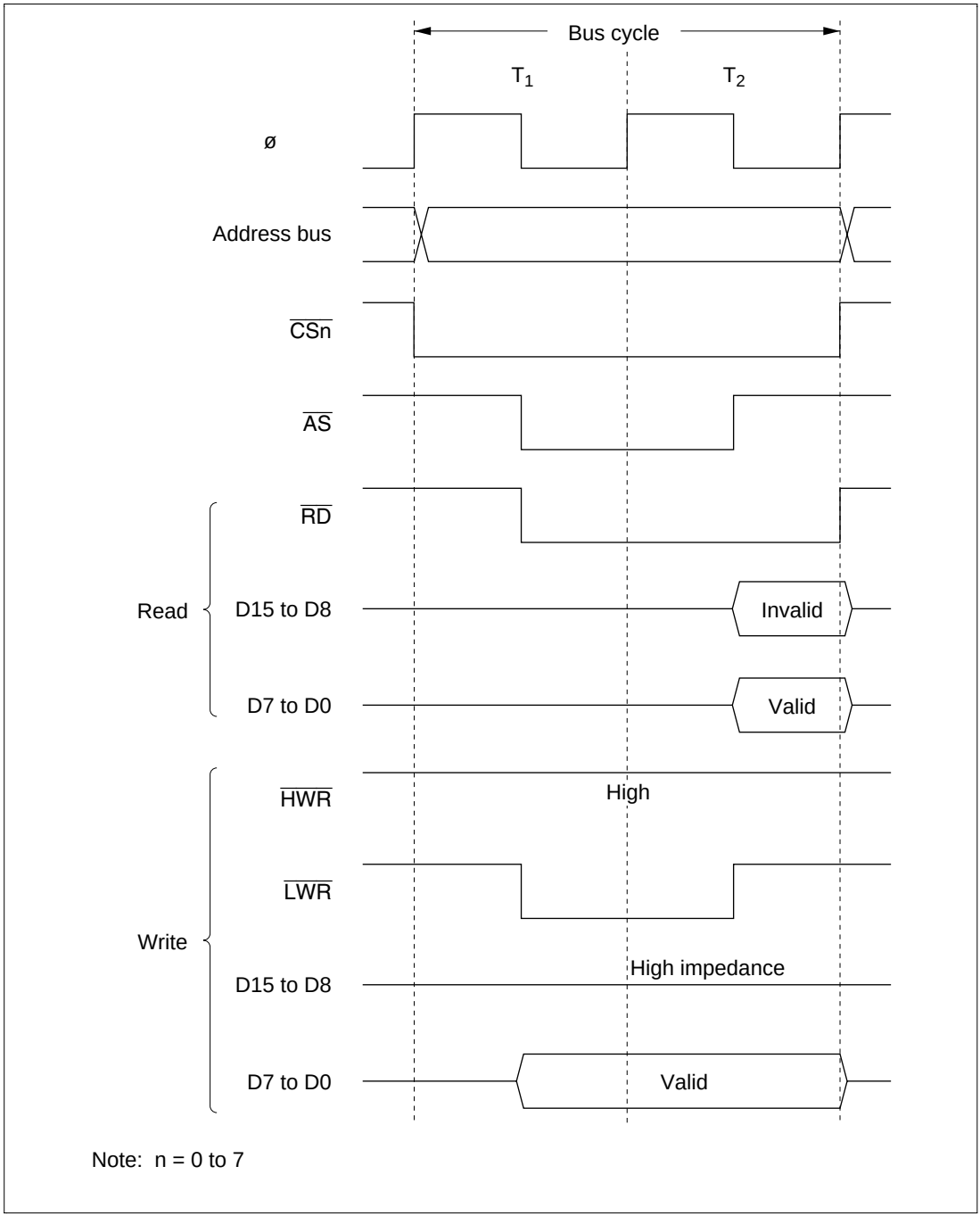


Figure 7-9 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)

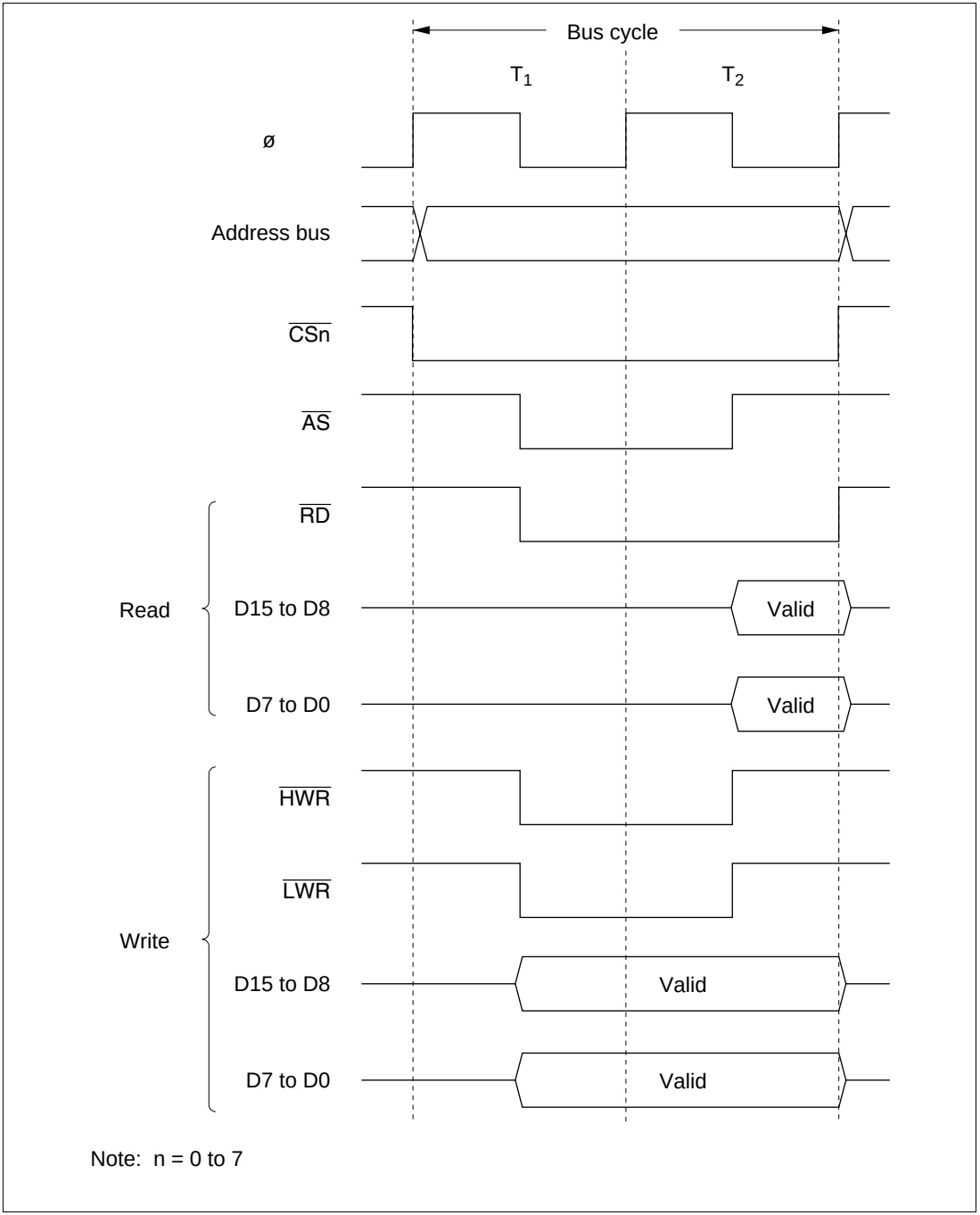


Figure 7-10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

16-Bit 3-State Access Space: Figures 7-11 to 7-13 show bus timings for a 16-bit 3-state access space. When a 16-bit access space is accessed , the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states can be inserted.

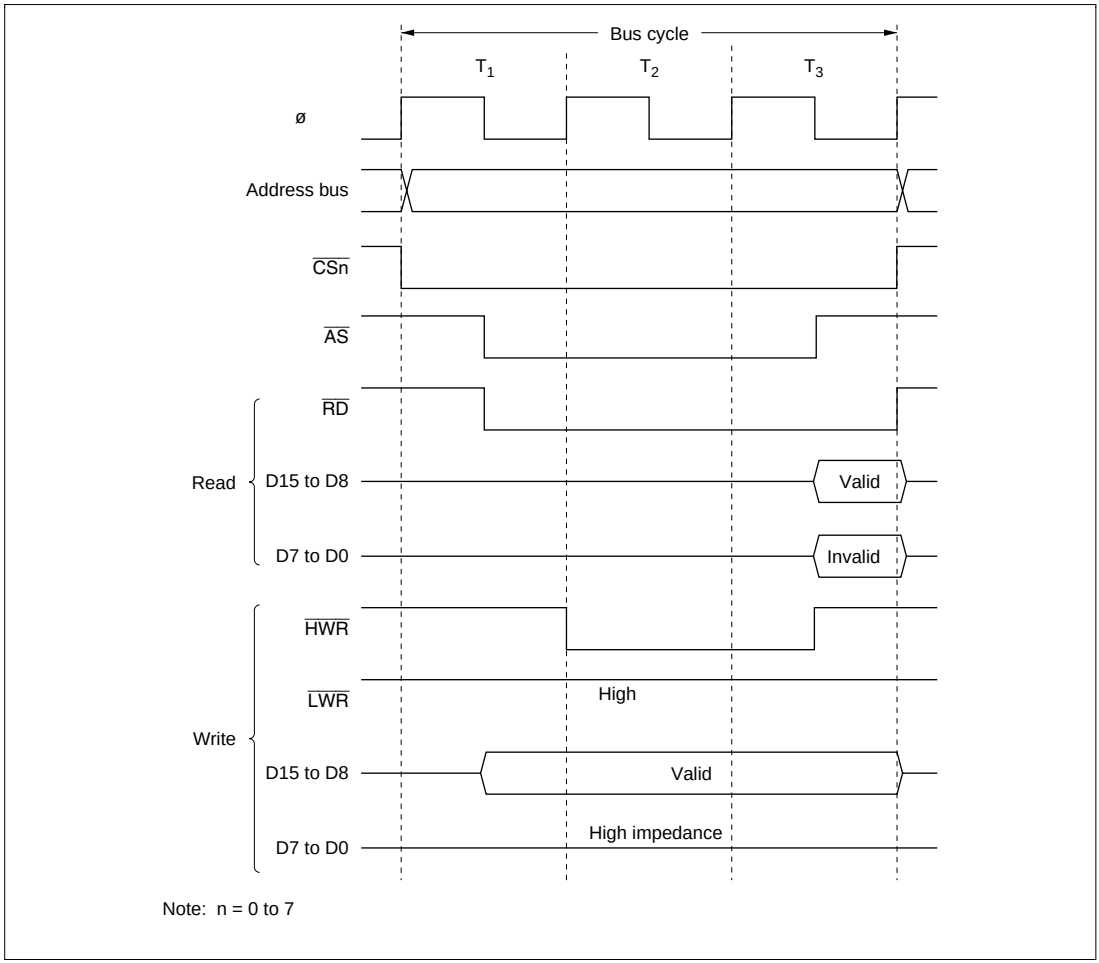


Figure 7-11 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

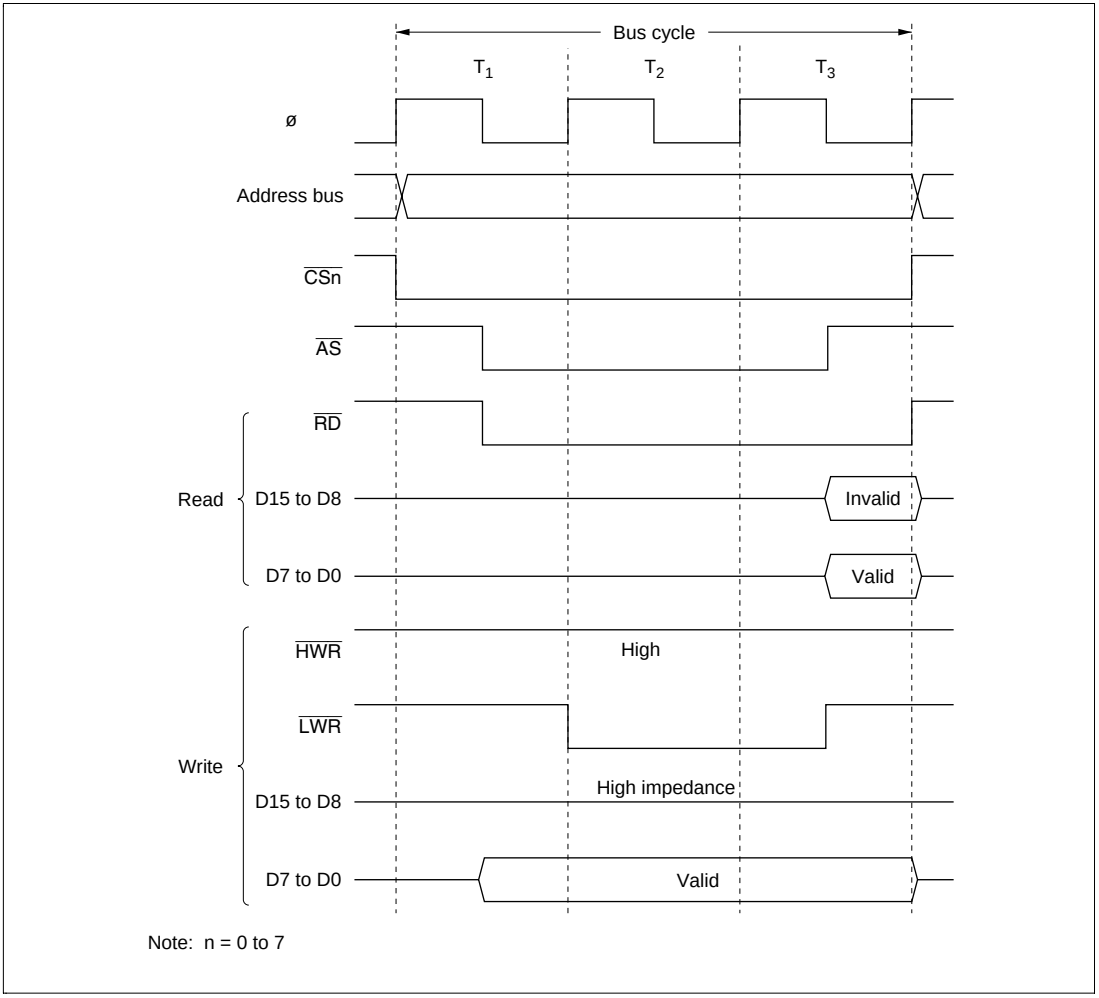


Figure 7-12 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

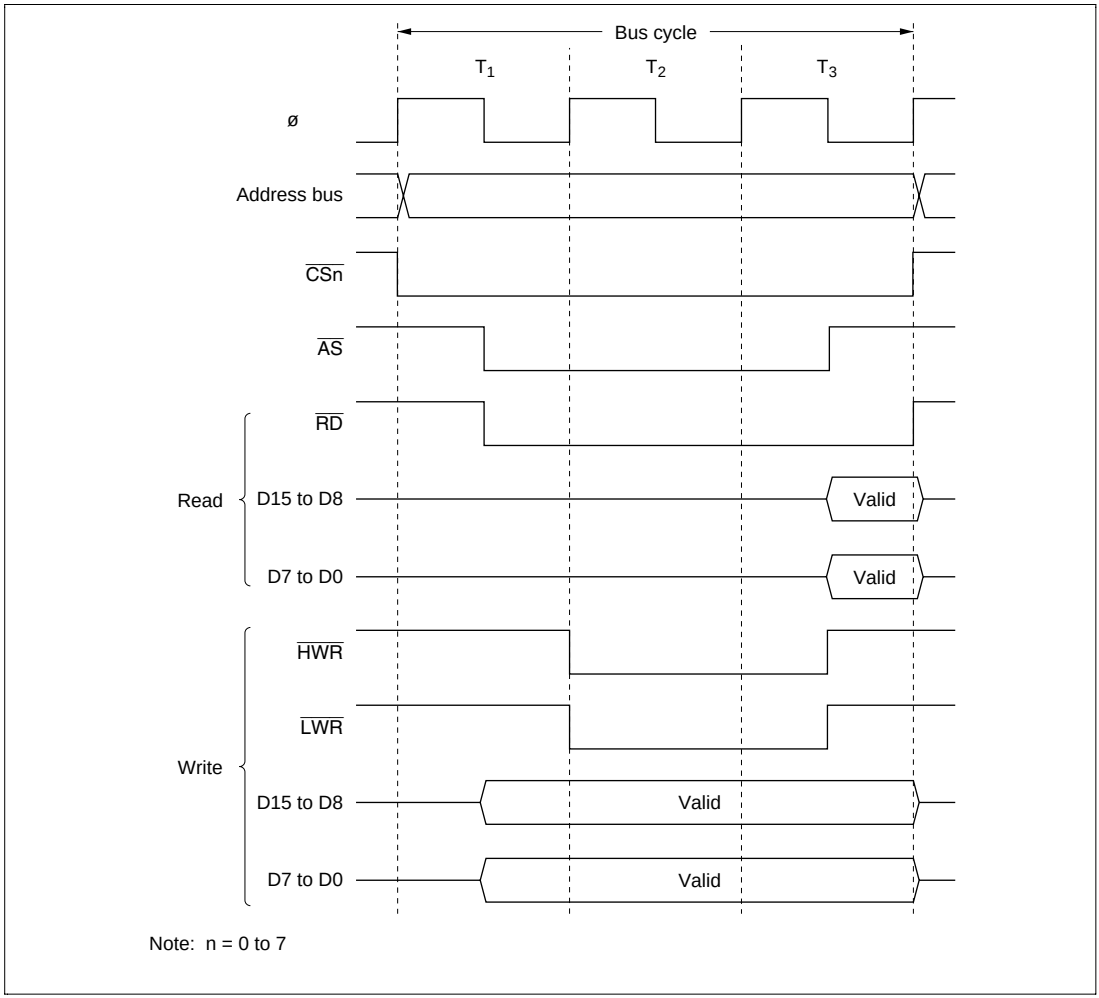


Figure 7-13 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

7.4.5 Wait Control

When accessing external space, the H8S/2633 Series can extend the bus cycle by inserting one or more wait states (T_w). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

Program Wait Insertion

From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in 3-state access space, according to the settings of WCRH and WCRL.

Pin Wait Insertion

Setting the WAITE bit in BCRL to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. Program wait insertion is first carried out according to the settings in WCRH and WCRL. Then , if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ø in the last T_2 or T_w state, a T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

This is useful when inserting four or more T_w states, or when changing the number of T_w states for different external devices.

The WAITE bit setting applies to all areas.

Figure 7-14 shows an example of wait state insertion timing.

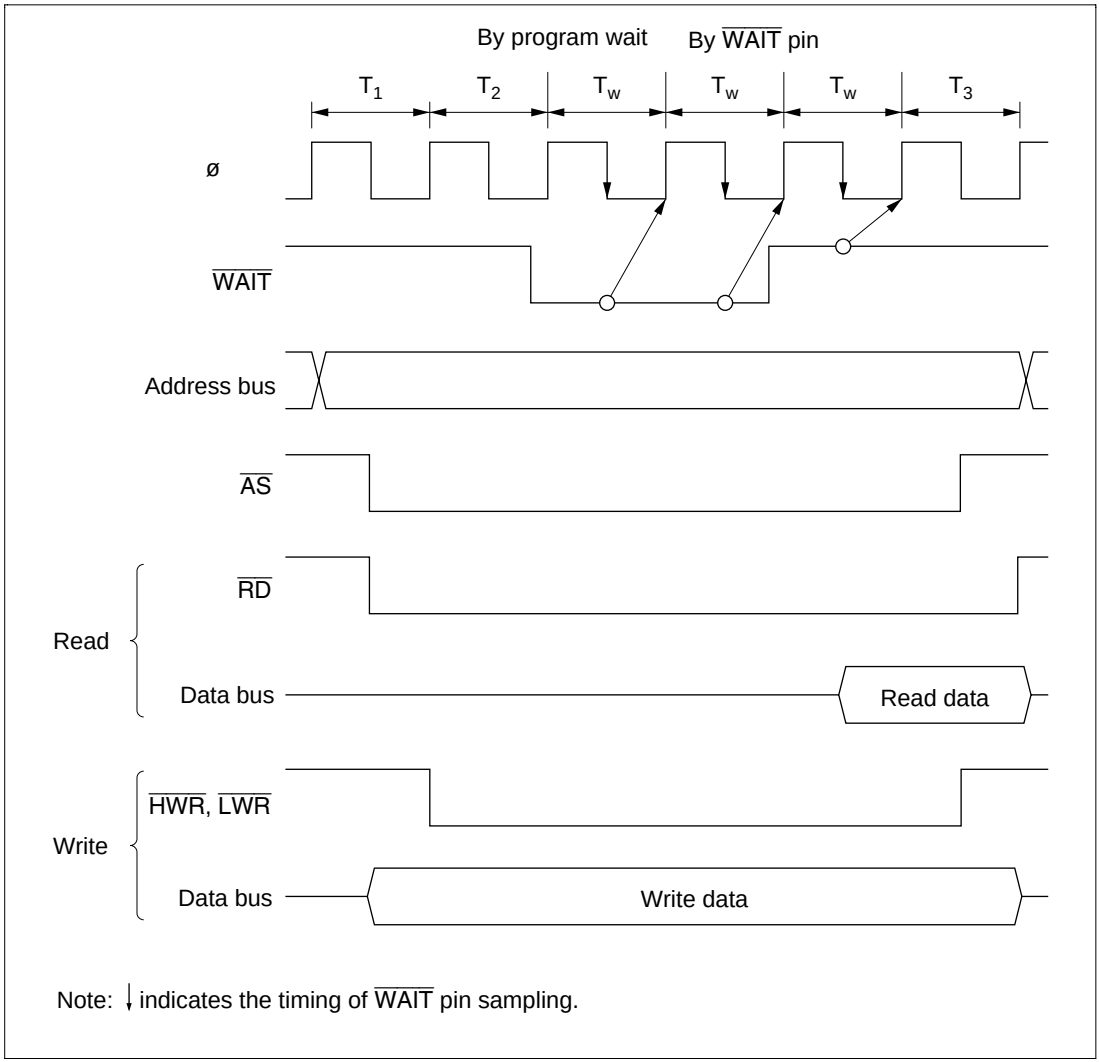


Figure 7-14 Example of Wait State Insertion Timing

The settings after a power-on reset are: 3-state access, 3 program wait state insertion, and WAIT input disabled. At a manual reset, the bus control register values are retained and wait control continues as before the reset.

7.5 DRAM Interface

7.5.1 Overview

This LSI allows area 2 to 5 external space to be set as DRAM space and DRAM interfacing to be performed. With the DRAM interface, DRAM can be directly connected to the LSI. BCRH RMTS2 to RMTS0 allow the setting up of 2, 4, or 8MB DRAM space. Burst operation is possible using high-speed page mode.

7.5.2 Setting up DRAM Space

To set up areas 2 to 5 as DRAM space, set the RMTS2 to RMTS0 bits of BCRH. Table 7-5 shows the relationship between the settings of the RMTS2 to RMTS0 bits and DRAM space. You can select (1) one area (area 2), (2) two areas (areas 2 and 3), or (3) four areas (areas 2 to 5).

Using 16 64M DRAMs requires a 4M word (8MB) contiguous space. Setting RMTS2 to RMTS0 to 1 allows areas 2 to 5 to be configured as one contiguous DRAM space. The RAS signal can be output from the $\overline{CS2}$ pin, and $\overline{CS3}$ to $\overline{CS5}$ can be used as input ports. In this configuration, the bus widths are the same for areas 2 to 5.

Table 7-5 RMTS2 to RMTS0 Settings vs DRAM Space

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space
1	1	1	Contiguous DRAM space	Contiguous DRAM space	Contiguous DRAM space	Contiguous DRAM space

7.5.3 Address Multiplexing

In the case of DRAM space, the row address and column address are multiplexed. With address multiplexing, the MXC1 and MXC0 bits of the MCR select the amount of shift in the row address. Table 7-6 shows the relationship between MXC1 and MXC0 settings and the shift amount.

Table 7-6 MXC1 and MXC0 Settings vs Address Multiplexing

	MCR		Shift		Address Pin												
	MXC1	MXC0	Amount	A23 to A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address	0	0	8 bits	A23 to A13	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
		1	9 bits	A23 to A13	A12	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
	1	0	10 bits	A23 to A13	A12	A11	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
		1	Do not set	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Column address	—	—	—	A23 to A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

7.5.4 Data Bus

Setting the ABWCR bit of an area set as DRAM space to 1 sets the corresponding area as 8-bit DRAM space. Clearing the ABWCR bit to 0 sets the area as 16-bit DRAM. 16-bit DRAMs can be directly connected in the case of 16-bit DRAM space.

With 8-bit DRAM space, the high data bus byte (D15 to D8) is valid. With 16-bit DRAM space, the high and low data bus bytes (D15 to D0) are valid.

The access size and data alignment are the same as for the standard bus interface. See Section 7.4.2, Data Size and Data Alignment for details.

7.5.5 DRAM Interface Pins

Table 7-7 shows the pins used for the DRAM interface, and their functions.

Table 7-7 DRAM Interface Pin Configuration

Pin	In DRAM Mode	Name	Direction	Function
$\overline{\text{HWR}}$	$\overline{\text{WE}}$	Write enable	Output	Write enable when accessing DRAM space in 2 CAS mode.
$\overline{\text{LCAS}}$	$\overline{\text{LCAS}}$	Lower column address strobe	Output	Lower column address strobe signal when accessing 16-bit DRAM space.
$\overline{\text{CS2}}$	$\overline{\text{RAS2}}$	Row address strobe 2	Output	Row address strobe when area 2 set as DRAM space.
$\overline{\text{CS3}}$	$\overline{\text{RAS3}}$	Row address strobe 3	Output	Row address strobe when area 3 set as DRAM space.
$\overline{\text{CS4}}$	$\overline{\text{RAS4}}$	Row address strobe 4	Output	Row address strobe when area 4 set as DRAM space.
$\overline{\text{CS5}}$	$\overline{\text{RAS5}}$	Row address strobe 5	Output	Row address strobe when area 5 set as DRAM space.
$\overline{\text{CAS}}$	$\overline{\text{UCAS}}$	Upper column address strobe	Output	Upper column address strobe when accessing DRAM space.
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal
A12 to A0	A12 to A0	Address pin	Output	Multiplexed output of row address and column address.
D15 to D0	D15 to D0	Data pin	Input/output	Data input/output pin.
$\overline{\text{OE}}$	$\overline{\text{OE}}$*	Output enable pin	Output	Output enable signal when accessing DRAM space in read mode.

Note: * Valid when OES bit set to 1.

7.5.6 Basic Timing

Figure 7-15 shows the basic access timing for DRAM space. There are four basic DRAM timing states. In contrast to the standard bus interface, the corresponding ASTCR bit only controls the enabling/disabling of wait insertion and has no effect on the number of access states. When the corresponding ASTCR bit is cleared to 0, no wait states can be inserted in the DRAM access cycle.

The four basic timing states are as follows: T_P (precharge cycle) 1 state, T_r (row address output cycle) 1 state, T_{c1} and T_{c2} (column address output cycle) two states.

When RCTS is set to 1, the $\overline{CAS}$ signal timing differs when reading and writing, being asserted Ω cycle earlier when reading.

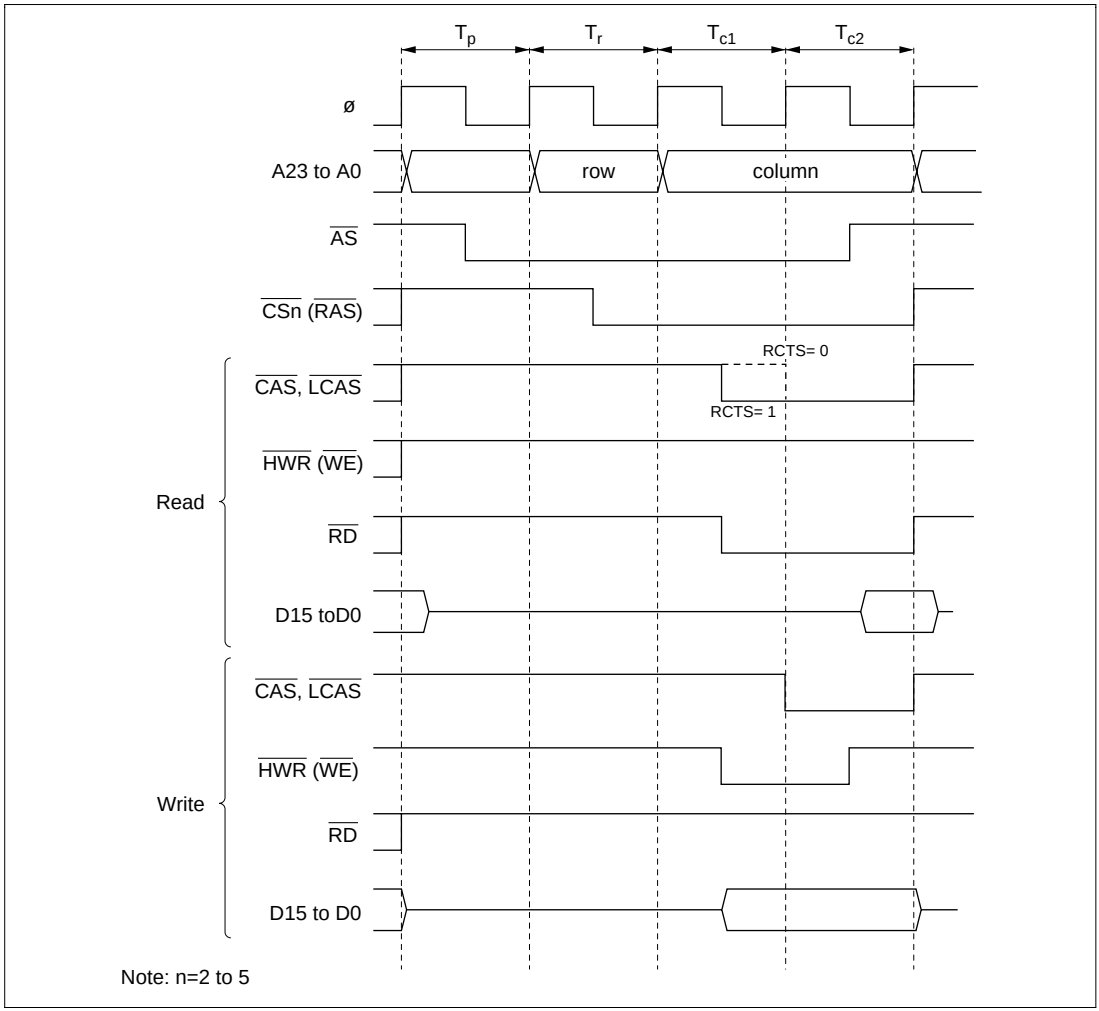


Figure 7-15 Basic Access Timing

7.5.7 Precharge State Control

When accessing DRAM, it is essential to secure a time for RAS precharging. In this LSI, it is therefore necessary to insert 1 T_p state when accessing DRAM space. By setting the TPC bit of the MCR to 1, T_p can be changed from 1 state to 2 states. Set the appropriate number of T_p cycles according to the type of DRAM connected and the operation frequency of the LSI. Figure 7-16 shows the timing when T_p is set for 2 states.

Setting the TPC bit to 1 also sets the refresh cycle T_p to 2 states.

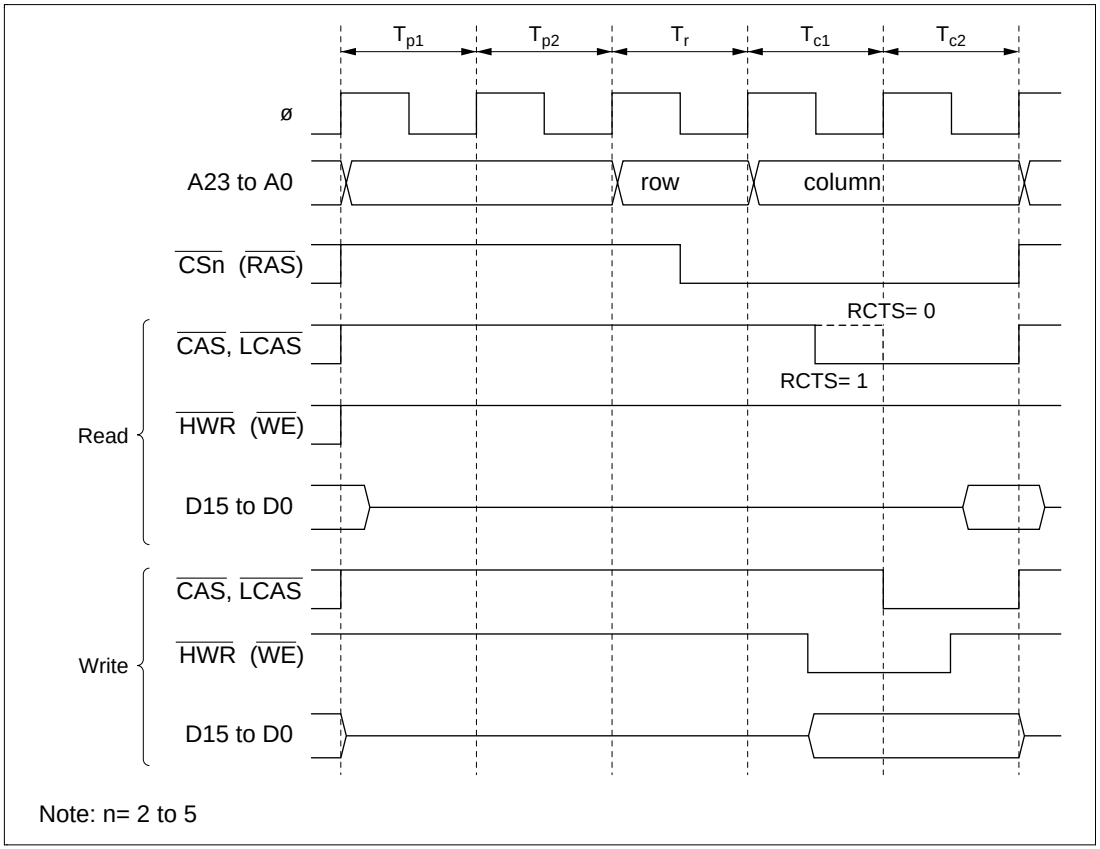


Figure 7-16 Timing With Two Precharge Cycles

7.5.8 Wait Control

There are two methods of inserting wait states in DRAM access: (1) insertion of program wait states, and (2) insertion of pin waits via $\overline{\text{WAIT}}$ pin.

(1) Insertion of Program Wait States

Setting the ASTCR bit of an area set for DRAM to 1 automatically inserts from 0 to 3 wait states, as set by WCRH and WCRL, between the T_{c1} state and T_{c2} state.

When a program wait is inserted, the write wait function is activated and only the $\overline{\text{CAS}}$ signal is output only during the T_{c2} state when writing.

Figure 7-17 shows example timing for the insertion of program waits.

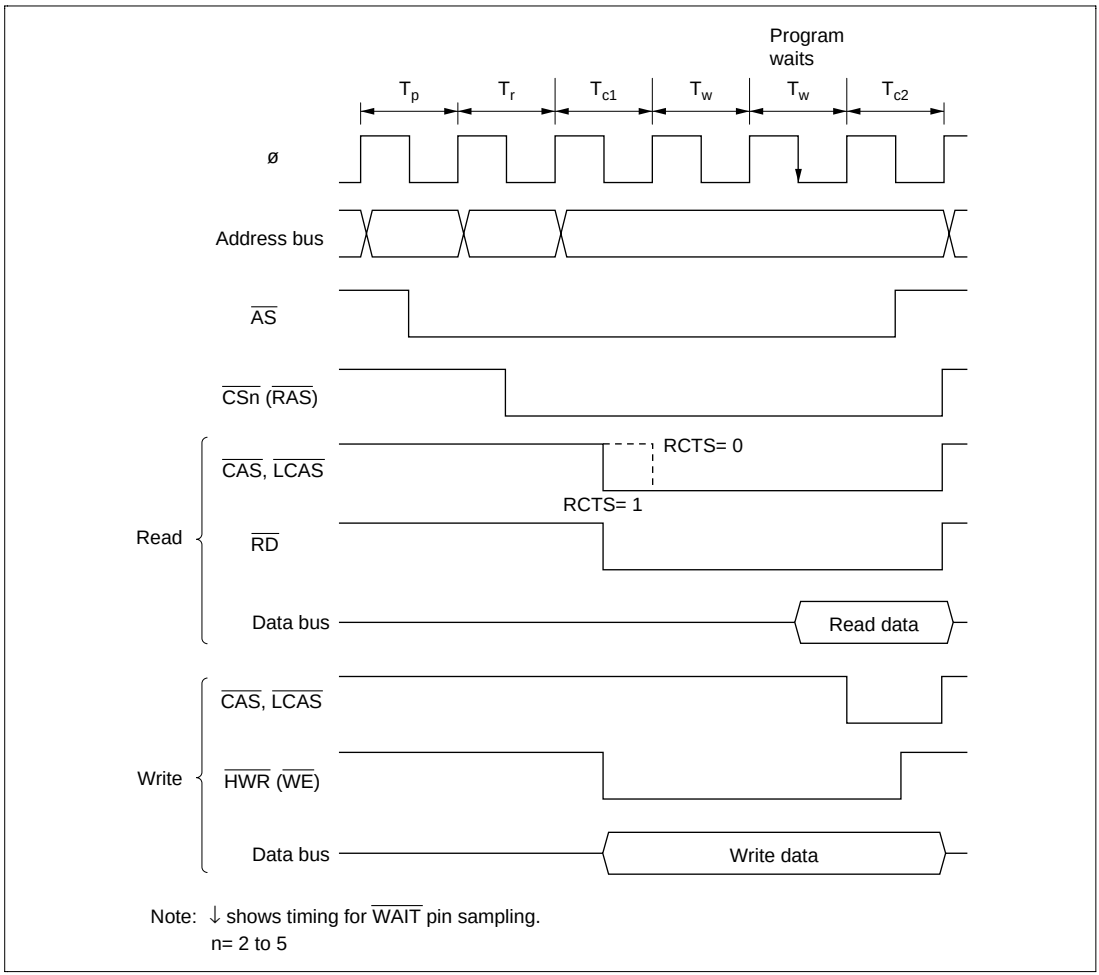


Figure 7-17 Example Program Wait Insertion Timing (Wait 2 State Insertion)

(2) Insertion of Pin Waits

When the WAITE bit of BCRH is set to 1, wait input via the $\overline{\text{WAIT}}$ pin is valid regardless of the ASTCR AST bit. In this state, a program wait is inserted when the DRAM space is accessed. If the $\overline{\text{WAIT}}$ pin level is Low at the fall in ø in the final T_{c1} or T_w state, a further T_w is inserted. If the level of the $\overline{\text{WAIT}}$ pin is kept Low, T_w is inserted until the level of the $\overline{\text{WAIT}}$ pin changes to High.

When wait states are inserted via the $\overline{\text{WAIT}}$ pin, the $\overline{\text{CAS}}$ when writing is output after the T_w state.

Figure 7-18 shows example timing for the insertion of wait states via the $\overline{\text{WAIT}}$ pin.

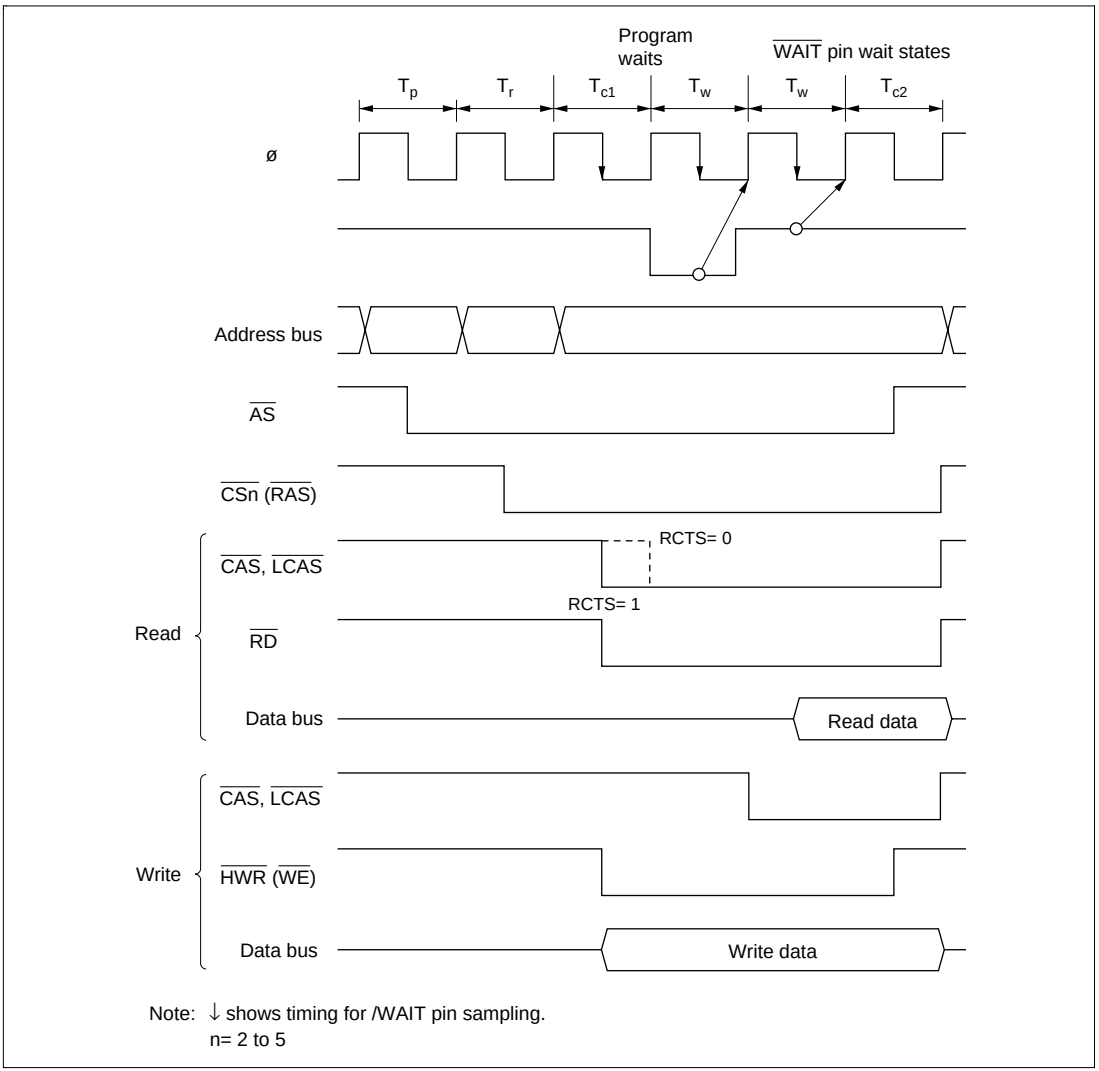


Figure 7-18 Example Timing for Insertion of Wait States via $\overline{\text{WAIT}}$ Pin

7.5.9 Byte Access Control

When 16-bit DRAMs are connected, the 2 CAS method can be used as the control signal required for byte access.

Figure 7-19 shows the 2 CAS method control timing. Figure 7-20 shows an example of connecting DRAM in high-speed page mode.

When all areas selected as DRAM space are set as 8-bit space, the $\overline{\text{LCAS}}$ pin functions as an I/O port.

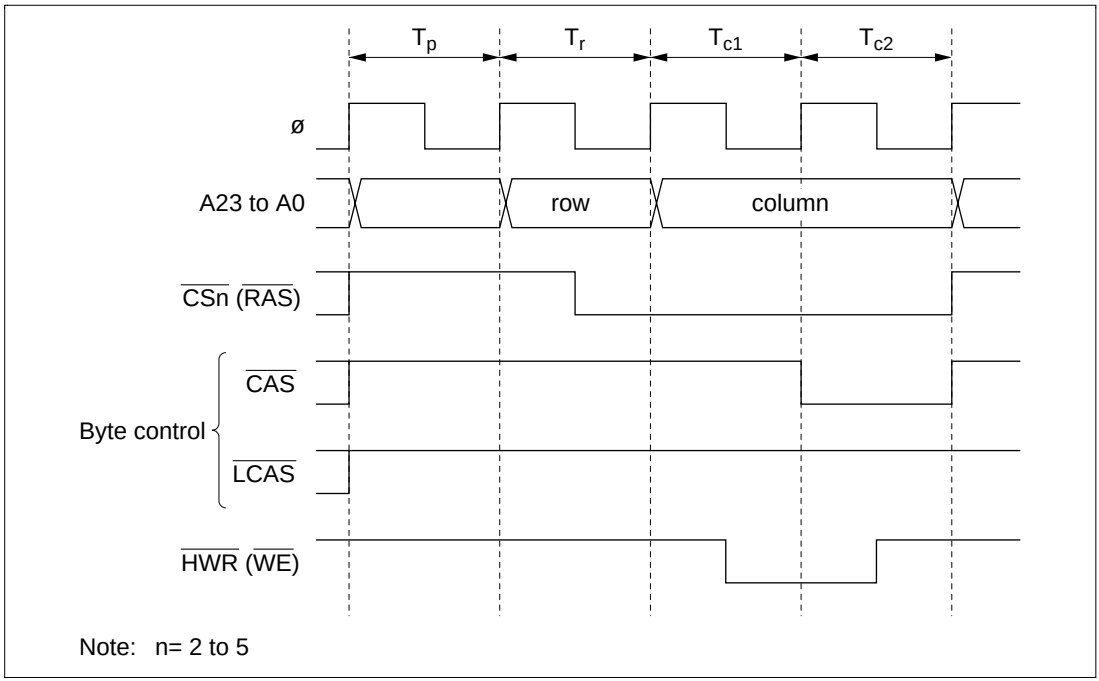


Figure 7-19 2 CAS Method Control Timing (For High Byte Write Access)

When using DRAM EDO page mode, either use $\overline{\text{OE}}$ to control the read data or, as shown in Figure 7-20, select RAS up mode. Figure 7-21 is an example of DRAM connection in EDO page mode when OES=1.

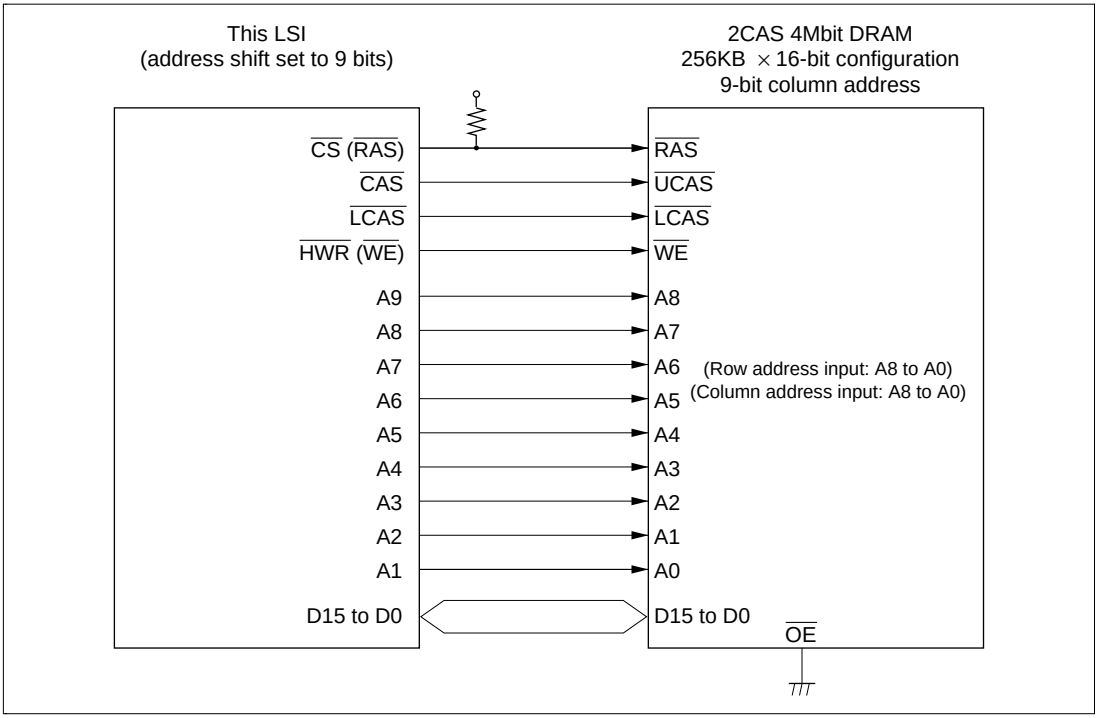


Figure 7-20 High-speed Page Mode DRAM

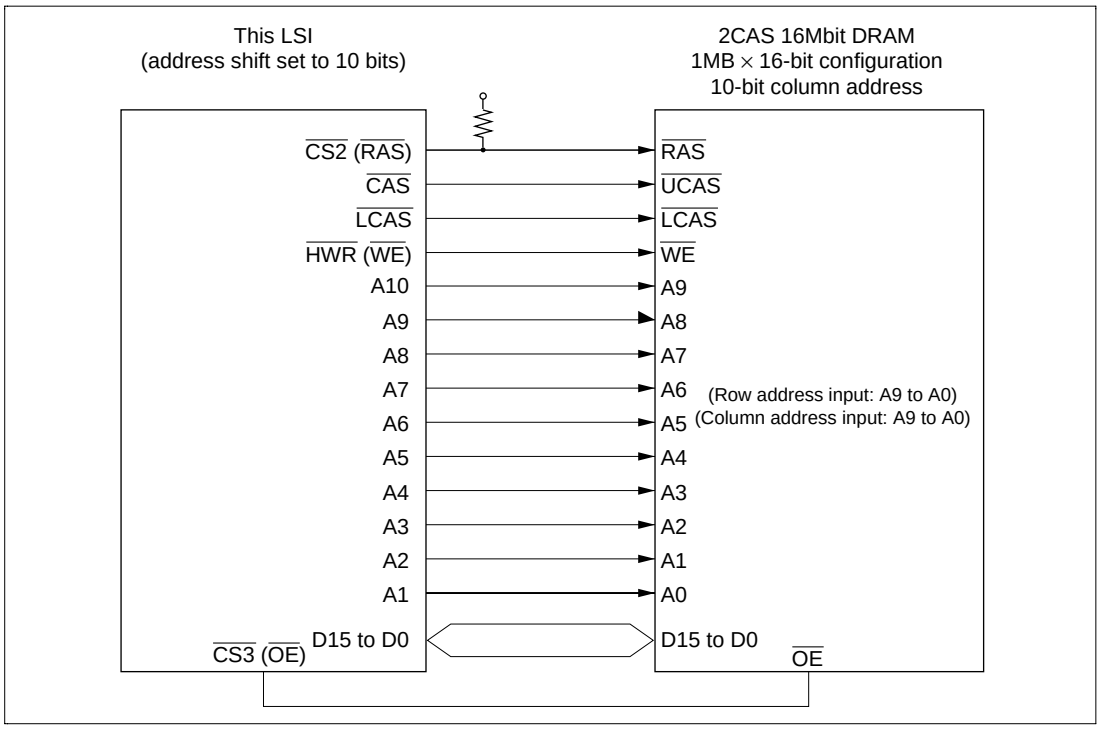


Figure 7-21 Example Connection of EDO Page Mode DRAM (OES=1)

7.5.10 Burst Operation

In addition to full DRAM access (normal DRAM access), in which the row address is output each time the data in DRAM is accessed, there is also a high-speed page mode that allows high-speed access (burst access). In this method, if the same row address is accessed successively, the row address is output once and then only the column address is changed. Burst access is selected by setting the BE bit of the MCR to 1.

(1) Operation Timing for Burst Access (High-Speed Page Mode)

Figure 7-22 shows the operation timing for burst access. When the DRAM space is successively accessed, the $\overline{CAS}$ signal and column address output cycle (2 state) are continued as long as the row address is the same in the preceding and succeeding access cycles. The MXC1 and MXC0 bits of the MCR specify which row address is compared.

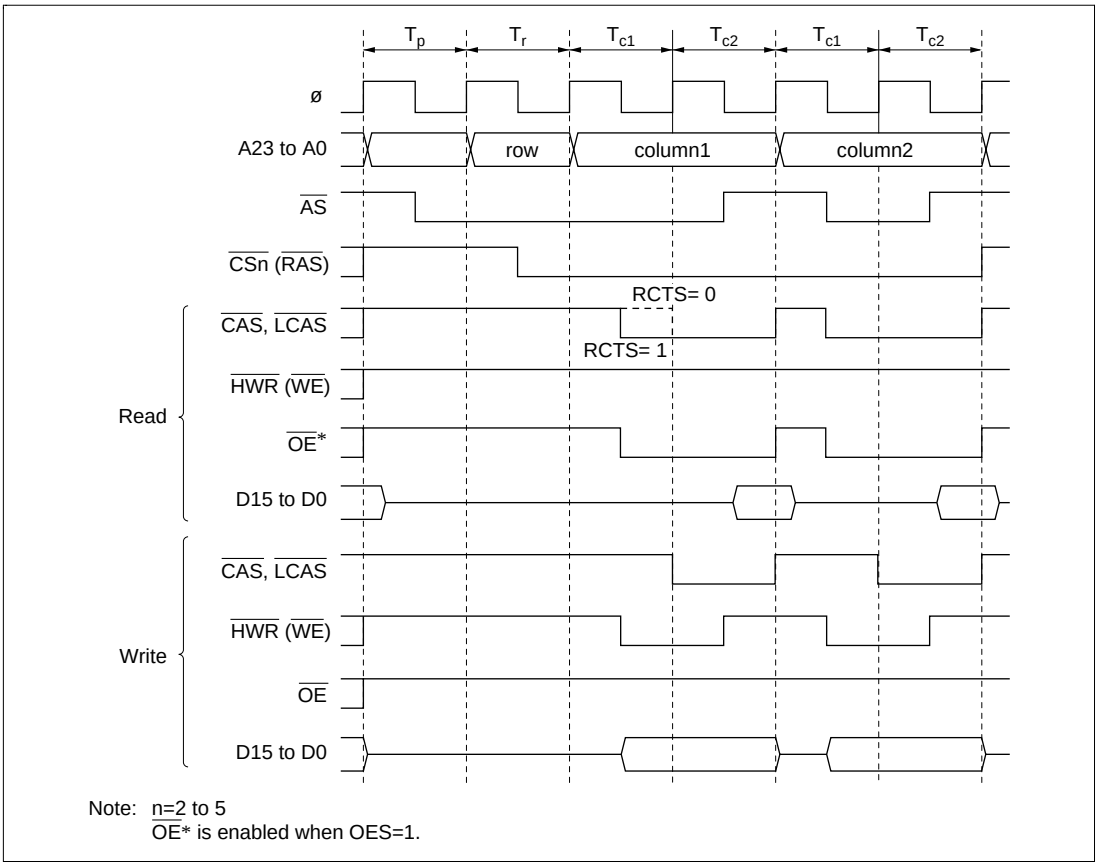


Figure 7-22 Operating Timing in High-Speed Page Mode

The bus cycle can also be extended in burst access by inserting wait states. The method and timing of inserting the wait states is the same as in full access. For details, see Section 7.5.8, “Wait Control.”

(2) RAS Down Mode and RAS Up Mode

Even when burst operation is selected, DRAM access may not be continuous, but may be interrupted by accessing another area. In this case, burst operation can be continued by keeping the $\overline{RAS}$ signal level Low while the other area is accessed and then accessing the same row address in the DRAM space.

- RAS down mode

  To select RAS down mode, set the RCDM bit of the MCR to 1. When DRAM access is interrupted and another area accessed, the $\overline{RAS}$ signal level is kept Low and, if the row address is the same as previously when the DRAM space is again accessed, burst access is continued. Figure 7-23 shows example RAS down mode timing.

  Note that if the refresh operation occurs when RAS is down, the $\overline{RAS}$ signal level changes to High.

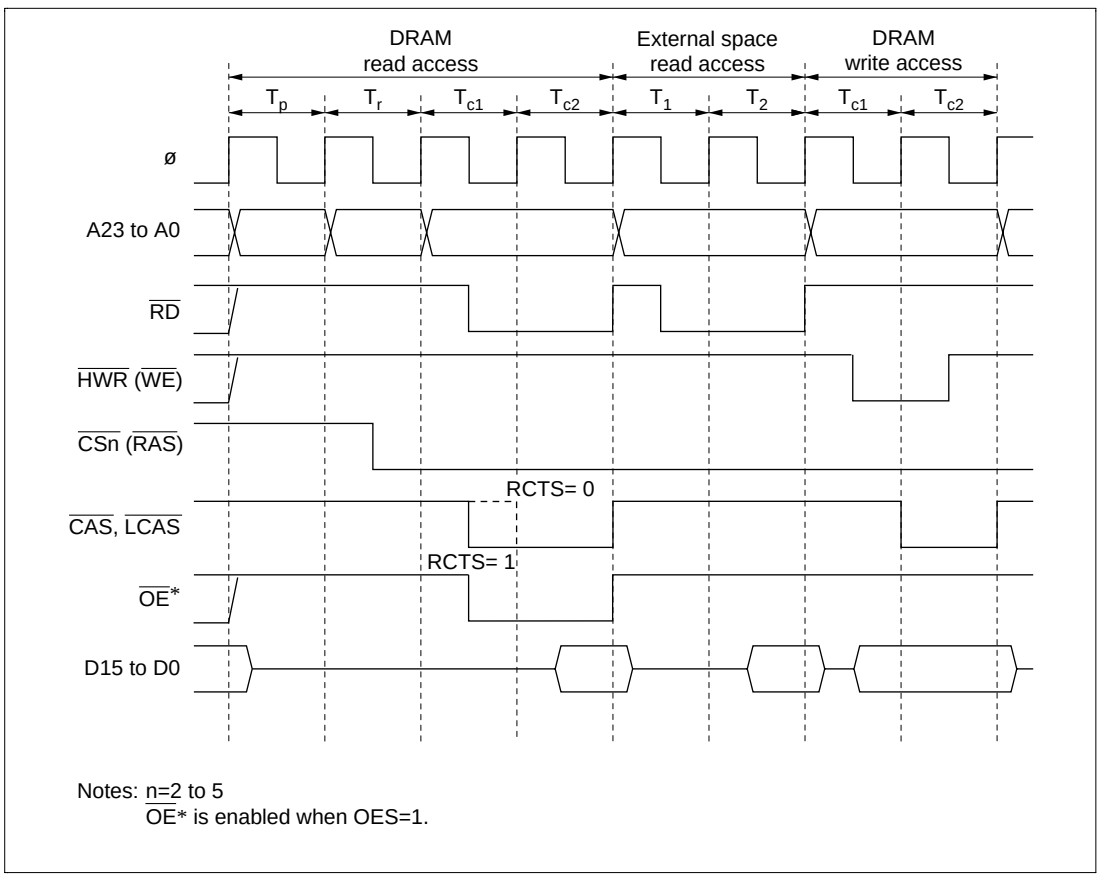


Figure 7-23 Example Operation Timing in RAS Down Mode

- RAS up mode

  To select RAS up mode, clear the RCDM bit of the MCR to 0. If DRAM access is interrupted to access another area, the $\overline{RAS}$ signal level returns to High. Burst operation is only possible when the DRAM space is contiguous. Figure 7-24 shows example timing in RAS up mode. Note that the $\overline{RAS}$ signal level does not return to High in burst ROM space access.

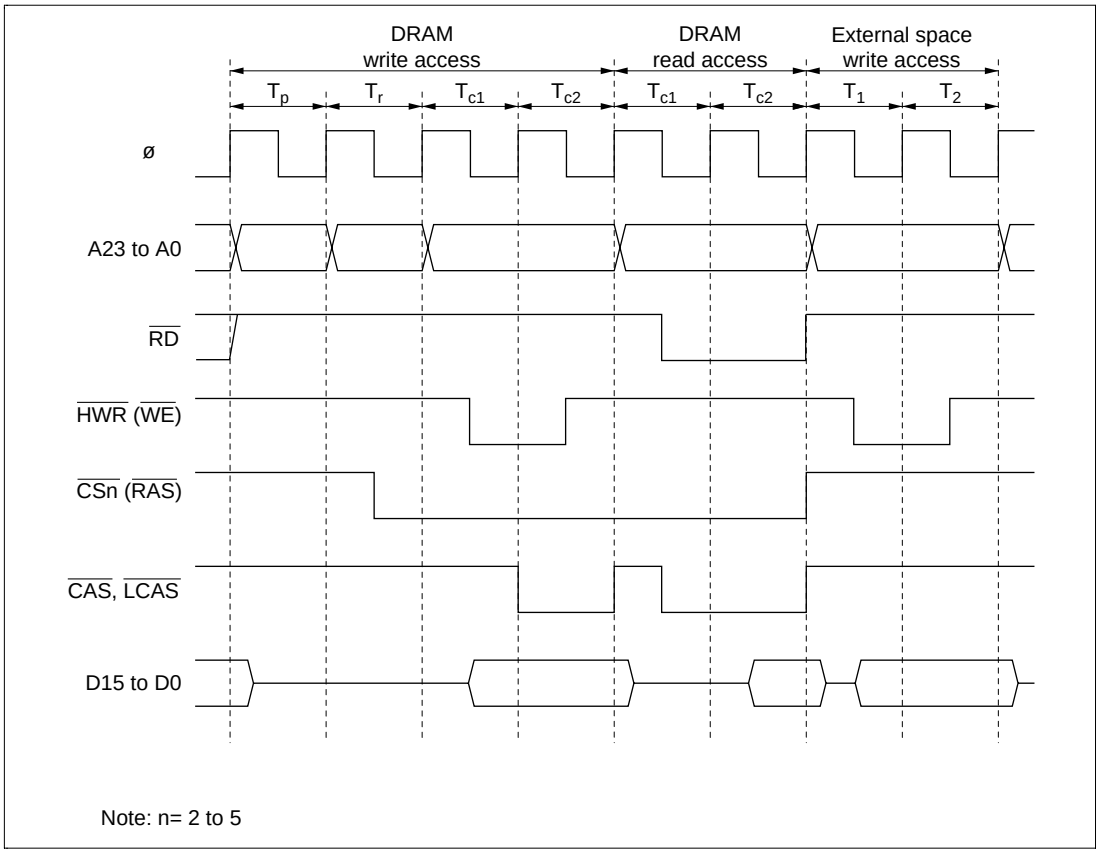


Figure 7-24 Example Operation Timing in RAS Up Mode

7.5.11 Refresh Control

This LSI has a DRAM refresh control function. There are two refresh methods: (1) CAS-before-RAS (CBR) and (2), self refresh.

(1) CAS-Before-RAS (CBR) Refresh

To select CBR refresh, set the RFSHE bit of DRAMCR to 1 and clear the RMODE bit to 0.

In CBR refresh, the input clock selected with the CKS2 to CKS0 bits of DRAMCR are used for the RTCNT count-up. Refresh control is performed when the count reaches the value set in RTCOR (compare match). The RTCNT is then reset and the count again started from H'00. That is, the refresh is repeated at the set interval determined by RTCOR and CKS2 to CKS0. Set RTCOR and CKS2 to CKS0 to satisfy the refresh cycle for the DRAM being used.

The RTCNT count up starts when the CKS2 to CKS0 bits are set. The RTCNT and RTCOR values should therefore be set before setting CKS2 to CKS0. When a value is set in RTCOR, RTCNT is cleared. When RTCNT is set at the same time that it is reset by a compare match, the value written to RTCNT takes precedence.

When performing refresh control (RFSHE=1), do not clear the CMF flag.

Figure 7-25 shows RTCNT operation. Figure 7-26 shows compare match timing. And Figure 7-27 show CBR refresh timing.

Some types of DRAM do not allow the $\overline{WE}$ signal to be changed during the refresh cycle. In this case, set CBRM to 1. Figure 7-28 shows the timing. The $\overline{CS}$ signal is not controlled and a Low level is output when an access request occurs.

Note that other normal spaces are accessed during the CBR refresh cycle.

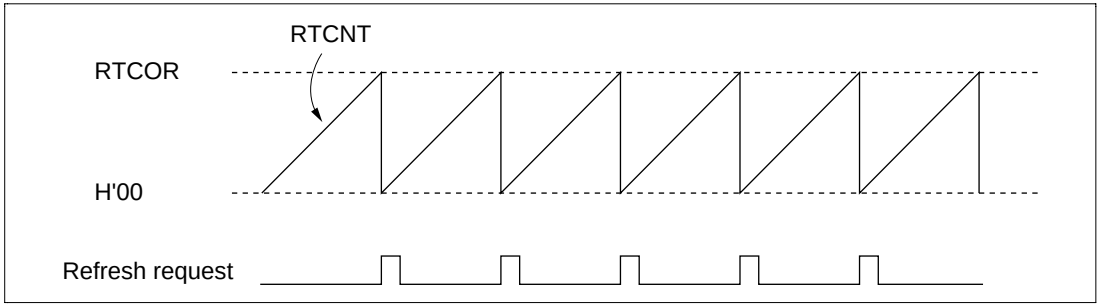


Figure 7-25 RTCNT Operation

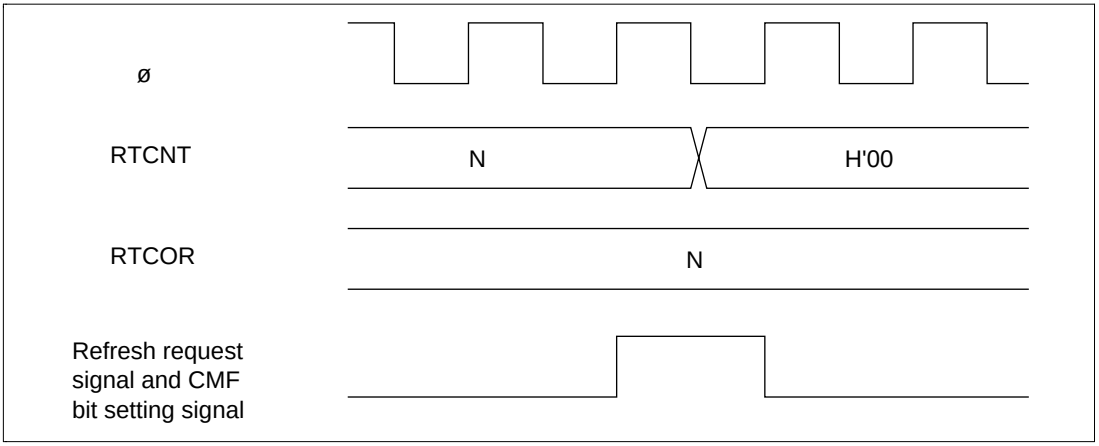


Figure 7-26 Compare Match Timing

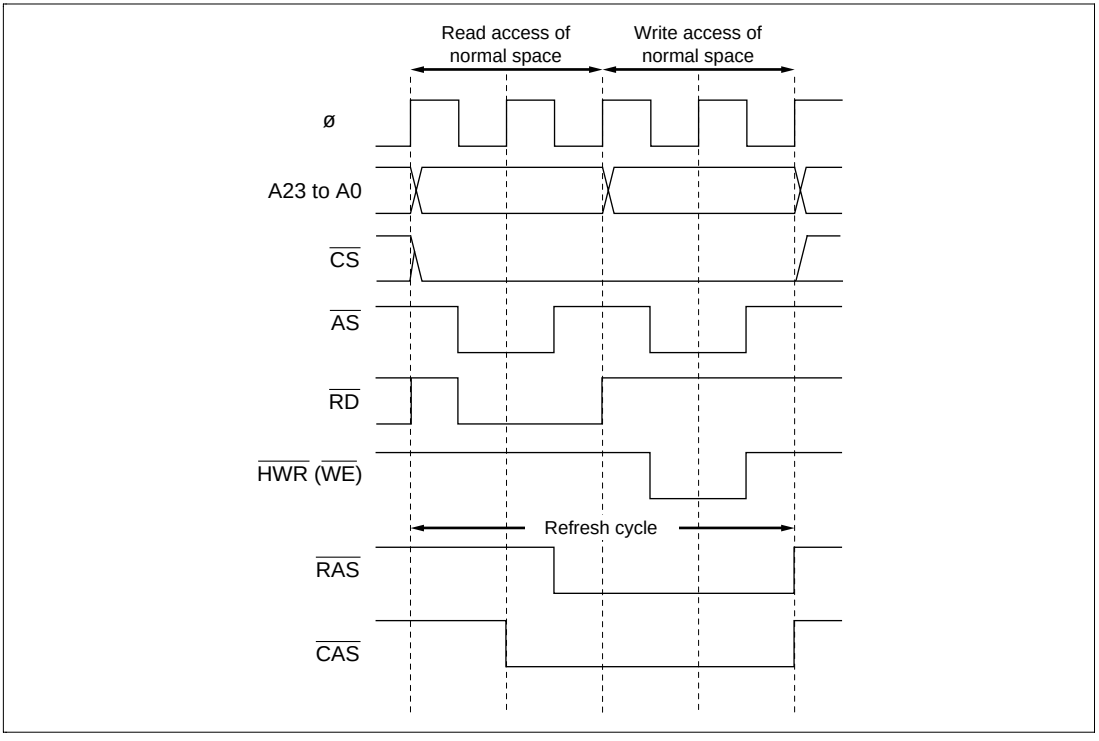


Figure 7-27 Example CBR Refresh Timing (CBRM=0)

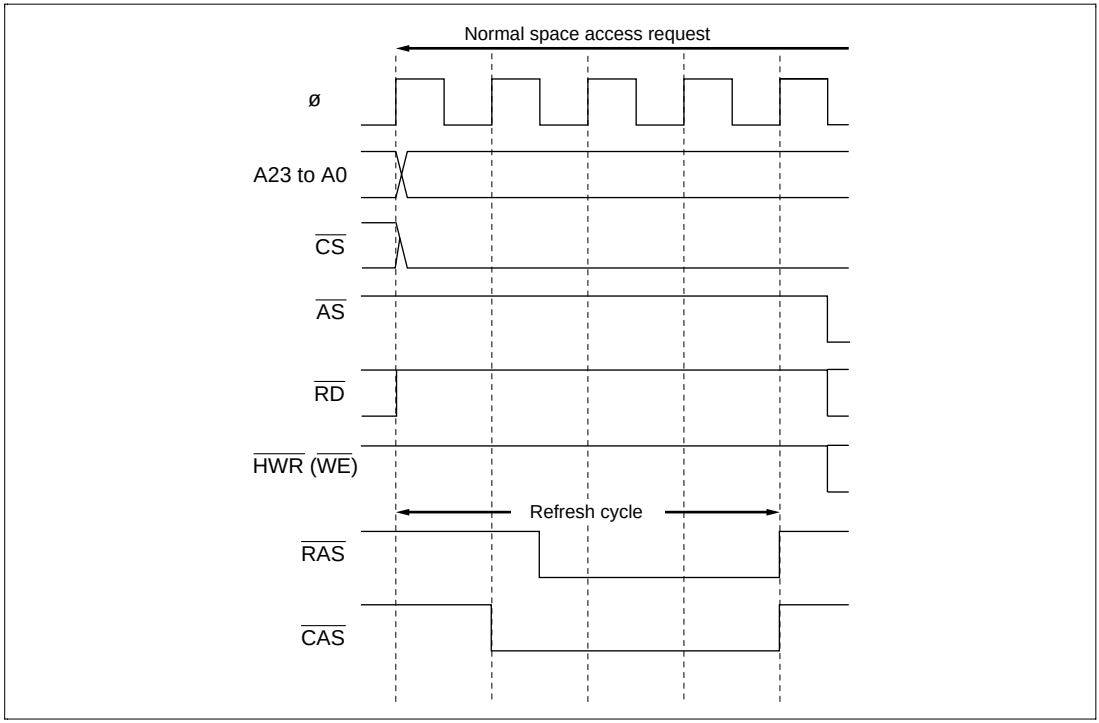


Figure 7-28 Example CBR Refresh Timing (CBRM=1)

(2) Self-Refresh

One of the DRAM standby modes is the self-refresh mode (battery backup mode), in which the DRAM generates its own refresh timing and refresh address.

To select self-refresh, set the RFSHE bit and RMODE bits of the DRAMCR to 1. Next, execute a SLEEP instruction to make a transition to software standby mode. As shown in Figure 7-29, the $\overline{CAS}$ and $\overline{RAS}$ signals are output and the DRAM enters self-refresh mode.

When you exit software standby mode, the RMODE bit is cleared to 0 and self-refresh mode is exited.

When making a transition to software standby mode, self-refresh mode starts after a CBR refresh, providing there is a CBR refresh request. CBR refresh requests occurring immediately before entering software standby mode are cleared on completion of the self-refresh when the software standby mode is exited.

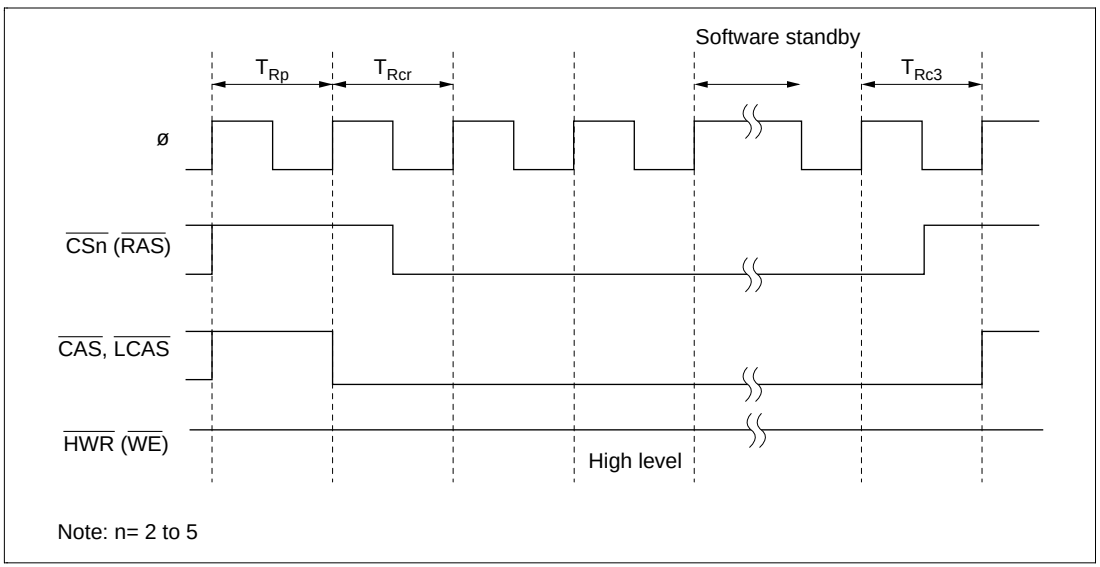


Figure 7-29 Self-Refresh Timing

7.6 DMAC Single Address Mode and DRAM Interface

When burst mode is set for the DRAM interface, the DDS bit selects the output timing for the $\overline{\text{DACK}}$ signal. It also selects whether or not to perform burst access when accessing the DRAM space in DMAC single address mode.

7.6.1 DDS=1

Burst access is performed on the basis of the address only, regardless of the bus master. The $\overline{\text{DACK}}$ output level changes to Low afer the T_{c1} state in the case of the DRAM interface.

Figure 7-30 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS=1.

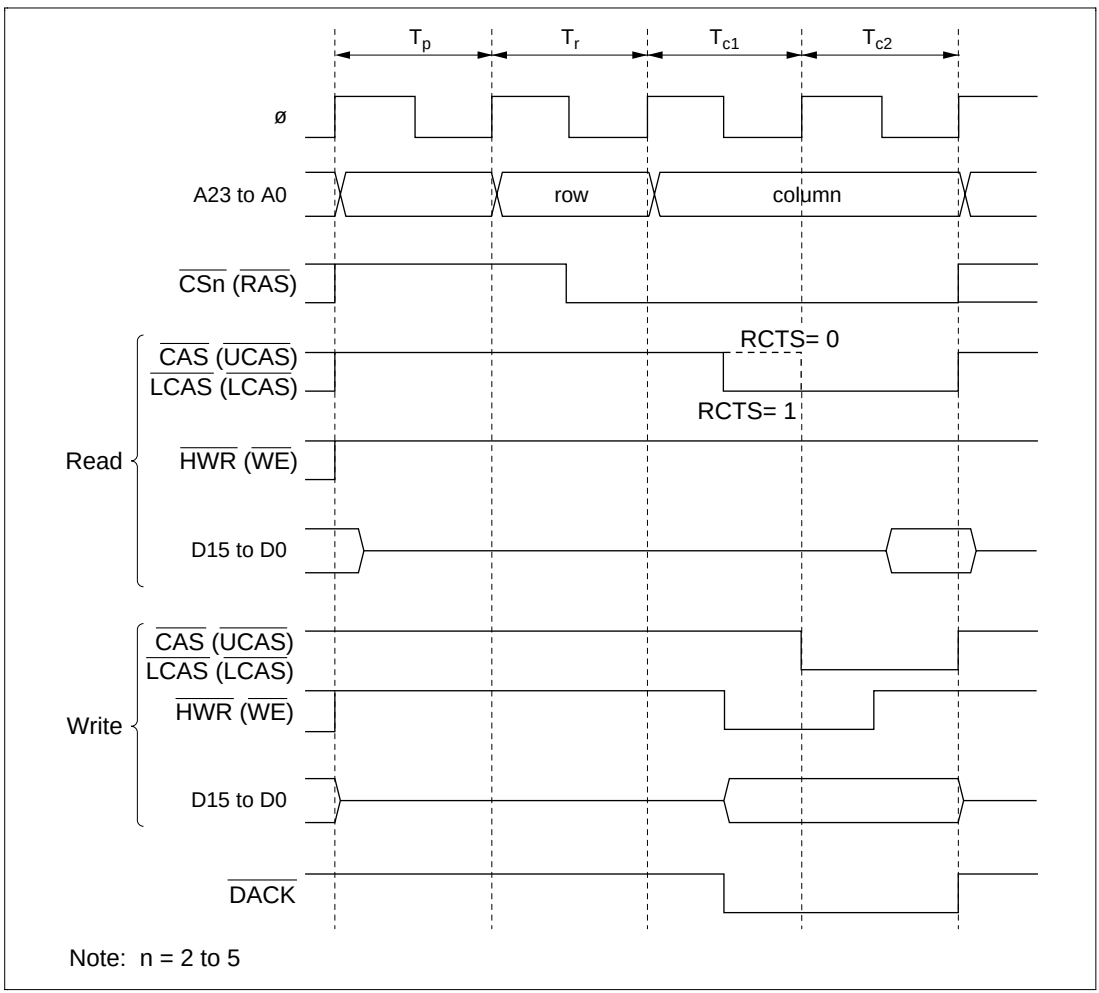


Figure 7-30 $\overline{\text{DACK}}$ Output Timing when DDS=1 (Example Showing DRAM Access)

7.6.2 DDS=0

When the DRAM space is accessed in DMAC single address mode, always perform full access (normal access). The $\overline{\text{DACK}}$ output level changes to Low afer the T_r state in the case of the DRAM interface.

In other than DMAC signle address mode, burst access is possible when the DRAM space is accessed.

Figure 7-31 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS=0.

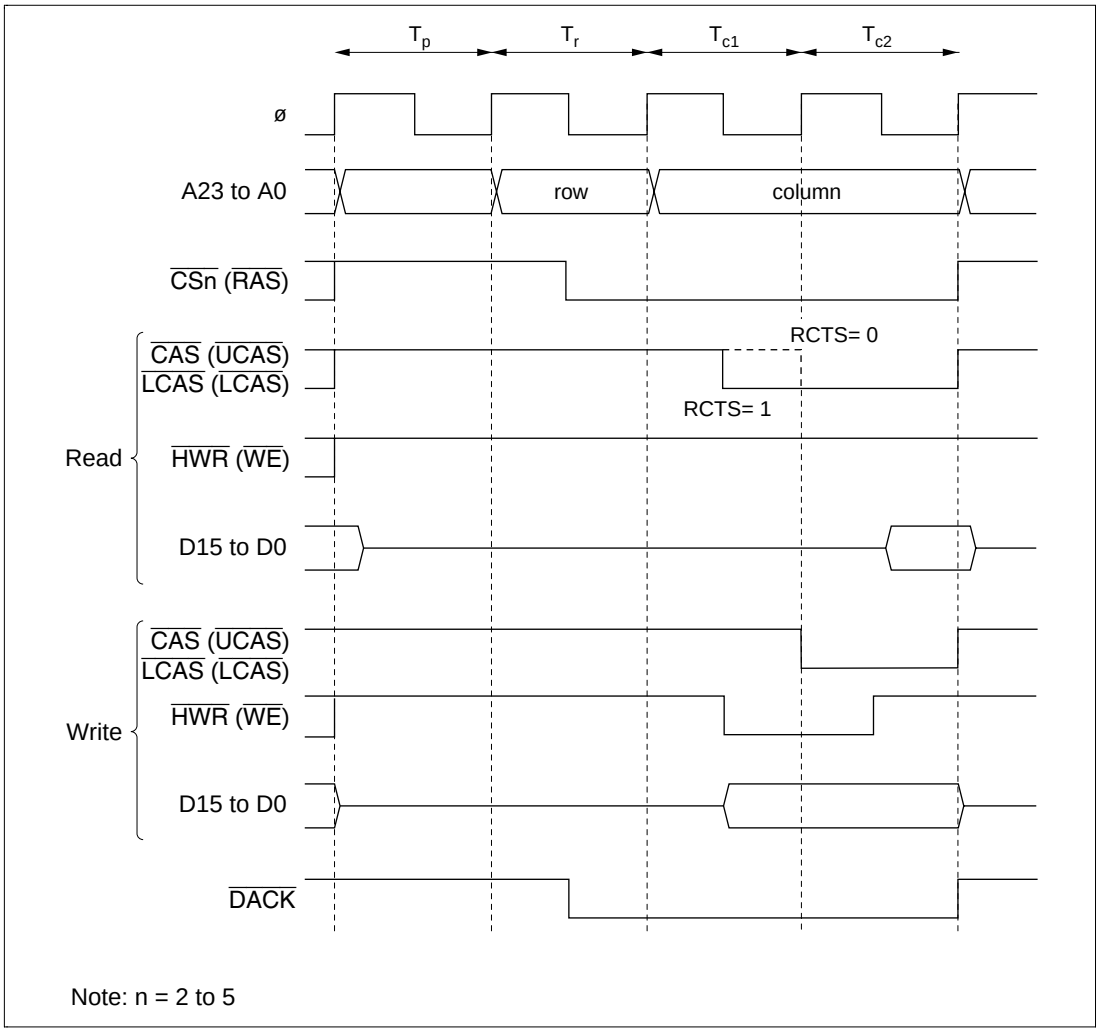


Figure 7-31 $\overline{\text{DACK}}$ Output Timing when DDS=0 (Example Showing DRAM Access)

7.7 Burst ROM Interface

7.7.1 Overview

In this LSI, the area 0 external space can be set as burst ROM space and burst ROM interfacing performed. Burst ROM space interfacing allows 16-bit ROM capable of burst access to be accessed at high-speed.

The BRSTRM bit of BCRH sets area 0 as burst ROM space. CPU instruction fetches (only) can be performed using a maximum of 4-word or 8-word continuous burst access. 1 state or 2 states can be selected in the case of burst access.

7.7.2 Basic Timing

The AST0 bit of ASTCR sets the number of access states in the initial cycle (full access) of the burst ROM interface. Wait states can be inserted when the AST0 bit is set to 1. The burst cycle can be set for 1 state or 2 sttes by setting the BRSTS1 bit of BCRH. Wait states cannot be inserted. When area 0 is set as burst ROM space, area 0 is a 16-bit access space regardless of the ABW0 bit of ABWCR.

When the BRSTS0 bit of BCRH is cleared to 0, 4-word max. burst access is performed. When the BRSTS0 bit is set to 1, 8-word max. burst access is performed.

Figure 7-32 (a) and (b) shows the basic access timing for the burst ROM space.

Figure 7-32 (a) is an example when both the AST0 and BRSTS1 bits are set to 1.

Figure 7-32 (b) is an example when both the AST0 and BRSTS1 bits are set to 0.

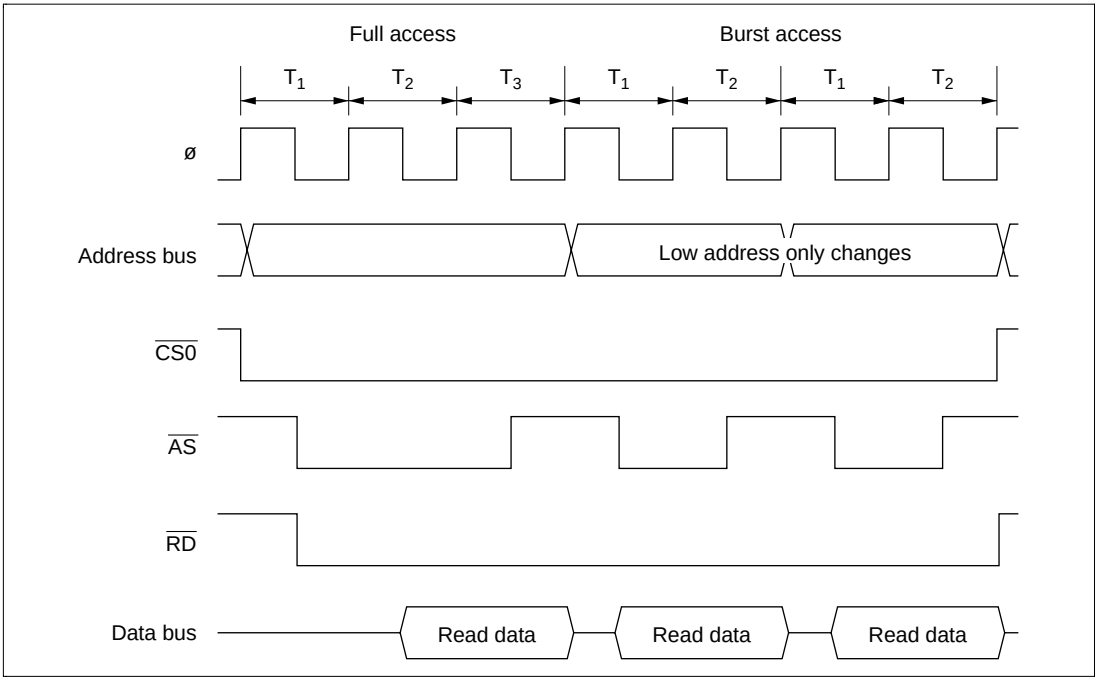


Figure 7-32 (a) Example Burst ROM Access Timing (AST0=BRSTS1=1)

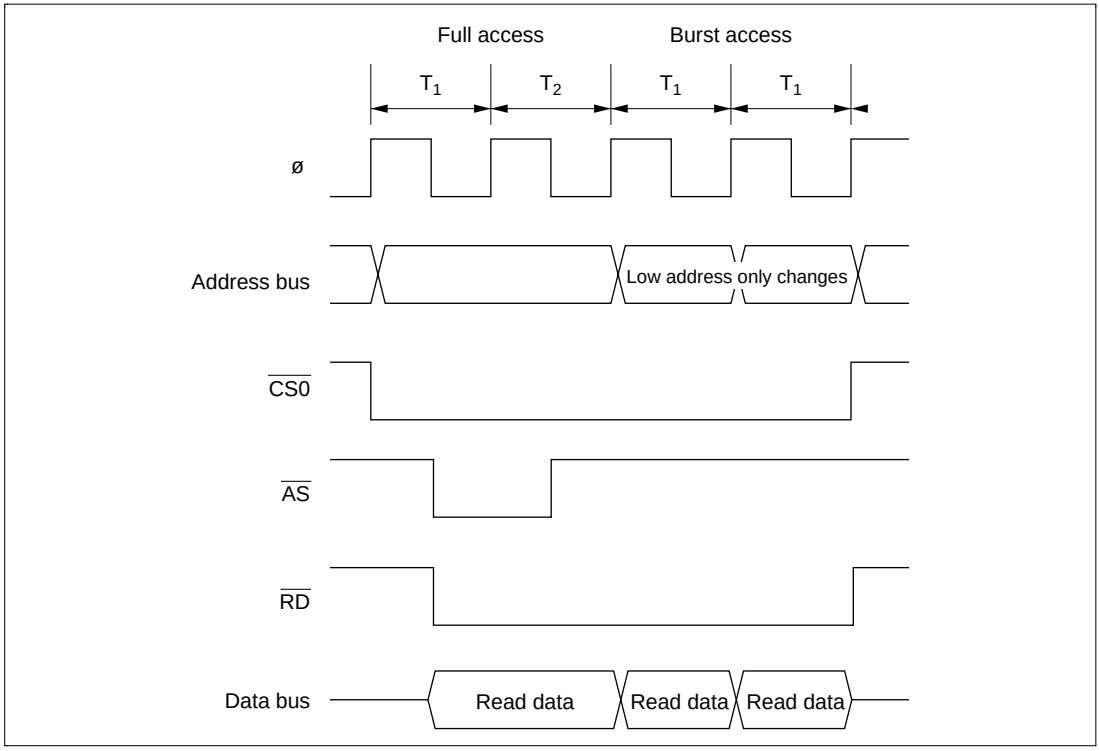


Figure 7-32 (b) Example Burst ROM Access Timing (AST0=BRSTS1=0)

7.7.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) of the burst ROM interface. See Section 7.4.5, Wait Control.

Wait states cannot be inserted in the burst cycle.

7.8 Idle Cycle

7.8.1 Operation

When the H8S/2633 Series accesses external space , it can insert a 1-state idle cycle (T_I) between bus cycles in the following two cases: (1) when read accesses between different areas occur consecutively, and (2) when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, with a long output floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads between Different Areas

If consecutive reads between different areas occur while the ICIS1 bit in BCRH is set to 1, an idle cycle is inserted at the start of the second read cycle.

Figure 7-33 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

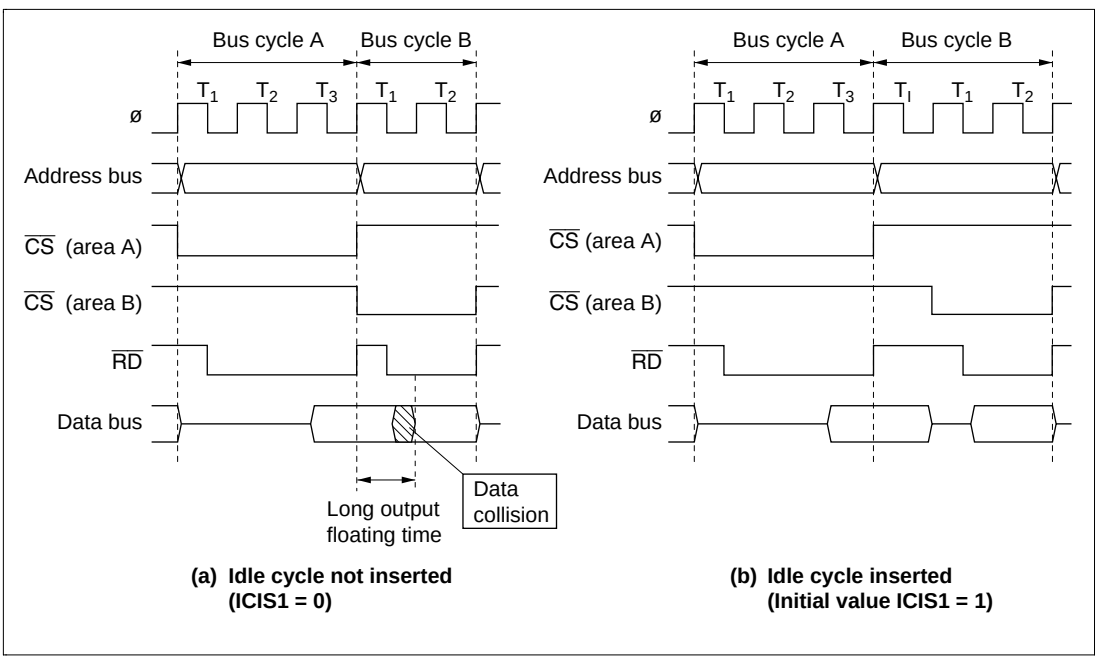


Figure 7-33 Example of Idle Cycle Operation (1)

(2) Write after Read

If an external write occurs after an external read while the ICIS0 bit in BCRH is set to 1, an idle cycle is inserted at the start of the write cycle.

Figure 7-34 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

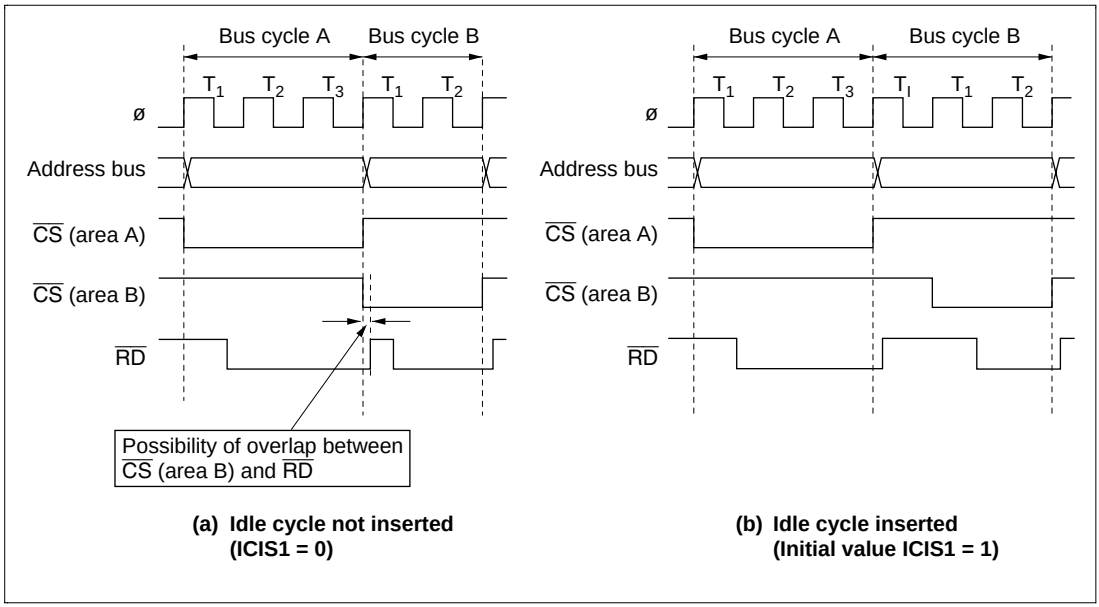


Figure 7-34 Example of Idle Cycle Operation (2)

(3) Relationship between Chip Select ($\overline{CS}$) Signal and Read ($\overline{RD}$) Signal

Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal. An example is shown in figure 7-35.

In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal.

Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals.

In the initial state after reset release, idle cycle insertion (b) is set.

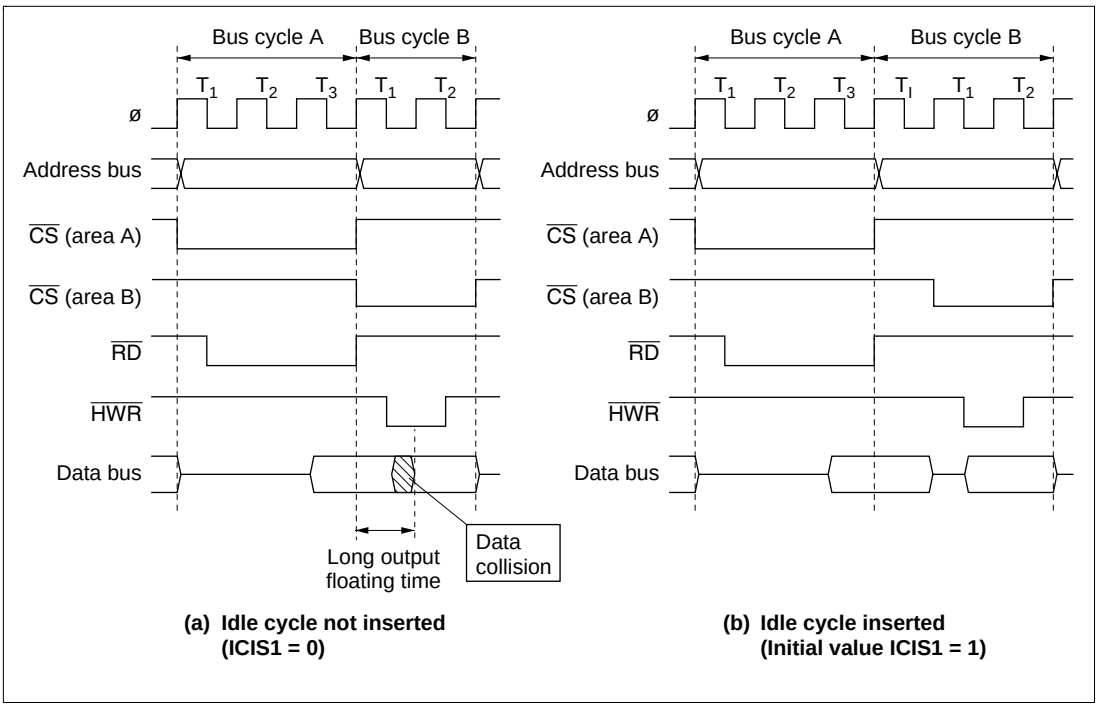


Figure 7-35 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

(4) Notes

The setting of the ICIS0 and ICIS1 bits is invalid when accessing the DRAM space. For example, if the 2nd of successive reads of different areas is a DRAM access, only the T_p cycle is inserted, not the T_1 cycle. Figure 7-36 shows the timing. Note, however, that ICIS0 and ICIS1 settings are valid in burst access in RAS down mode, and an idle cycle is inserted. Figure 7-37 (a) and (b) shows the timing.

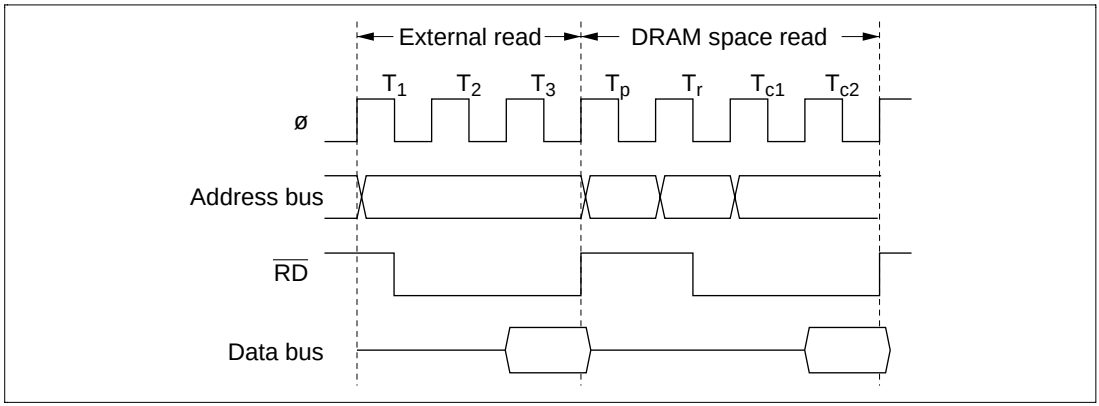


Figure 7-36 Example of DRAM Access after External Read

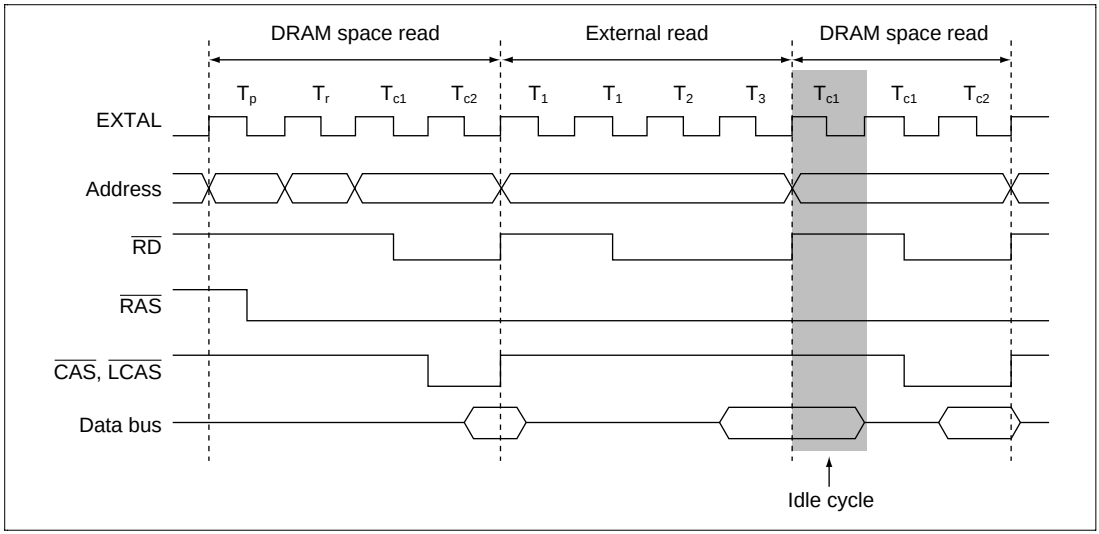


Figure 7-37 (a) Example Idle Cycle Operation in RAS Down Mode (ICIS1=1)

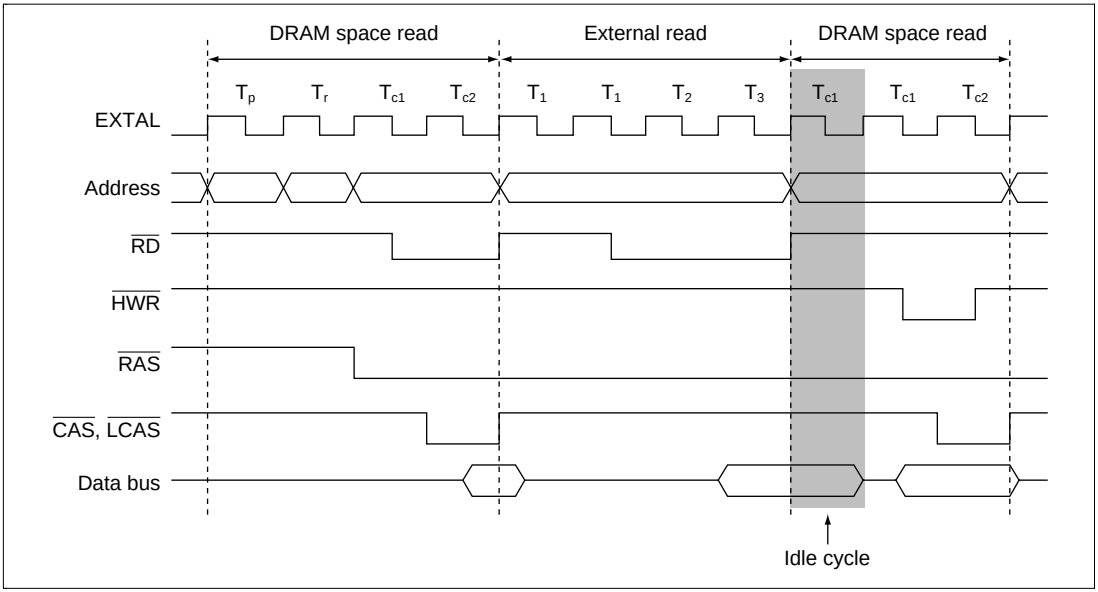


Figure 7-37 (b) Example Idle Cycle Operation in RAS Down Mode (ICIS0=1)

7.8.2 Pin States in Idle Cycle

Table 7-8 shows pin states in an idle cycle.

Table 7-8 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of next bus cycle
D15 to D0	High impedance
$\overline{\text{CSn}}$	High*
$\overline{\text{CAS}}$	High
$\overline{\text{AS}}$	High
$\overline{\text{RD}}$	High
$\overline{\text{HWR}}$	High
$\overline{\text{LWR}}$	High
$\overline{\text{DACKn}}$	High

Note: * Remains low in DRAM space RAS down mode or a refresh cycle.

7.9 Write Data Buffer Function

The H8S/2633 Series has a write data buffer function in the external data bus. Using the write data buffer function enables external writes and DMA single address mode transmission to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit in BCRL to 1.

Figure 7-38 shows an example of the timing when the write data buffer function is used. When this function is used, if an external write and DMA single address mode transmission continues for 2 states or longer, and there is an internal access next, only an external write is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external write rather than waiting until it ends.

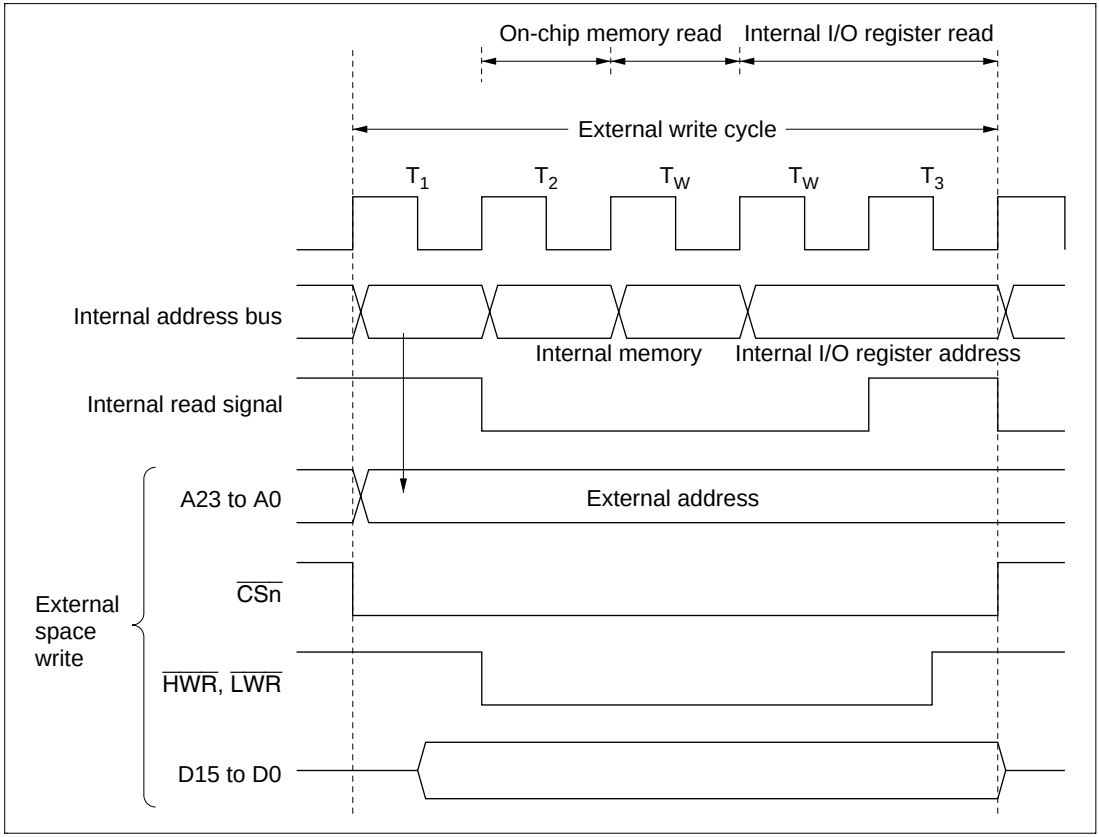


Figure 7-38 Example of Timing when Write Data Buffer Function is Used

7.10 Bus Release

7.10.1 Overview

The H8S/2633 Series can release the external bus in response to a bus request from an external device. In the external bus released state, the internal bus master continues to operate as long as there is no external access.

If an internal bus master wants to make an external access and when a refresh request occurs in the external bus released state, it can issue a bus request off-chip.

7.10.2 Operation

In external expansion mode, the bus can be released to an external device by setting the BRLE bit in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus request to the H8S/2633 Series. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescribed timing the $\overline{\text{BACK}}$ pin is driven low, and the address bus, data bus, and bus control signals are placed in the high-impedance state, establishing the external bus-released state.

In the external bus released state, an internal bus master can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers activation of the bus cycle, and waits for the bus request from the external bus master to be dropped. Also, when a refresh request occurs in the external bus released state, refresh control is deferred until the external bus master drops the bus request.

If the BREQOE bit in BCRL is set to 1, when an internal bus master wants to make an external access and when a refresh request occurs in the external bus released state, the $\overline{\text{BREQO}}$ pin is driven low and a request can be made off-chip to drop the bus request.

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timing and the external bus released state is terminated.

The following shows the order of priority when an external bus release request, refresh request, and external access by the internal bus master occur simultaneously:

When CBRM=1

(High) Refresh > External bus release > External access by internal bus master (Low)

When CBRM=0

(High) Refresh > External bus release (Low)

(High) External bus release > External access by internal bus master (Low)

Note: A refresh can be executed at the same time as external access by the internal bus master.

7.10.3 Pin States in External Bus Released State

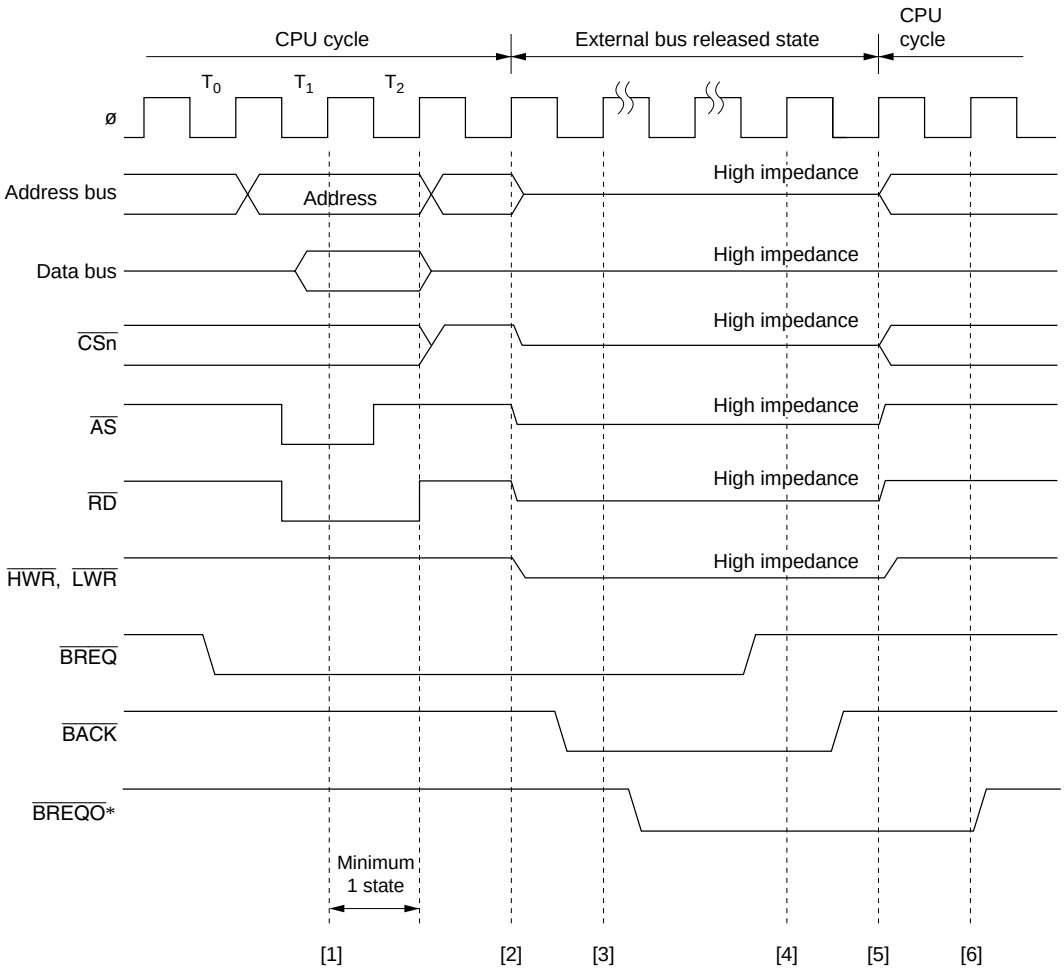
Table 7-9 shows pin states in the external bus released state.

Table 7-9 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$	High impedance
$\overline{\text{CAS}}$	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$	High impedance
$\overline{\text{LWR}}$	High impedance
$\overline{\text{DACKn}}$	High

7.10.4 Transition Timing

Figure 7-39 shows the timing for transition to the bus-released state.



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at rise of T_2 state.

[2] $\overline{\text{BACK}}$ pin is driven low at end of CPU read cycle, releasing bus to external bus master.

[3] $\overline{\text{BREQ}}$ pin state is still sampled in external bus released state.

[4] High level of $\overline{\text{BREQ}}$ pin is sampled.

[5] $\overline{\text{BACK}}$ pin is driven high, ending bus release cycle.

[6] $\overline{\text{BREQO}}$ signal goes high 1.5 clocks after $\overline{\text{BACK}}$ signal goes high.

Note: * Output only when BREQOE is set to 1.

Figure 7-39 Bus-Released State Transition Timing

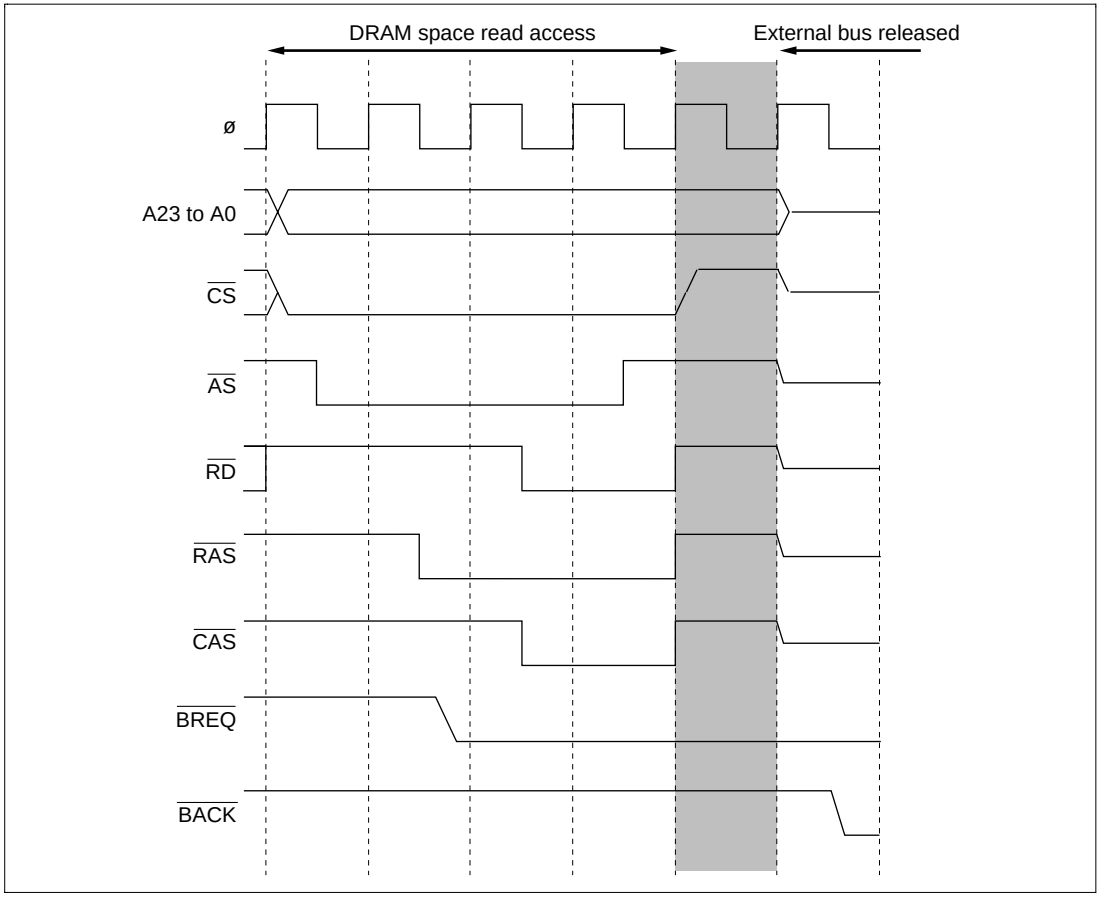


Figure 7-40 Example Bus Release Transition Timing After DRAM Access (Reading DRAM)

7.10.5 Notes

The external bus release function is deactivated when MSTPCR is set to H'FFFFFF or H'EFFFFF and a transition is made to sleep mode. To use the external bus release function in sleep mode, do not set MSTPCR to H'FFFFFF and H'EFFFFF.

When the CBRM bit is set to 1 to use the CBR refresh function, set the $\overline{\text{BREQ}}$=1 width greater than the number of the slowest external access states. Otherwise, CBR refresh requests from the refresh timer may not be performed.

7.11 Bus Arbitration

7.11.1 Overview

The H8S/2633 Series has a bus arbiter that arbitrates bus master operations.

There are two bus masters, the CPU, DTC, and DMAC which perform read/write operations when they have possession of the bus. Each bus master requests the bus by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus request acknowledge signal. The selected bus master then takes possession of the bus and begins its operation.

7.11.2 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master making the request. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DMAC > DTC > CPU (Low)

An internal bus access by an internal bus master, external bus release, and refresh can be executed in parallel.

In the event of simultaneous external bus release request, refresh request, and internal bus master external access request generation, the order of priority is as follows:

When CBRM=1

(High) Refresh > External bus release > External access by internal bus master (Low)

When CBRM=0

(High) Refresh > External bus release (Low)

(High) External bus release > External access by internal bus master (Low)

7.11.3 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific times at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the operations. See Appendix A-5, Bus States During Instruction Execution, for timings at which the bus is not transferred.
- If the CPU is in sleep mode, it transfers the bus immediately.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

DMAC: When a start request occurs, the DMAC requests the bus arbiter for bus privileges.

The DMAC releases bus privileges on completion of one transmission in short address mode, normal mode external requests, and cycle steal mode.

The DMAC releases the bus on completion of the transmission of one block in block transmission mode, or after a transmission in burst mode.

7.12 Resets and the Bus Controller

In a power-on reset, the H8S/2633 Series, including the bus controller, enters the reset state at that point, and an executing bus cycle is discontinued.

The bus controller registers and internal states are retained at a manual reset. The current external bus cycle is executed to completion. The $\overline{\text{WAIT}}$ input is ignored. Write data is not retained. Also, because the DMAC is initialized at a manual reset, $\overline{\text{DACK}}$ and $\overline{\text{TEND}}$ outputs are disabled and function as I/O ports controlled by DDR and DR.

Section 8 DMA Controller

8.1 Overview

The H8S/2633 Series has a built-in DMA controller (DMAC) which can carry out data transfer on up to 4 channels.

8.1.1 Features

The features of the DMAC are listed below.

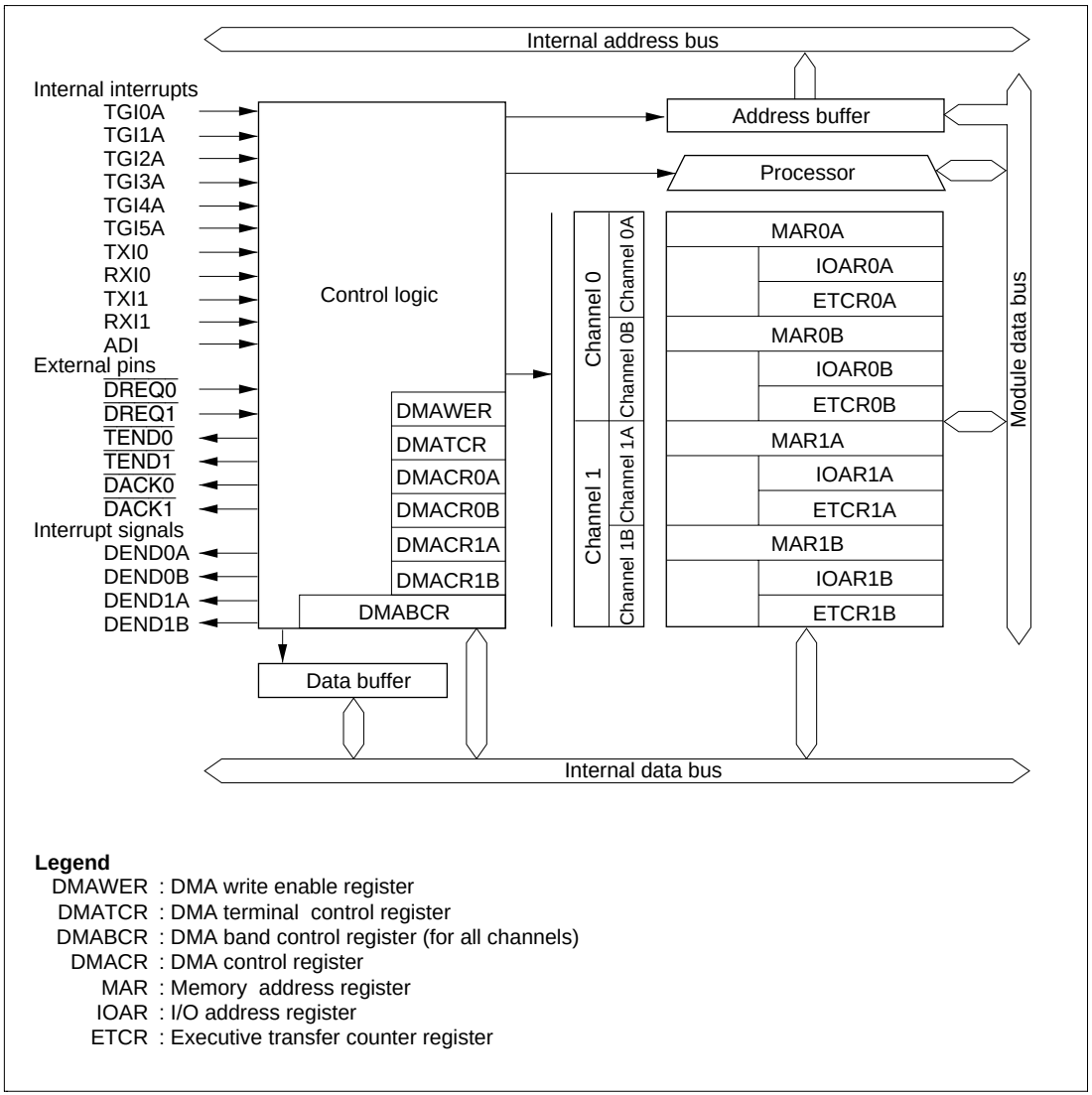
- Choice of short address mode or full address mode
  Short address mode
  - Maximum of 4 channels can be used
  - Choice of dual address mode or single address mode
  - In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as16 bits
  - In single address mode, transfer source or transfer destination address only is specified as 24 bits
  - In single address mode, transfer can be performed in one bus cycle
  - Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode

  Full address mode
  - Maximum of 2 channels can be used
  - Transfer source and transfer destination address specified as 24 bits
  - Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)
  - Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  - Serial communication interface (SCI0, SCI1) transmit-data-empty interrupt, reception complete interrupt
  - A/D converter conversion end interrupt
  - External request
  - Auto-request

- Module stop mode can be set
  — The initial setting enables DMAC registers to be accessed. DMAC operation is halted by setting module stop mode

8.1.2 Block Diagram

A block diagram of the DMAC is shown in figure 8-1.



Legend

DMAWER : DMA write enable register
DMATCR : DMA terminal control register
DMABCR : DMA band control register (for all channels)
DMACR : DMA control register
MAR : Memory address register
IOAR : I/O address register
ETCR : Executive transfer counter register

Figure 8-1 Block Diagram of DMAC

8.1.3 Overview of Functions

Tables 8-1 (1) and (2) summarize DMAC functions in short address mode and full address mode, respectively.

Table 8-1 (1) Overview of DMAC Functions (Short Address Mode)

		Address Register Bit Length	
Transfer Mode	**Transfer Source**	**Source**	**Destination**
Dual address mode • Sequential mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/decremented by 1 or 2 — 1 to 65536 transfers • Idle mode — 1-byte or 1-word transfer executed for one transfer request — Memory address fixed — 1 to 65536 transfers • Repeat mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/ decremented by 1 or 2 — After specified number of transfers (1 to 256), initial state is restored and operation continues	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	24/16	16/24
Single address mode • 1-byte or 1-word transfer executed for one transfer request • Transfer in 1 bus cycle using $\overline{\text{DACK}}$ pin in place of address specifying I/O • Specifiable for sequential, idle, and repeat modes	• External request	24/$\overline{\text{DACK}}$	$\overline{\text{DACK}}$/24

Table 8-1 (2) Overview of DMAC Functions (Full Address Mode)

Transfer Mode	Transfer Source	Address Register Bit Length	
		Source	Destination
• Normal mode Auto-request — Transfer request retained internally — Transfers continue for the specified number of times (1 to 65536) — Choice of burst or cycle steal transfer	• Auto-request	24	24
External request — 1-byte or 1-word transfer executed for one transfer request — 1 to 65536 transfers	• External request		
• Block transfer mode — Specified block size transfer executed for one transfer request — 1 to 65536 transfers — Either source or destination specifiable as block area — Block size: 1 to 256 bytes or words	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI reception complete interrupt • External request • A/D converter conversion end interrupt	24	24

8.1.4 Pin Configuration

Table 8-2 summarizes the DMAC pins.

In short address mode, external request transfer, single address transfer, and transfer end output are not performed for channel A.

The DMA transfer acknowledge function is used in channel B single address mode in short address mode.

When the $\overline{\text{DREQ}}$ pin is used, do not designate the corresponding port for output.

With regard to the $\overline{\text{DACK}}$ pins, setting single address transfer automatically sets the corresponding port to output, functioning as a $\overline{\text{DACK}}$ pin.

With regard to the $\overline{\text{TEND}}$ pins, whether or not the corresponding port is used as a $\overline{\text{TEND}}$ pin can be specified by means of a register setting.

Table 8-2 DMAC Pins

Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	$\overline{\text{DREQ0}}$	Input	DMAC channel 0 external request
	DMA transfer acknowledge 0	$\overline{\text{DACK0}}$	Output	DMAC channel 0 single address transfer acknowledge
	DMA transfer end 0	$\overline{\text{TEND0}}$	Output	DMAC channel 0 transfer end
1	DMA request 1	$\overline{\text{DREQ1}}$	Input	DMAC channel 1 external request
	DMA transfer acknowledge 1	$\overline{\text{DACK1}}$	Output	DMAC channel 1 single address transfer acknowledge
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	DMAC channel 1 transfer end

8.1.5 Register Configuration

Table 8-3 summarizes the DMAC registers.

Table 8-3 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*	Bus Width
0	Memory address register 0A	MAR0A	R/W	Undefined	H'FEE0	16 bits
	I/O address register 0A	IOAR0A	R/W	Undefined	H'FEE4	16 bits
	Transfer count register 0A	ETCR0A	R/W	Undefined	H'FEE6	16 bits
	Memory address register 0B	MAR0B	R/W	Undefined	H'FEE8	16 bits
	I/O address register 0B	IOAR0B	R/W	Undefined	H'FEEC	16 bits
	Transfer count register 0B	ETCR0B	R/W	Undefined	H'FEEE	16 bits
1	Memory address register 1A	MAR1A	R/W	Undefined	H'FEF0	16 bits
	I/O address register 1A	IOAR1A	R/W	Undefined	H'FEF4	16 bits
	Transfer count register 1A	ETCR1A	R/W	Undefined	H'FEF6	16 bits
	Memory address register 1B	MAR1B	R/W	Undefined	H'FEF8	16 bits
	I/O address register 1B	IOAR1B	R/W	Undefined	H'FEFC	16 bits
	Transfer count register 1B	ETCR1B	R/W	Undefined	H'FEFE	16 bits
0, 1	DMA write enable register	DMAWER	R/W	H'00	H'FF60	8 bits
	DMA terminal control register	DMATCR	R/W	H'00	H'FF61	8 bits
	DMA control register 0A	DMACR0A	R/W	H'00	H'FF62	16 bits
	DMA control register 0B	DMACR0B	R/W	H'00	H'FF63	16 bits
	DMA control register 1A	DMACR1A	R/W	H'00	H'FF64	16 bits
	DMA control register 1B	DMACR1B	R/W	H'00	H'FF65	16 bits
	DMA band control register	DMABCR	R/W	H'0000	H'FF66	16 bits
	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8	8 bits

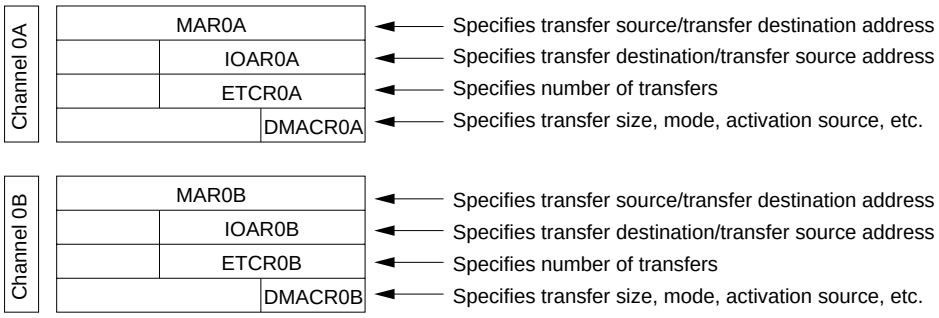
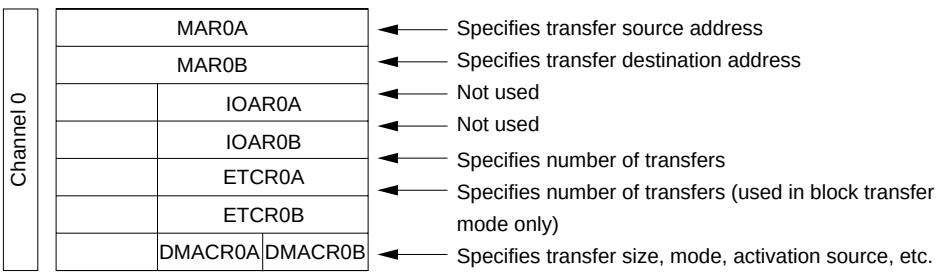
Note: * Lower 16 bits of the address.

8.2 Register Descriptions (1) (Short Address Mode)

Short address mode transfer can be performed for channels A and B independently.

Short address mode transfer is specified for each channel by clearing the FAE bit in DMABCR to 0, as shown in table 8-4. Short address mode or full address mode can be selected for channels 1 and 0 independently by means of bits FAE1 and FAE0.

Table 8-4 Short Address Mode and Full Address Mode (For 1 Channel: Example of Channel 0)

FAE0	Description
0	Short address mode specified (channels A and B operate independently) 
1	Full address mode specified (channels A and B operate in combination) 

8.2.1 Memory Address Registers (MAR)

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

MAR is a 32-bit readable/writable register that specifies the transfer source address or destination address.

The upper 8 bits of MAR are reserved: they are always read as 0, and cannot be modified.

Whether MAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the address specified by MAR is constantly updated. For details, see section 8.2.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

8.2.2 I/O Address Register (IOAR)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the transfer source address or destination address. The upper 8 bits of the transfer address are automatically set to H'FF.

Whether IOAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

IOAR is invalid in single address mode.

IOAR is not incremented or decremented each time a transfer is executed, so that the address specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

8.2.3 Execute Transfer Count Register (ETCR)

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The setting of this register is different for sequential mode and idle mode on the one hand, and for repeat mode on the other.

(1) Sequential Mode and Idle Mode

Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter (with a count range of 1 to 65536). ETCR is decremented by 1 each time a transfer is performed, and when the count reaches H'0000, the DTE bit in DMABCR is cleared, and transfer ends.

(2) Repeat Mode

Transfer Number Storage

Bit	:	15	14	13	12	11	10	9	8
ETCRH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer Counter

Bit	:	7	6	5	4	3	2	1	0
ETCRL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In repeat mode, ETCR functions as transfer counter ETCRL (with a count range of 1 to 256) and transfer number storage register ETCRH. ETCRL is decremented by 1 each time a transfer is performed, and when the count reaches H'00, ETCRL is loaded with the value in ETCRH. At this point, MAR is automatically restored to the value it had when the count was started. The DTE bit in DMABCR is not cleared, and so transfers can be performed repeatedly until the DTE bit is cleared by the user.

ETCR is not initialized by a reset or in standby mode.

8.2.4 DMA Control Register (DMACR)

Bit	:	7	6	5	4	3	2	1	0
DMACR	:	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMACR is an 8-bit readable/writable register that controls the operation of each DMAC channel.

DMACR is initialized to H'00 by a reset, and in standby mode.

Bit 7—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 7		
DTSZ	**Description**	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 6—Data Transfer Increment/Decrement (DTID): Selects incrementing or decrementing of MAR every data transfer in sequential mode or repeat mode.

In idle mode, MAR is neither incremented nor decremented.

Bit 6		
DTID	**Description**	
0	MAR is incremented after a data transfer • When DTSZ = 0, MAR is incremented by 1 after a transfer • When DTSZ = 1, MAR is incremented by 2 after a transfer	(Initial value)
1	MAR is decremented after a data transfer • When DTSZ = 0, MAR is decremented by 1 after a transfer • When DTSZ = 1, MAR is decremented by 2 after a transfer	

Bit 5—Repeat Enable (RPE): Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed.

Bit 5	DMABCR		
RPE	**DTIE**	**Description**	
0	0	Transfer in sequential mode (no transfer end interrupt)	(Initial value)
	1	Transfer in sequential mode (with transfer end interrupt)	
1	0	Transfer in repeat mode (no transfer end interrupt)	
	1	Transfer in idle mode (with transfer end interrupt)	

For details of operation in sequential, idle, and repeat mode, see section 8.5.2, Sequential Mode, section 8.5.3, Idle Mode, and section 8.5.4, Repeat Mode.

Bit 4—Data Transfer Direction (DTDIR): Used in combination with the SAE bit in DMABCR to specify the data transfer direction (source or destination). The function of this bit is therefore different in dual address mode and single address mode.

DMABCR	Bit 4	
SAE	DTDIR	Description
0	0	Transfer with MAR as source address and IOAR as destination address (Initial value)
	1	Transfer with IOAR as source address and MAR as destination address
1	0	Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe
	1	Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transfer factor (activation source). There are some differences in activation sources for channel A and for channel B.

Channel A

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	DTF2	DTF1	DTF0	Description
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	—
			1	—
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Channel B

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 8.5.13, DMAC Multi-Channel Operation.

8.2.5 DMA Band Control Register (DMABCR)

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In short address mode, channels 1A and 1B are used as independent channels.

Bit 15

FAE1	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in short address mode or full address mode.

In short address mode, channels 0A and 0B are used as independent channels.

Bit 14

FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 13—Single Address Enable 1 (SAE1): Specifies whether channel 1B is to be used for transfer in dual address mode or single address mode.

Bit 13

SAE1	Description	
0	Transfer in dual address mode	(Initial value)
1	Transfer in single address mode	

This bit is invalid in full address mode.

Bit 12—Single Address Enable 0 (SAE0): Specifies whether channel 0B is to be used for transfer in dual address mode or single address mode.

Bit 12

SAE0	Description	
0	Transfer in dual address mode	(Initial value)
1	Transfer in single address mode	

This bit is invalid in full address mode.

Bits 11 to 8—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

When DTE = 1 and DTA = 0, the internal interrupt source selected by the data transfer factor setting is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE = 0, the internal interrupt source selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

Bit 11—Data Transfer Acknowledge 1B (DTA1B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1B data transfer factor setting.

Bit 11

DTA1B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 10—Data Transfer Acknowledge 1A (DTA1A): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1A data transfer factor setting.

Bit 10

DTA1A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 9—Data Transfer Acknowledge 0B (DTA0B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0B data transfer factor setting.

Bit 9

DTA0B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 8—Data Transfer Acknowledge 0A (DTA0A): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0A data transfer factor setting.

Bit 8

DTA0A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 7 to 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE bit is set to 1

when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed in a transfer mode other than repeat mode
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reason

When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 7—Data Transfer Enable 1B (DTE1B): Enables or disables data transfer on channel 1B.

Bit 7

DTE1B	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 6—Data Transfer Enable 1A (DTE1A): Enables or disables data transfer on channel 1A.

Bit 6

DTE1A	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 5—Data Transfer Enable 0B (DTE0B): Enables or disables data transfer on channel 0B.

Bit 5

DTE0B	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4—Data Transfer Enable 0A (DTE0A): Enables or disables data transfer on channel 0A.

Bit 4

DTE0A	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 to 0—Data Transfer End Interrupt Enable (DTIE): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Bit 3—Data Transfer End Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1B transfer end interrupt.

Bit 3

DTIE1B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 2—Data Transfer End Interrupt Enable 1A (DTIE1A): Enables or disables the channel 1A transfer end interrupt.

Bit 2

DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 1—Data Transfer End Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0B transfer end interrupt.

Bit 1

DTIE0B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer End Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0A transfer end interrupt.

Bit 0		
DTIE0A	**Description**	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

8.3 Register Descriptions (2) (Full Address Mode)

Full address mode transfer is performed with channels A and B together. For details of full address mode setting, see table 8-4.

8.3.1 Memory Address Register (MAR)

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

MAR is a 32-bit readable/writable register; MARA functions as the transfer source address register, and MARB as the destination address register.

MAR is composed of two 16-bit registers, MARH and MARL. The upper 8 bits of MARH are reserved: they are always read as 0, and cannot be modified.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the source or destination memory address can be updated automatically. For details, see section 8.3.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

8.3.2 I/O Address Register (IOAR)

IOAR is not used in full address transfer.

8.3.3 Execute Transfer Count Register (ETCR)

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The function of this register is different in normal mode and in block transfer mode.

ETCR is not initialized by a reset or in standby mode.

(1) Normal Mode

ETCRA

Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used at this time.

ETCRB

ETCRB is not used in normal mode.

(2) Block Transfer Mode

ETCRA

Holds block size

Bit	:	15	14	13	12	11	10	9	8
ETCRAH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block size counter

Bit	:	7	6	5	4	3	2	1	0
ETCRAL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

ETCRB

Block Transfer Counter

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCRB																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH holds the block size. ETCRAL is decremented each time a 1-byte or 1-word transfer is performed, and when the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting the block size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting of any desired number of bytes or words.

ETCRB functions in block transfer mode, as a 16-bit block transfer counter. ETCRB is decremented by 1 each time a block is transferred, and transfer ends when the count reaches H'0000.

8.3.4 DMA Control Register (DMACR)

DMACR is a 16-bit readable/writable register that controls the operation of each DMAC channel. In full address mode, DMACRA and DMACRB have different functions.

DMACR is initialized to H'0000 by a reset, and in standby mode.

DMACRA

Bit	15	14	13	12	11	10	9	8
DMACRA	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMACRB

Bit	7	6	5	4	3	2	1	0
DMACRB	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 15		
DTSZ	**Description**	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 14—Source Address Increment/Decrement (SAID)

Bit 13—Source Address Increment/Decrement Enable (SAIDE): These bits specify whether source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 14	Bit 13	
SAID	**SAIDE**	**Description**
0	0	MARA is fixed (Initial value)
	1	MARA is incremented after a data transfer • When DTSZ = 0, MARA is incremented by 1 after a transfer • When DTSZ = 1, MARA is incremented by 2 after a transfer
1	0	MARA is fixed
	1	MARA is decremented after a data transfer • When DTSZ = 0, MARA is decremented by 1 after a transfer • When DTSZ = 1, MARA is decremented by 2 after a transfer

Bit 12—Block Direction (BLKDIR)

Bit 11—Block Enable (BLKE): These bits specify whether normal mode or block transfer mode is to be used. If block transfer mode is specified, the BLKDIR bit specifies whether the source side or the destination side is to be the block area.

Bit 12	Bit 11	
BLKDIR	**BLKE**	**Description**
0	0	Transfer in normal mode (Initial value)
	1	Transfer in block transfer mode, destination side is block area
1	0	Transfer in normal mode
	1	Transfer in block transfer mode, source side is block area

For operation in normal mode and block transfer mode, see section 8.5, Operation.

Bits 10 to 7—Reserved: Can be read or written to.

Bit 6—Destination Address Increment/Decrement (DAID)

Bit 5—Destination Address Increment/Decrement Enable (DAIDE): These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 6	Bit 5		
DAID	**DAIDE**	**Description**	
0	0	MARB is fixed	(Initial value)
	1	MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 after a transfer • When DTSZ = 1, MARB is incremented by 2 after a transfer	
1	0	MARB is fixed	
	1	MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 after a transfer • When DTSZ = 1, MARB is decremented by 2 after a transfer	

Bit 4—Reserved: Can be read or written to.

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transfer factor (activation source). The factors that can be specified differ between normal mode and block transfer mode.

- Normal Mode

Bit 3	Bit 2	Bit 1	Bit 0		
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**	
0	0	0	0	—	(Initial value)
			1	—	
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input	
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	
	1	0	*	—	
		1	0	Auto-request (cycle steal)	
			1	Auto-request (burst)	
1	*	*	*	—	

*: Don't care

- Block Transfer Mode

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 8.5.13, DMAC Multi-Channel Operation.

8.3.5 DMA Band Control Register (DMABCR)

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	—	—	DTA1	—	DTA0	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In full address mode, channels 1A and 1B are used together as a single channel.

Bit 15

FAE1	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in short address mode or full address mode.

In full address mode, channels 0A and 0B are used together as a single channel.

Bit 14

FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bits 13 and 12—Reserved: Can be read or written to.

Bits 11 and 9—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

When the DTE = 1 and the DTA = 0, the internal interrupt source selected by the data transfer factor setting is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When the DTE = 0, the internal interrupt source selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

The state of the DTME bit does not affect the above operations.

Bit 11—Data Transfer Acknowledge 1 (DTA1): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1 data transfer factor setting.

Bit 11

DTA1	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 9—Data Transfer Acknowledge 0 (DTA0): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0 data transfer factor setting.

Bit 9

DTA0	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 10 and 8—Reserved: Can be read or written to.

Bits 7 and 5—Data Transfer Master Enable (DTME): Together with the DTE bit, these bits control enabling or disabling of data transfer on the relevant channel. When both the DTME bit and the DTE bit are set to 1, transfer is enabled for the channel.

If the relevant channel is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME bit is not cleared by an NMI interrupt, and transfer is not interrupted.

The conditions for the DTME bit being cleared to 0 are as follows:

- When initialization is performed
- When NMI is input in burst mode
- When 0 is written to the DTME bit

The condition for DTME being set to 1 is as follows:

- When 1 is written to DTME after DTME is read as 0

Bit 7—Data Transfer Master Enable 1 (DTME1): Enables or disables data transfer on channel 1.

Bit 7

DTME1	Description	
0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt	(Initial value)
1	Data transfer enabled	

Bit 5—Data Transfer Master Enable 0 (DTME0): Enables or disables data transfer on channel 0.

Bit 5

DTME0	Description	
0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt	(Initial value)
1	Data transfer enabled	

Bits 6 and 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reason

When DTE = 1 and DTME = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 6—Data Transfer Enable 1 (DTE1): Enables or disables data transfer on channel 1.

Bit 6

DTE1	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4—Data Transfer Enable 0 (DTE0): Enables or disables data transfer on channel 0.

Bit 4

DTE0	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 and 1—Data Transfer Interrupt Enable B (DTIEB): These bits enable or disable an interrupt to the CPU or DTC when transfer is interrupted. If the DTIEB bit is set to 1 when DTME = 0, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC.

A transfer break interrupt can be canceled either by clearing the DTIEB bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1 transfer break interrupt.

Bit 3

DTIE1B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0 transfer break interrupt.

Bit 1

DTIE0B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bits 2 and 0—Data Transfer End Interrupt Enable A (DTIEA): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If DTIEA bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIEA bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Bit 2—Data Transfer End Interrupt Enable 1A (DTIE1A): Enables or disables the channel 1 transfer end interrupt.

Bit 2

DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer End Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0 transfer end interrupt.

Bit 0

DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

8.4 Register Descriptions (3)

8.4.1 DMA Write Enable Register (DMAWER)

The DMAC can activate the DTC with a transfer end interrupt, rewrite the channel on which the transfer ended using a DTC chain transfer, and reactivate the DTC. DMAWER applies restrictions so that only specific bits of DMACR for the specific channel and also DMATCR and DMABCR can be changed to prevent inadvertent changes being made to registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Figure 8-2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt, and reactivating channel 0A. The address register and count register area is re-set by the first DTC transfer, then the control register area is re-set by the second DTC chain transfer.

When re-setting the control register area, perform masking by setting bits in DMAWER to prevent modification of the contents of the other channels.

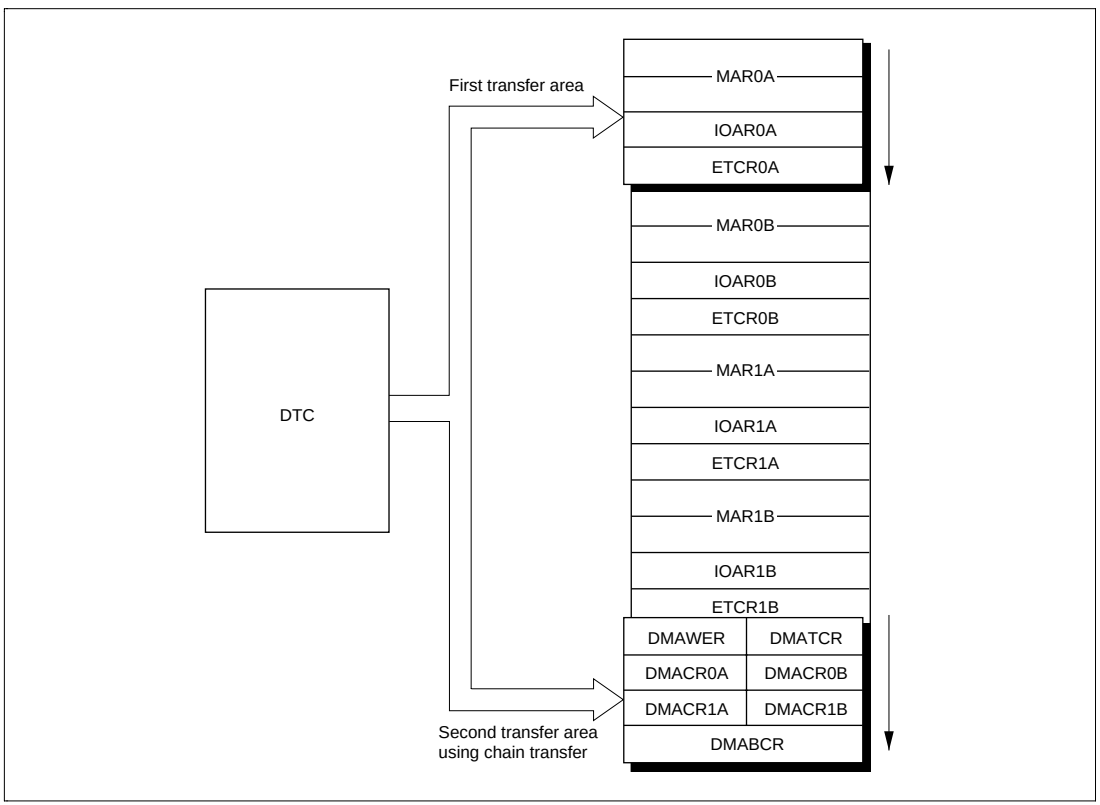


Figure 8-2 Areas for Register Re-Setting by DTC (Example: Channel 0A)

Bit	:	7	6	5	4	3	2	1	0
DMAWER	:	—	—	—	—	WE1B	WE1A	WE0B	WE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

DMAWER is an 8-bit readable/writable register that controls enabling or disabling of writes to the DMACR, DMABCR, and DMATCR by the DTC.

DMAWER is initialized to H'00 by a reset, and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Write Enable 1B (WE1B): Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR by the DTC.

Bit 3

WE1B	Description
0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are enabled

Bit 2—Write Enable 1A (WE1A): Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR by the DTC.

Bit 2

WE1A	Description
0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enabled

Bit 1—Write Enable 0B (WE0B): Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR.

Bit 1

WE0B	Description
0	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are enabled

Bit 0—Write Enable 0A (WE0A): Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR.

Bit 0

WE0A	Description
0	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are enabled

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless of the DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first reading 0. To reactivate a channel set to full address mode, write 1 to both Write Enable A and Write Enable B for the channel to be reactivated.

MAR, IOAR, and ETCR are always write-enabled regardless of the DMAWER settings. When modifying these registers, the channel for which the modification is to be made should be halted.

8.4.2 DMA Terminal Control Register (DMATCR)

Bit	:	7	6	5	4	3	2	1	0
DMATCR	:	—	—	TEE1	TEE0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	—	—	—	—

DMATCR is an 8-bit readable/writable register that controls enabling or disabling of DMAC transfer end pin output. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit.

DMATCR is initialized to H'00 by a reset, and in standby mode.

Bits 7 and 6—Reserved: These bits are always read as 0 and cannot be modified.

Bit 5—Transfer End Enable 1 (TEE1): Enables or disables transfer end pin 1 ($\overline{\text{TEND1}}$) output.

Bit 5

TEE1	Description
0	$\overline{\text{TEND1}}$ pin output disabled (Initial value)
1	$\overline{\text{TEND1}}$ pin output enabled

Bit 4—Transfer End Enable 0 (TEE0): Enables or disables transfer end pin 0 ($\overline{\text{TEND0}}$) output.

Bit 4		
TEE0	**Description**	
0	$\overline{\text{TEND0}}$ pin output disabled	(Initial value)
1	$\overline{\text{TEND0}}$ pin output enabled	

The $\overline{\text{TEND}}$ pins are assigned only to channel B in short address mode.

The transfer end signal indicates the transfer cycle in which the transfer counter reached 0, regardless of the transfer source. An exception is block transfer mode, in which the transfer end signal indicates the transfer cycle in which the block counter reached 0.

Bits 3 to 0—Reserved: These bits are always read as 0 and cannot be modified.

8.4.3 Module Stop Control Register (MSTPCR)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRA is a 8-bit readable/writable register that performs module stop mode control.

When the MSTPA7 bit in MSTPCR is set to 1, the DMAC operation stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 7—Module Stop (MSTP7): Specifies the DMAC module stop mode.

Bits 7		
MSTPA7	**Description**	
0	DMAC module stop mode cleared	(Initial value)
1	DMAC module stop mode set	

8.5 Operation

8.5.1 Transfer Modes

Table 8-5 lists the DMAC modes.

Table 8-5 DMAC Transfer Modes

Transfer Mode			Transfer Source	Remarks
Short address mode	Dual address mode	(1) Sequential mode (2) Idle mode (3) Repeat mode	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	• Up to 4 channels can operate independently • External request applies to channel B only • Single address mode applies to channel B only • Modes (1), (2), and (3) can also be specified for single address mode
	(4) Single address mode			
Full address mode		(5) Normal mode	• External request • Auto-request	• Max. 2-channel operation, combining channels A and B • With auto-request, burst mode transfer or cycle steal transfer can be selected
		(6) Block transfer mode	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI reception complete interrupt • A/D converter conversion end interrupt • External request	

Operation in each mode is summarized below.

(1) Sequential mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

(2) Idle mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer source address and transfer destination address are fixed. The transfer direction is programmable.

(3) Repeat mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. When the specified number of transfers have been completed, the addresses and transfer counter are restored to their original settings, and operation is continued. No interrupt request is sent to the CPU or DTC. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

(4) Single address mode

In response to a single transfer request, the specified number of transfers are carried out between external memory and an external device, one byte or one word at a time. Unlike dual address mode, source and destination accesses are performed in parallel. Therefore, either the source or the destination is an external device which can be accessed with a strobe alone, using the $\overline{\text{DACK}}$ pin. One address is specified as 24 bits, and for the other, the pin is set automatically. The transfer direction is programmable.

Modes (1), (2) and (3) can also be specified for single address mode.

(5) Normal mode

- Auto-request

  By means of register settings only, the DMAC is activated, and transfer continues until the specified number of transfers have been completed. An interrupt request can be sent to the CPU or DTC when transfer is completed. Both addresses are specified as 24 bits.

  — Cycle steal mode: The bus is released to another bus master every byte or word transfer.

  — Burst mode: The bus is held and transfer continued until the specified number of transfers have been completed.

- External request

  In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. Both addresses are specified as 24 bits.

(6) Block transfer mode

In response to a single transfer request, a block transfer of the specified block size is carried out. This is repeated the specified number of times, once each time there is a transfer request. At the end of each single block transfer, one address is restored to its original setting. An interrupt request can be sent to the CPU or DTC when the specified number of block transfers have been completed. Both addresses are specified as 24 bits.

8.5.2 Sequential Mode

Sequential mode can be specified by clearing the RPE bit in DMACR to 0. In sequential mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 8-6 summarizes register functions in sequential mode.

Table 8-6 Register Functions in Sequential Mode

Register	Function: DTDIR = 0	Function: DTDIR = 1	Initial Setting	Operation
23 MAR 0	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer
23 H'FF 15 IOAR 0	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
15 ETCR 0	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 8-3 illustrates operation in sequential mode.

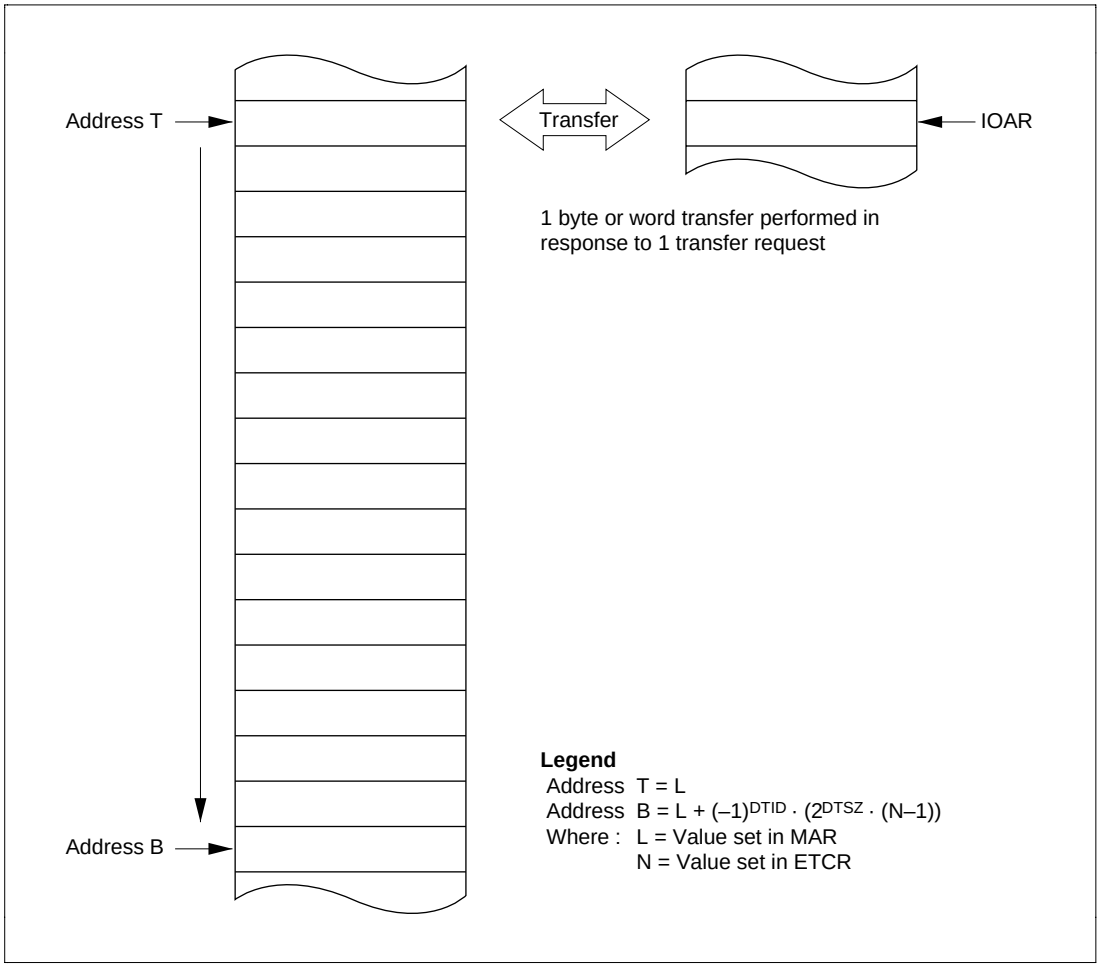


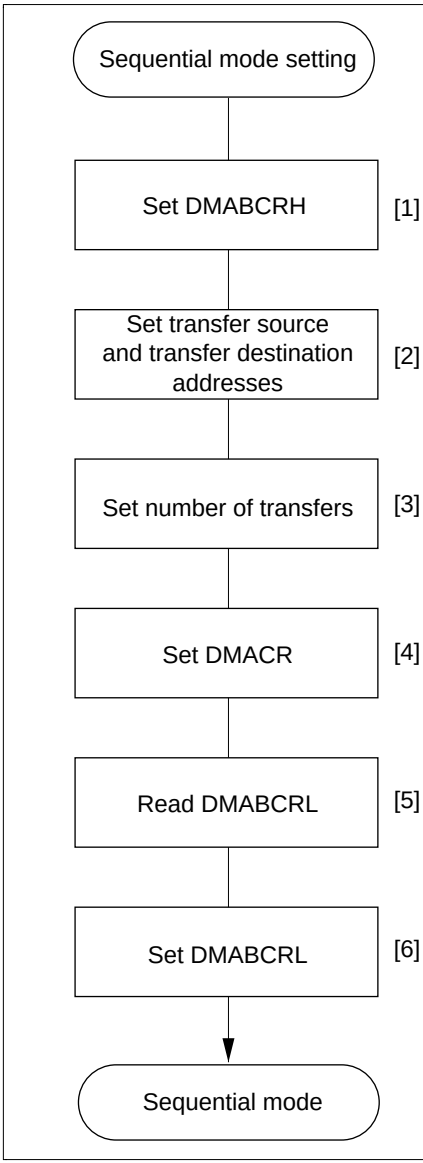
Figure 8-3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can be set for channel B only.

Figure 8-4 shows an example of the setting procedure for sequential mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 8-4 Example of Sequential Mode Setting Procedure

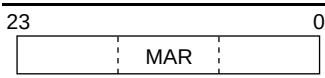
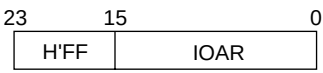
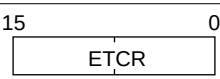
8.5.3 Idle Mode

Idle mode can be specified by setting the RPE bit and DTIE bit in DMACR to 1. In idle mode, one byte or word is transferred in response to a single transfer request, and this is executed the number of times specified in ETCR.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 8-7 summarizes register functions in idle mode.

Table 8-7 Register Functions in Idle Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is neither incremented nor decremented each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 8-5 illustrates operation in idle mode.

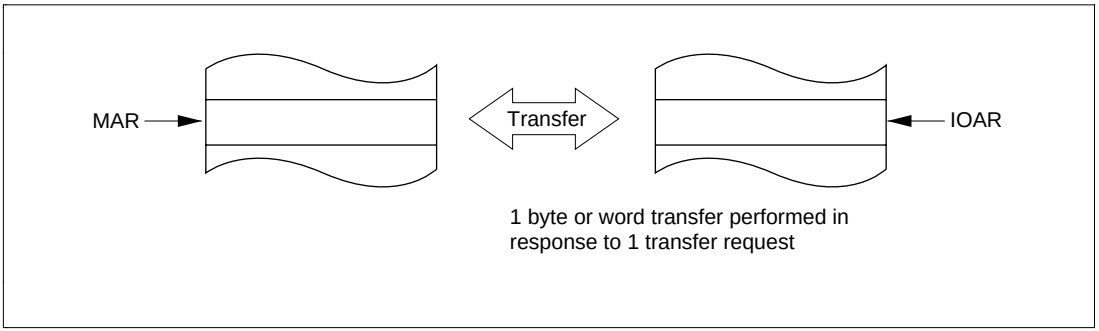


Figure 8-5 Operation in Idle Mode

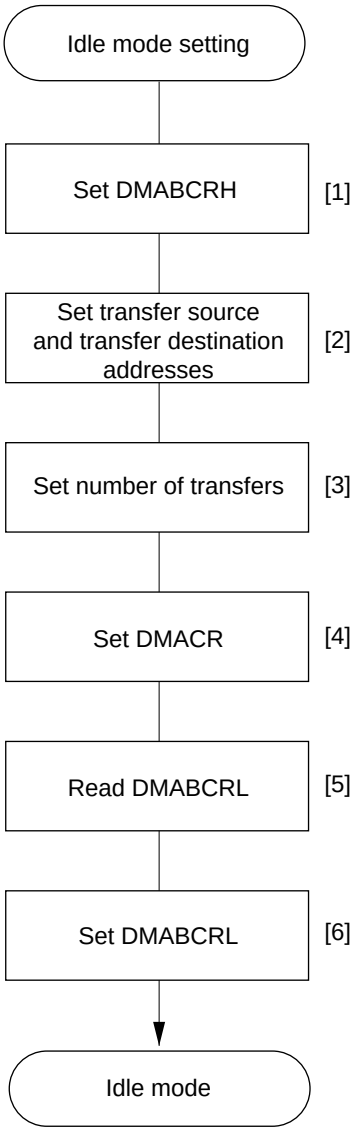
The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can be set for channel B only.

When the DMAC is used in single address mode, only channel B can be set.

Figure 8-6 shows an example of the setting procedure for idle mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

Figure 8-6 Example of Idle Mode Setting Procedure

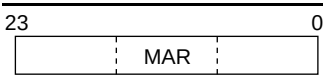
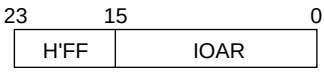
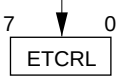
8.5.4 Repeat Mode

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit to 0. In repeat mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR. On completion of the specified number of transfers, MAR and ETCRL are automatically restored to their original settings and operation continues.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 8-8 summarizes register functions in repeat mode.

Table 8-8 Register Functions in Repeat Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer. Initial setting is restored when value reaches H'0000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers		Number of transfers	Fixed
	Transfer counter		Number of transfers	Decremented every transfer. Loaded with ETCRH value when count reaches H'00

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

The number of transfers is specified as 8 bits by ETCRH and ETCRL. The maximum number of transfers, when H'00 is set in both ETCRH and ETCRL, is 256.

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold the number of transfers. ETCRL is decremented by 1 each time a transfer is executed, and when its value reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in MAR is restored in accordance with the values of the DTSZ and DTID bits in DMACR. The MAR restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit is cleared. To end the transfer operation, therefore, you should clear the DTE bit to 0. A transfer end interrupt request is not sent to the CPU or DTC.

By setting the DTE bit to 1 again after it has been cleared, the operation can be restarted from the transfer after that terminated when the DTE bit was cleared.

Figure 8-7 illustrates operation in repeat mode.

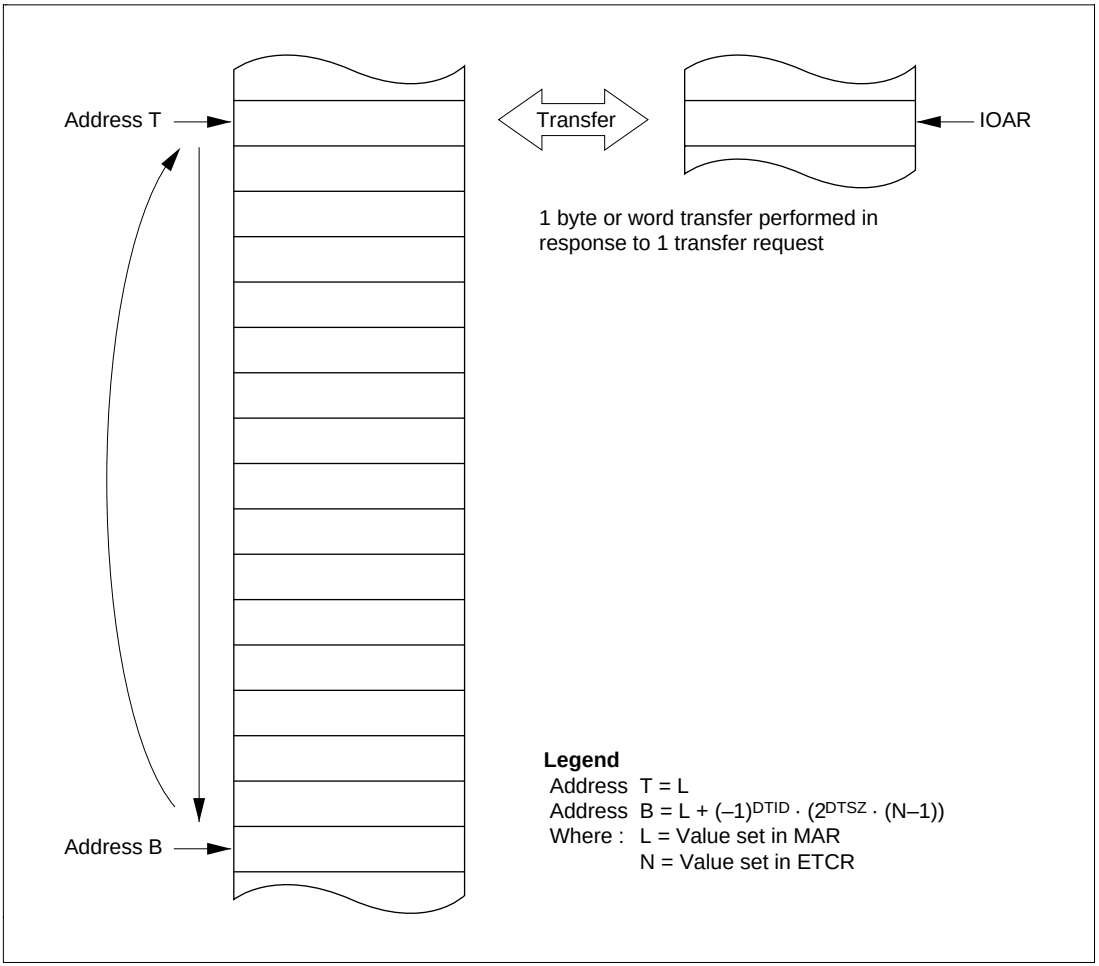
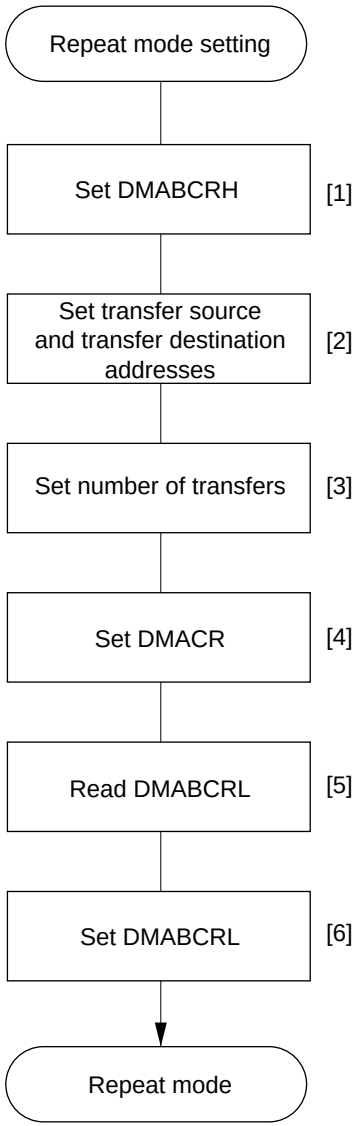


Figure 8-7 Operation in Repeat mode

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can be set for channel B only.

Figure 8-8 shows an example of the setting procedure for repeat mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCRH and ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer.

Figure 8-8 Example of Repeat Mode Setting Procedure

8.5.5 Single Address Mode

Single address mode can only be specified for channel B. This mode can be specified by setting the SAE bit in DMABCR to 1 in short address mode.

One address is specified by MAR, and the other is set automatically to the data transfer acknowledge pin ($\overline{\text{DACK}}$). The transfer direction can be specified by the DTDIR in DMACR.

Table 8-9 summarizes register functions in single address mode.

Table 8-9 Register Functions in Single Address Mode

Register	Function: DTDIR = 0	Function: DTDIR = 1	Initial Setting	Operation
23 … 0 MAR	Source address register	Destination address register	Start address of transfer destination or transfer source	*
$\overline{\text{DACK}}$ pin	Write strobe	Read strobe	(Set automatically by SAE bit; IOAR is invalid)	Strobe for external device
15 … 0 ETCR	Transfer counter		Number of transfers	*

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit
$\overline{\text{DACK}}$: Data transfer acknowledge

Note: * See the operation descriptions in sections 8.5.2, Sequential Mode, 8.5.3, Idle Mode, and 8.5.4, Repeat Mode.

MAR specifies the start address of the transfer source or transfer destination as 24 bits.

IOAR is invalid; in its place the strobe for external devices ($\overline{\text{DACK}}$) is output.

Figure 8-9 illustrates operation in single address mode (when sequential mode is specified).

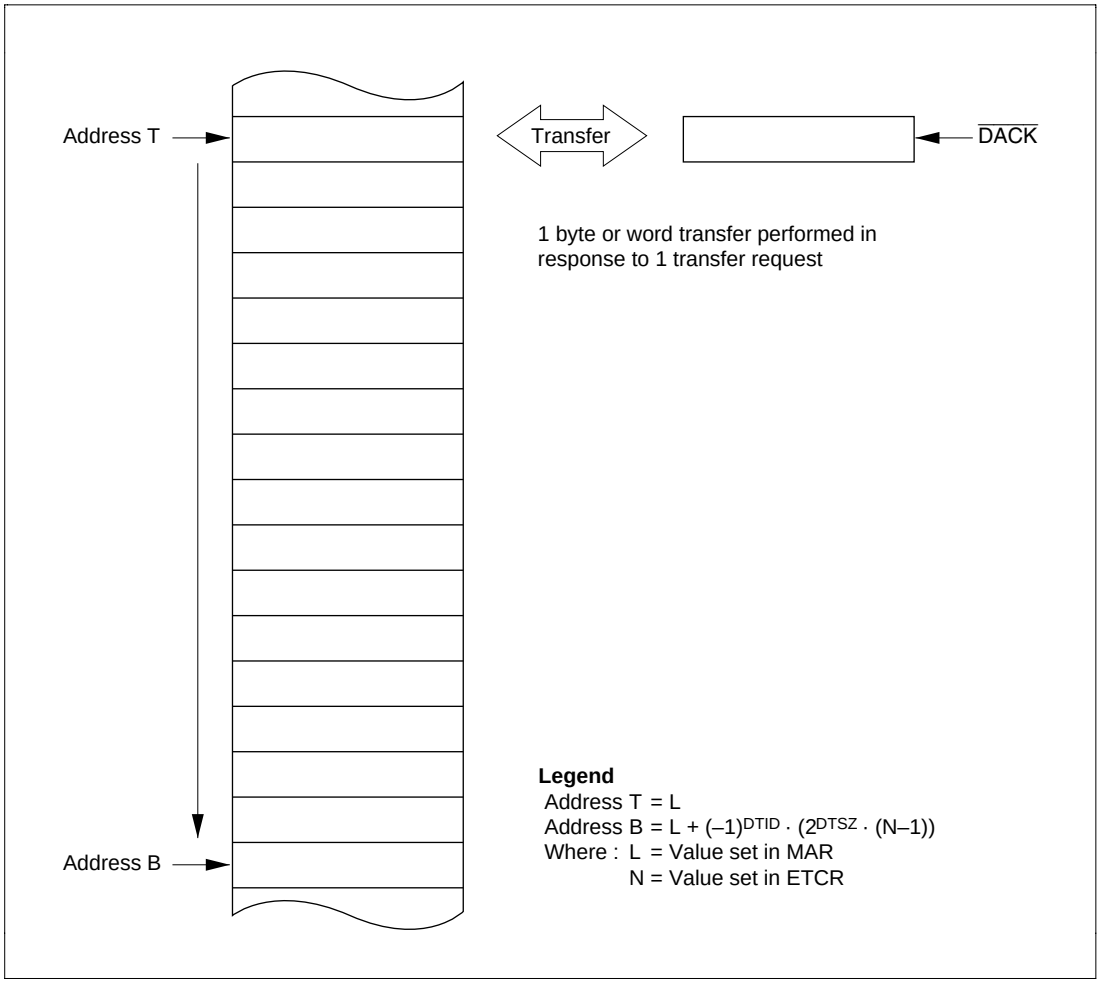
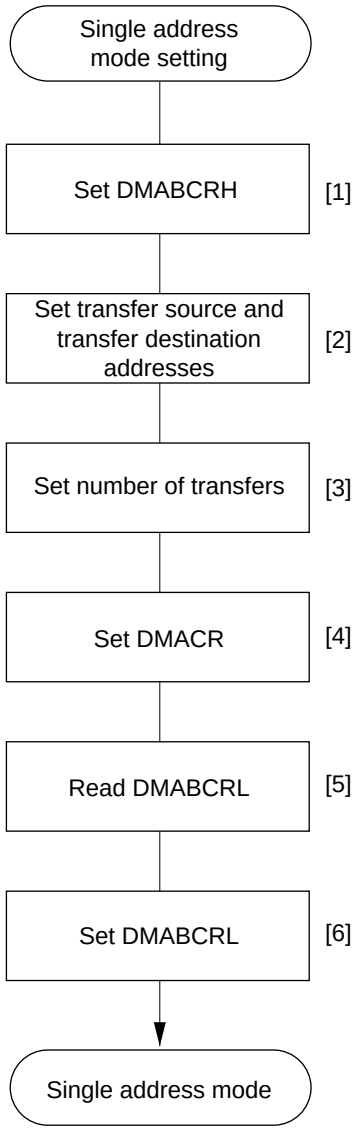


Figure 8-9 Operation in Single Address Mode (When Sequential Mode is Specified)

Figure 8-10 shows an example of the setting procedure for single address mode (when sequential mode is specified).



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Set the SAE bit to 1 to select single address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address/transfer destination address in MAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 8-10 Example of Single Address Mode Setting Procedure (When Sequential Mode is Specified)

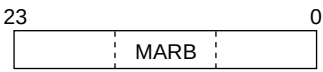
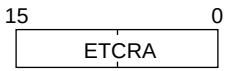
8.5.6 Normal Mode

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCR to 1 and clearing the BLKE bit in DMACRA to 0.

In normal mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB.

Table 8-10 summarizes register functions in normal mode.

Table 8-10 Register Functions in Normal Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Transfer counter	Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend

MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed.

Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented each time a transfer is performed, and when its value reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

Figure 8-11 illustrates operation in normal mode.

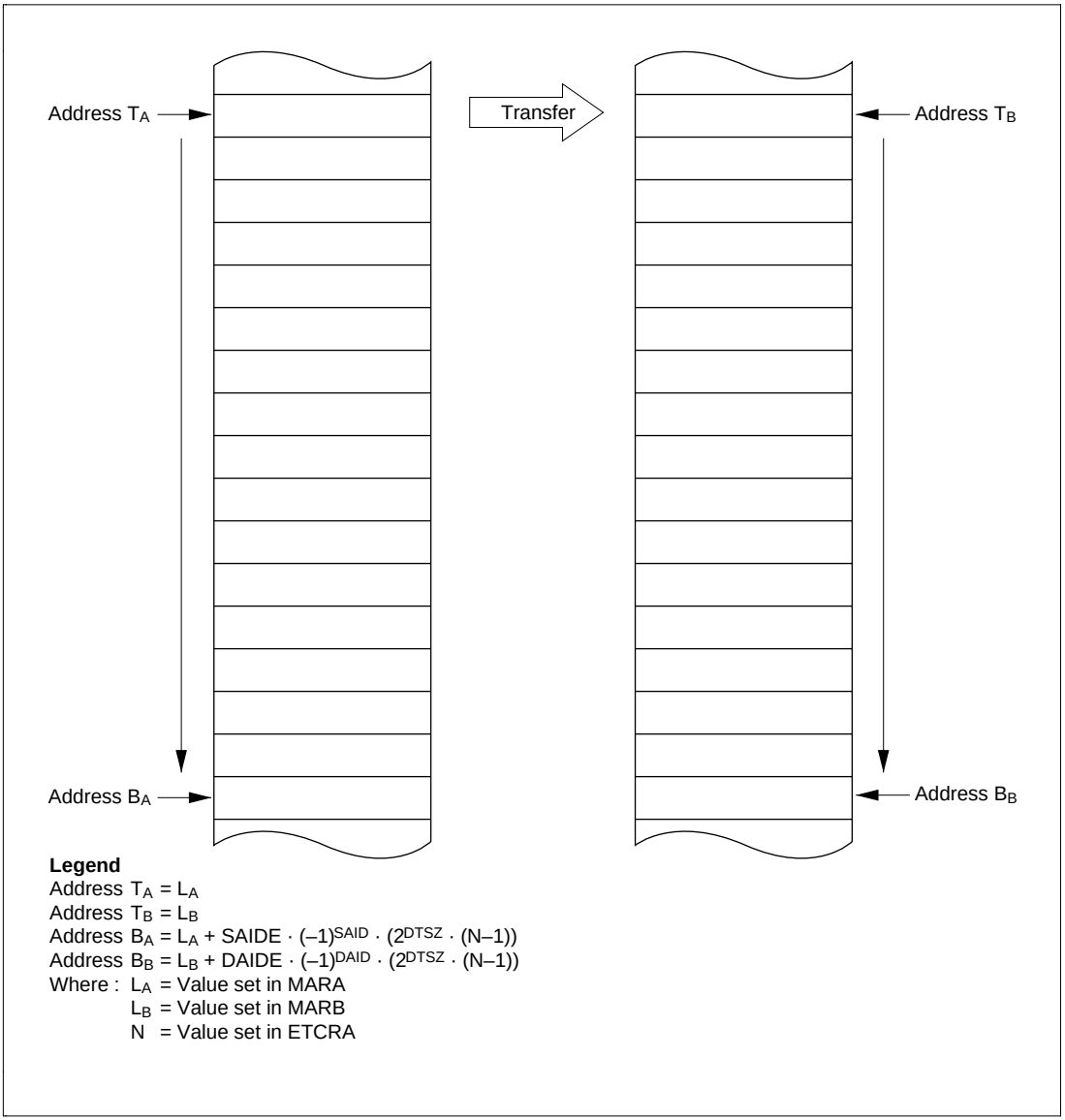


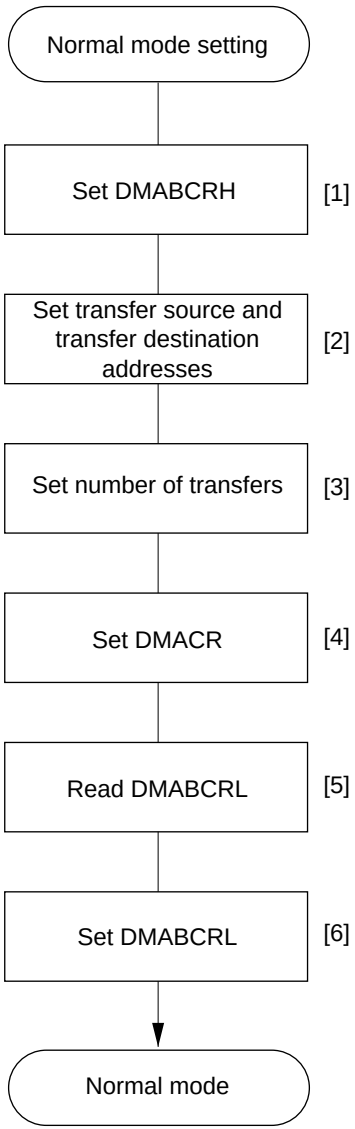
Figure 8-11 Operation in Normal Mode

Transfer requests (activation sources) are external requests and auto-requests.

With auto-request, the DMAC is only activated by register setting, and the specified number of transfers are performed automatically. With auto-request, cycle steal mode or burst mode can be selected. In cycle steal mode, the bus is released to another bus master each time a transfer is performed. In burst mode, the bus is held continuously until transfer ends.

For setting details, see section 8.3.4, DMA Controller Register (DMACR).

Figure 8-12 shows an example of the setting procedure for normal mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Clear the BLKE bit to 0 to select normal mode.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 8-12 Example of Normal Mode Setting Procedure

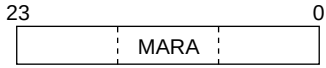
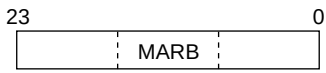
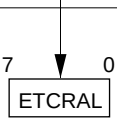
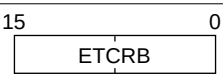
8.5.7 Block Transfer Mode

In block transfer mode, transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCR and the BLKE bit in DMACRA to 1.

In block transfer mode, a transfer of the specified block size is carried out in response to a single transfer request, and this is executed the specified number of times. The transfer source is specified by MARA, and the transfer destination by MARB. Either the transfer source or the transfer destination can be selected as a block area (an area composed of a number of bytes or words).

Table 8-11 summarizes register functions in block transfer mode.

Table 8-11 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Holds block size	Block size	Fixed
	Block size counter	Block size	Decremented every transfer; ETCRH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decremented every block transfer; transfer ends when count reaches H'0000

Legend

MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A
ETCRB : Transfer count register B

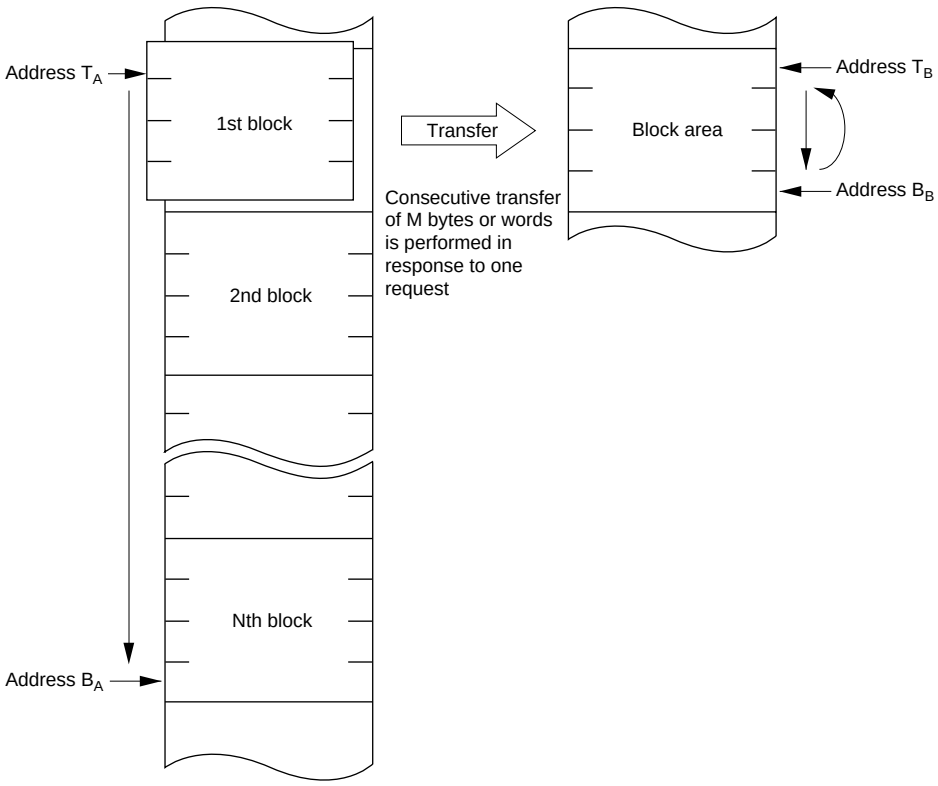
MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed.

Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

Whether a block is to be designated for MARA or for MARB is specified by the BLKDIR bit in DMACRA.

To specify the number of transfers, if M is the size of one block (where M = 1 to 256) and N transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and ETCRAL, and N in ETCRB.

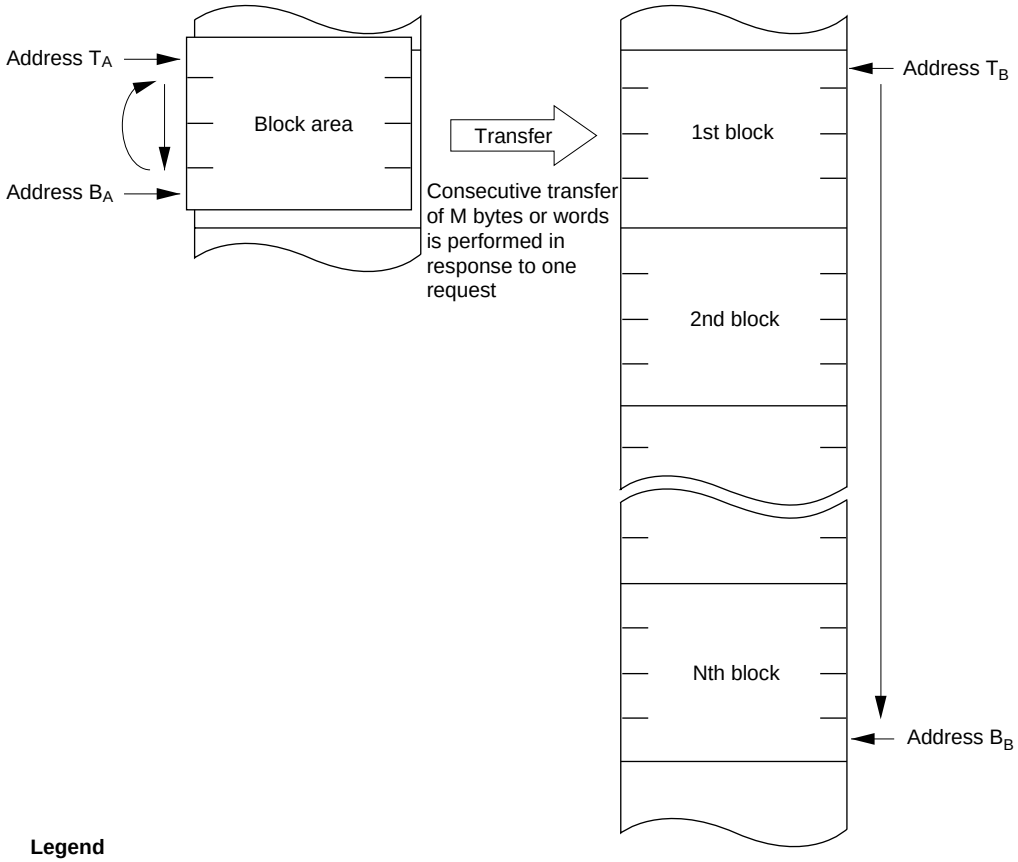
Figure 8-13 illustrates operation in block transfer mode when MARB is designated as a block area.



Legend
Address T_A = L_A
Address T_B = L_B
Address B_A = L_A + SAIDE · $(-1)^{SAID}$ · $(2^{DTSZ}$ · (M·N–1))
Address B_B = L_B + DAIDE · $(-1)^{DAID}$ · $(2^{DTSZ}$ · (N–1))
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 8-13 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 8-14 illustrates operation in block transfer mode when MARA is designated as a block area.



Legend
Address $T_A = L_A$
Address $T_B = L_B$
Address $B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (N-1))$
Address $B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (M \cdot N-1))$
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 8-14 Operation in Block Transfer Mode (BLKDIR = 1)

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In response to a single transfer request, burst transfer is performed until the value in ETCRAL reaches H'00. ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MAR register for which a block designation has been given by the BLKDIR bit in DMACRA is restored in accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

ETCRB is decremented by 1 every block transfer, and when the count reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this point, an interrupt request is sent to the CPU or DTC.

Figure 8-15 shows the operation flow in block transfer mode.

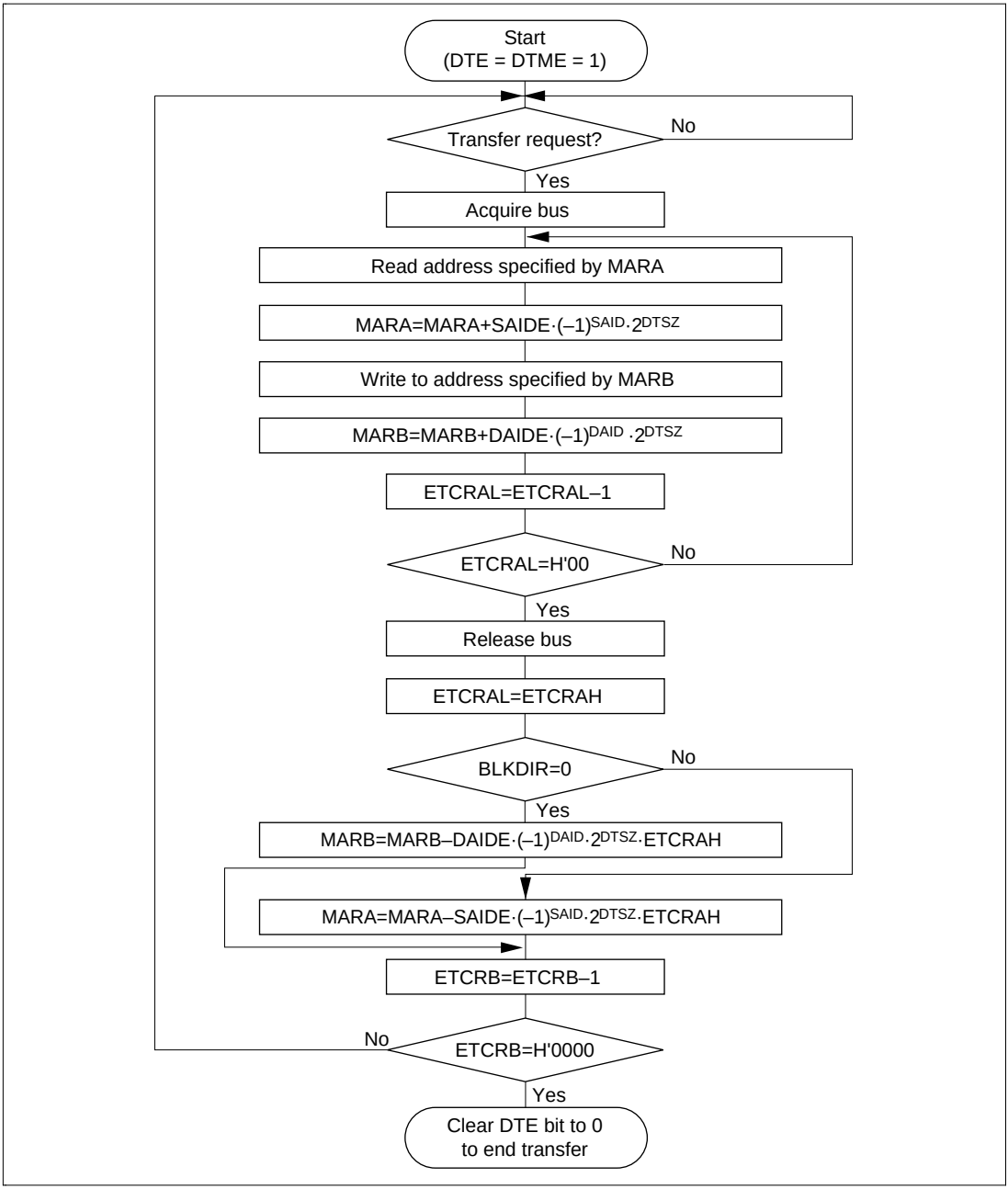
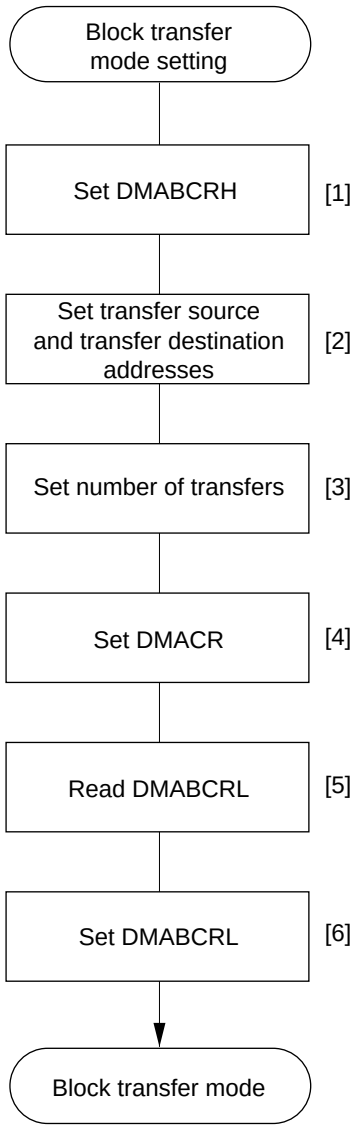


Figure 8-15 Operation Flow in Block Transfer Mode

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts.

For details, see section 8.3.4, DMA Control Register (DMACR).

Figure 8-16 shows an example of the setting procedure for block transfer mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the block size in both ETCRAH and ETCRAL. Set the number of transfers in ETCRB.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Set the BLKE bit to 1 to select block transfer mode.
- Specify whether the transfer source or the transfer destination is a block area with the BLKDIR bit.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts to the CPU with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 8-16 Example of Block Transfer Mode Setting Procedure

8.5.8 DMAC Activation Sources

DMAC activation sources consist of internal interrupts, external requests, and auto-requests. The activation sources that can be specified depend on the transfer mode and the channel, as shown in table 8-12.

Table 8-12 DMAC Activation Sources

		Short Address Mode		Full Address Mode	
Activation Source		**Channels 0A and 1A**	**Channels 0B and 1B**	**Normal Mode**	**Block Transfer Mode**
Internal Interrupts	ADI	○	○	X	○
	TXI0	○	○	X	○
	RXI0	○	○	X	○
	TXI1	○	○	X	○
	RXI1	○	○	X	○
	TGI0A	○	○	X	○
	TGI1A	○	○	X	○
	TGI2A	○	○	X	○
	TGI3A	○	○	X	○
	TGI4A	○	○	X	○
	TGI5A	○	○	X	○
External Requests	$\overline{\text{DREQ}}$ pin falling edge input	X	○	○	○
	$\overline{\text{DREQ}}$ pin low-level input	X	○	○	○
Auto-request		X	X	○	X

Legend

○: Can be specified

X : Cannot be specified

Activation by Internal Interrupt: An interrupt request selected as a DMAC activation source can be sent simultaneously to the CPU and DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt, the DMAC accepts the request independently of the interrupt controller. Consequently, interrupt controller priority settings are not accepted.

If the DMAC is activated by a CPU interrupt source or an interrupt source that is not used as a DTC activation source (DTA = 1), the interrupt source flag is cleared automatically by the DMA transfer. With ADI, TXI, and RXI interrupts, however, the interrupt source flag is not cleared unless the prescribed register is accessed in a DMA transfer. If the same interrupt is used as an

activation source for more than one channel, the interrupt request flag is cleared when the highest-priority channel is activated first. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When DTE = 0, such as after completion of a transfer, a request from the selected activation source is not sent to the DMAC, regardless of the DTA bit. In this case, the relevant interrupt request is sent to the CPU or DTC.

In case of overlap with a CPU interrupt source or DTC activation source (DTA = 0), the interrupt request flag is not cleared by the DMAC.

Activation by External Request: If an external request ($\overline{\text{DREQ}}$ pin) is specified as an activation source, the relevant port should be set to input mode in advance.

Level sensing or edge sensing can be used for external requests.

External request operation in normal mode (short address mode or full address mode) is described below.

When edge sensing is selected, a 1-byte or 1-word transfer is executed each time a high-to-low transition is detected on the $\overline{\text{DREQ}}$ pin. The next transfer may not be performed if the next edge is input before transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{DREQ}}$ pin is held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

Activation by Auto-Request: Auto-request activation is performed by register setting only, and transfer continues to the end.

With auto-request activation, cycle steal mode or burst mode can be selected.

In cycle steal mode, the DMAC releases the bus to another bus master each time a byte or word is transferred. DMA and CPU cycles usually alternate.

In burst mode, the DMAC keeps possession of the bus until the end of the transfer, and transfer is performed continuously.

Single Address Mode: The DMAC can operate in dual address mode in which read cycles and write cycles are separate cycles, or single address mode in which read and write cycles are executed in parallel.

In dual address mode, transfer is performed with the source address and destination address specified separately.

In single address mode, on the other hand, transfer is performed between external space in which either the transfer source or the transfer destination is specified by an address, and an external device for which selection is performed by means of the $\overline{\text{DACK}}$ strobe, without regard to the address. Figure 8-17 shows the data bus in single address mode.

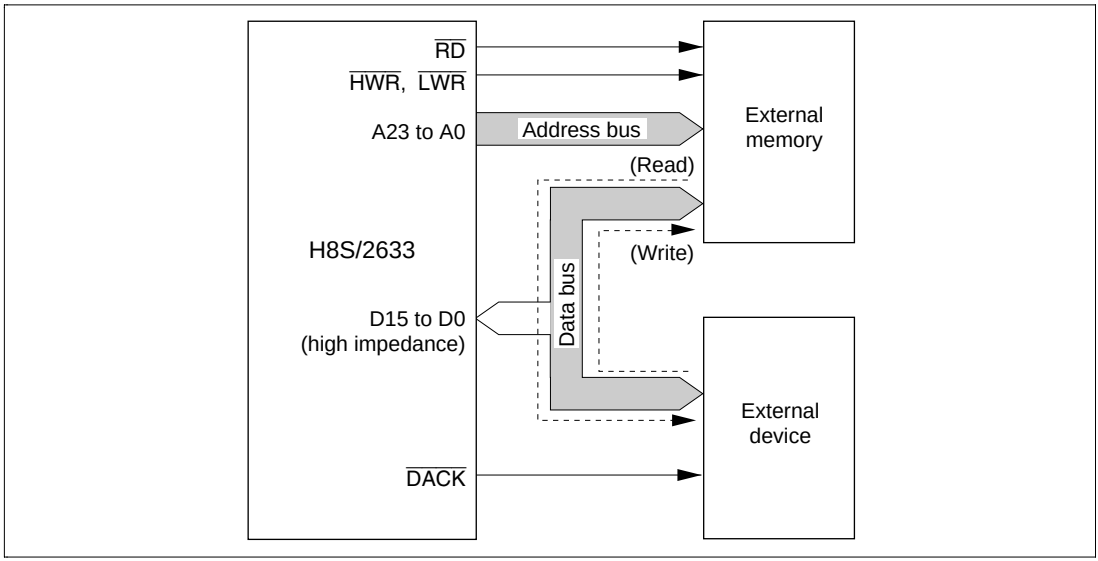


Figure 8-17 Data Bus in Single Address Mode

When using the DMAC for single address mode reading, transfer is performed from external memory to the external device, and the $\overline{\text{DACK}}$ pin functions as a write strobe for the external device. When using the DMAC for single address mode writing, transfer is performed from the external device to external memory, and the $\overline{\text{DACK}}$ pin functions as a read strobe for the external device. Since there is no directional control for the external device, one or other of the above single directions should be used.

Bus cycles in single address mode are in accordance with the settings of the bus controller for the external memory area. On the external device side, $\overline{\text{DACK}}$ is output in synchronization with the address strobe. For details of bus cycles, see section 8.5.11, DMAC Bus Cycles (Single Address Mode).

Do not specify internal space for transfer addresses in single address mode.

8.5.9 Basic DMAC Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 8-18. In this example, word-size transfer is performed from 16-bit , 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As with CPU cycles, DMA cycles conform to the bus controller settings.

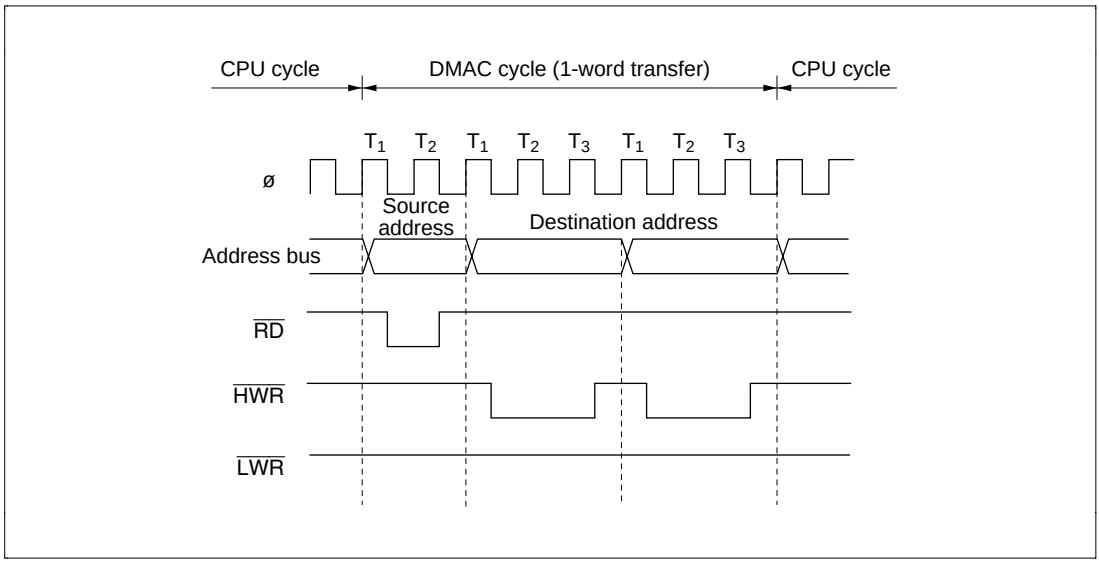


Figure 8-18 Example of DMA Transfer Bus Timing

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

8.5.10 DMAC Bus Cycles (Dual Address Mode)

Short Address Mode: Figure 8-19 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size short address mode transfer (sequential/idle/repeat mode) is performed from external 8-bit, 2-state access space to internal I/O space.

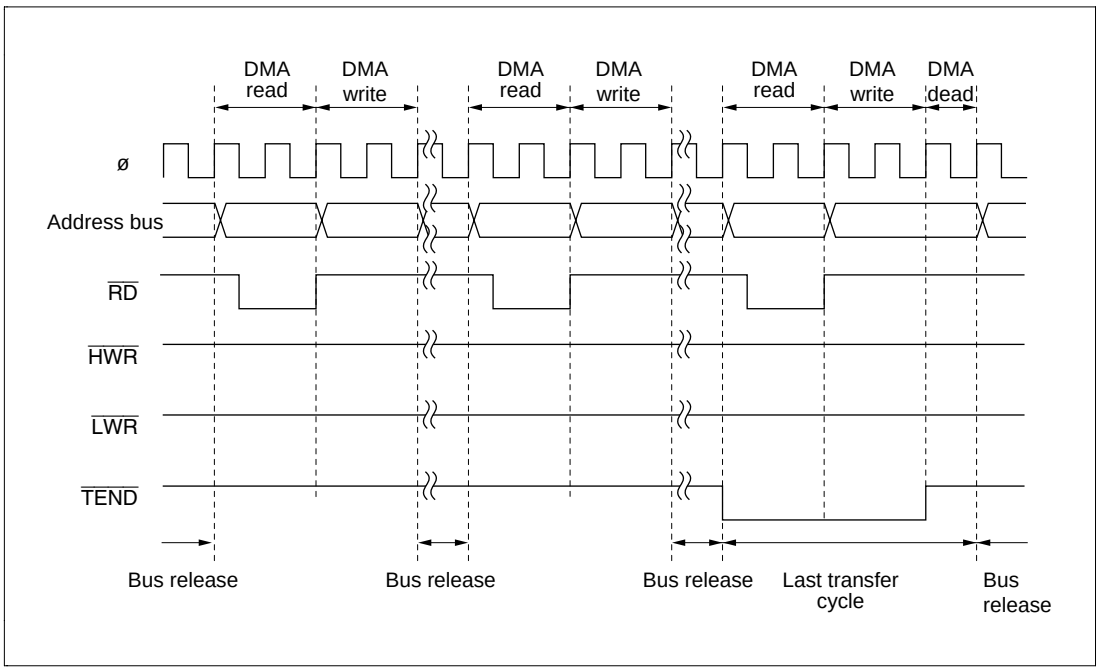


Figure 8-19 Example of Short Address Mode Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released one or more bus cycles are inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

In repeat mode, when $\overline{\text{TEND}}$ output is enabled, $\overline{\text{TEND}}$ output goes low in the transfer cycle in which the transfer counter reaches 0.

Full Address Mode (Cycle Steal Mode): Figure 8-20 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

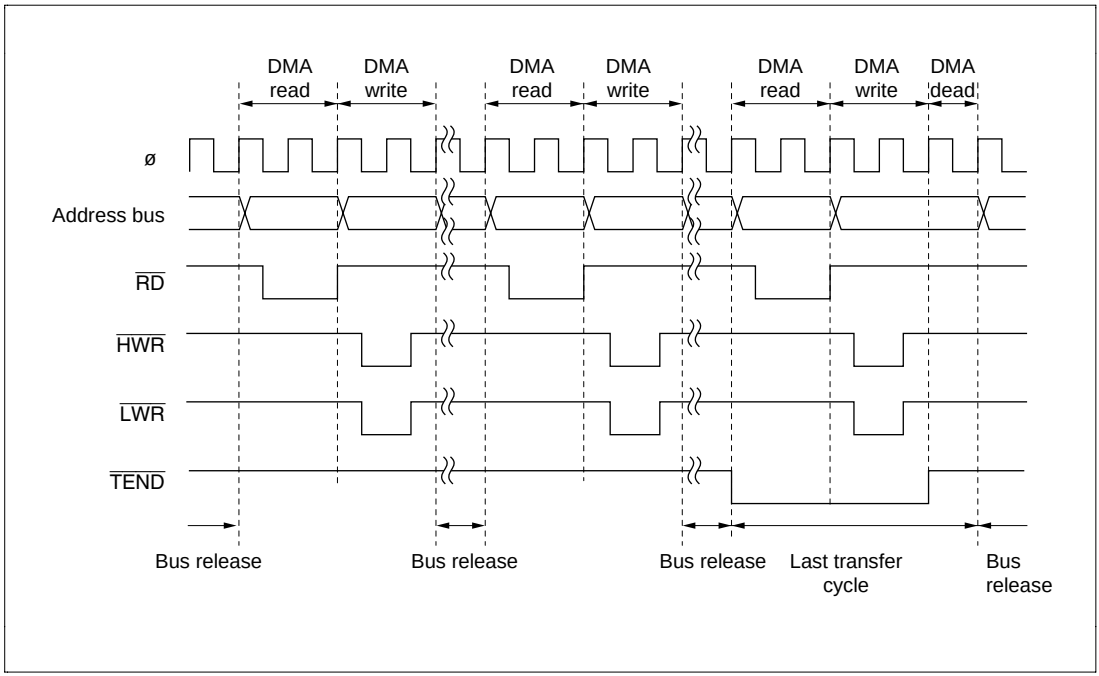


Figure 8-20 Example of Full Address Mode (Cycle Steal) Transfer

A one-byte or one-word transfer is performed, and after the transfer the bus is released. While the bus is released one bus cycle is inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Full Address Mode (Burst Mode): Figure 8-21 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

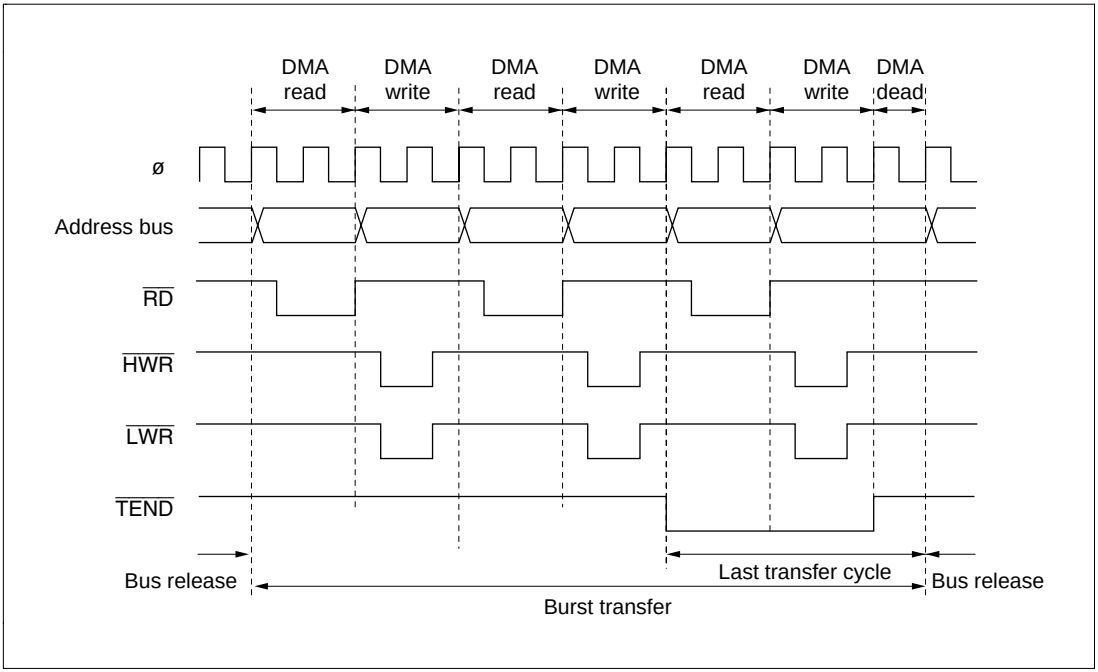


Figure 8-21 Example of Full Address Mode (Burst Mode) Transfer

In burst mode, one-byte or one-word transfers are executed consecutively until transfer ends.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, that channel has to wait until the burst transfer ends.

If an NMI is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit is cleared and the channel is placed in the transfer disabled state. If burst transfer has already been activated inside the DMAC, the bus is released on completion of a one-byte or one-word transfer within the burst transfer, and burst transfer is suspended. If the last transfer cycle of the burst transfer has already been activated inside the DMAC, execution continues to the end of the transfer even if the DTME bit is cleared.

Full Address Mode (Block Transfer Mode): Figure 8-22 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size full address mode transfer (block transfer mode) is performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.

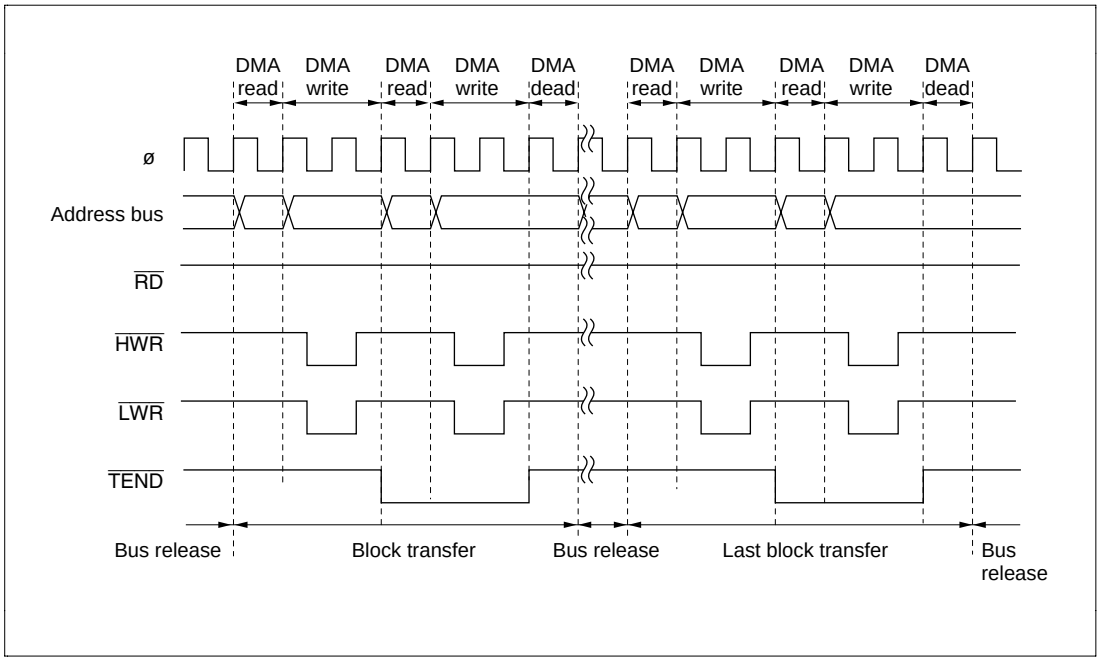


Figure 8-22 Example of Full Address Mode (Block Transfer Mode) Transfer

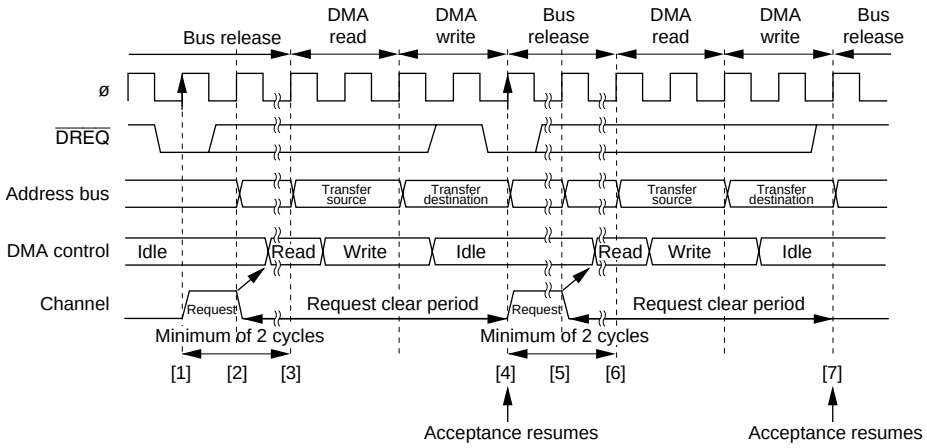
A one-block transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are inserted by the CPU or DTC.

In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

One block is transmitted without interruption. NMI generation does not affect block transfer operation.

$\overline{\text{DREQ}}$ Pin Falling Edge Activation Timing: Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 8-23 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

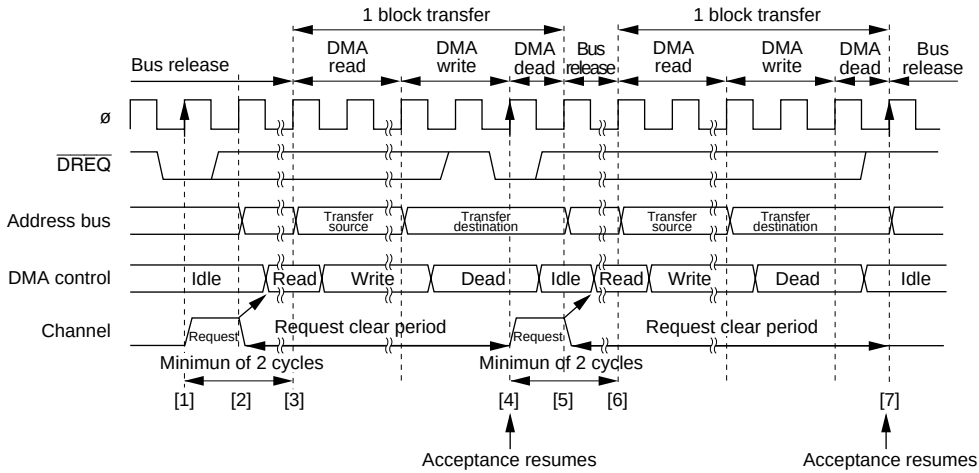
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 8-23 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA write cycle ends, acceptance resumes after the end of the write cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 8-24 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated block transfer mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

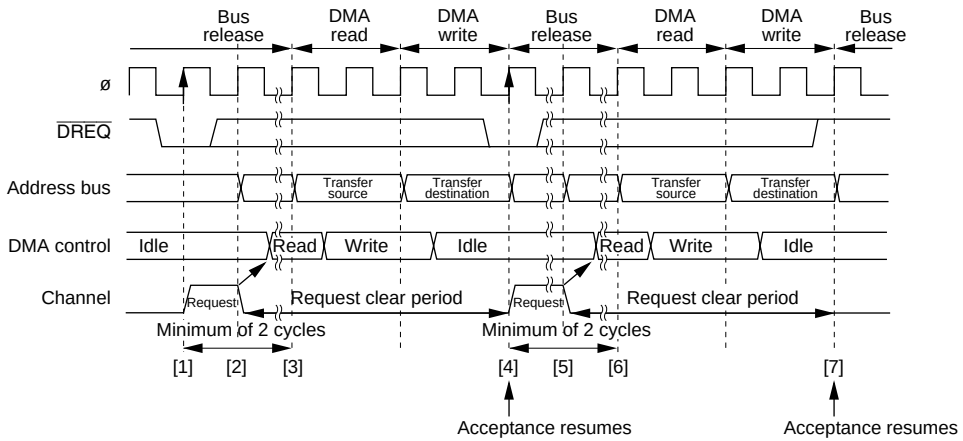
Figure 8-24 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA dead cycle ends, acceptance resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

$\overline{\text{DREQ}}$ Level Activation Timing (Normal Mode): Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 8-25 shows an example of $\overline{\text{DREQ}}$ level activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMA cycle is started.
[4] [7] Acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

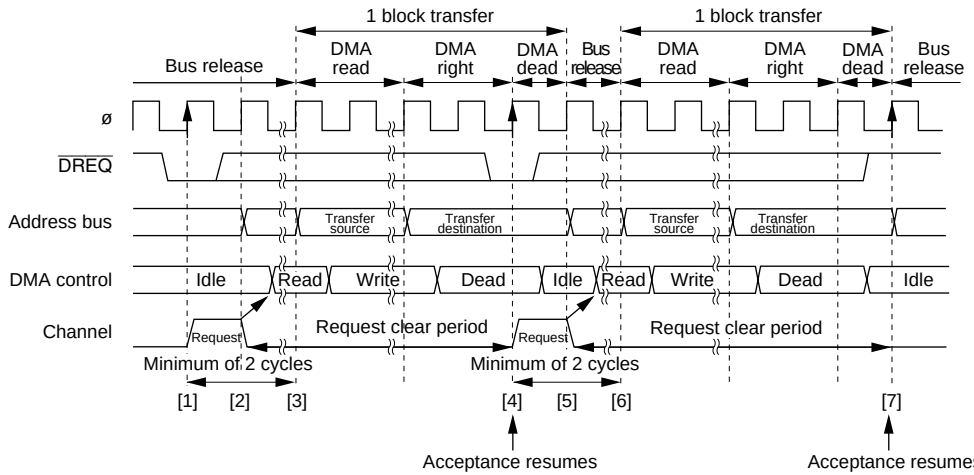
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 8-25 Example of $\overline{\text{DREQ}}$ Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the write cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 8-26 shows an example of $\overline{\text{DREQ}}$ level activated block transfer mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMA cycle is started.
[4] [7] Acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 8-26 Example of $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the dead cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

8.5.11 DMAC Bus Cycles (Single Address Mode)

Single Address Mode (Read): Figure 8-27 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

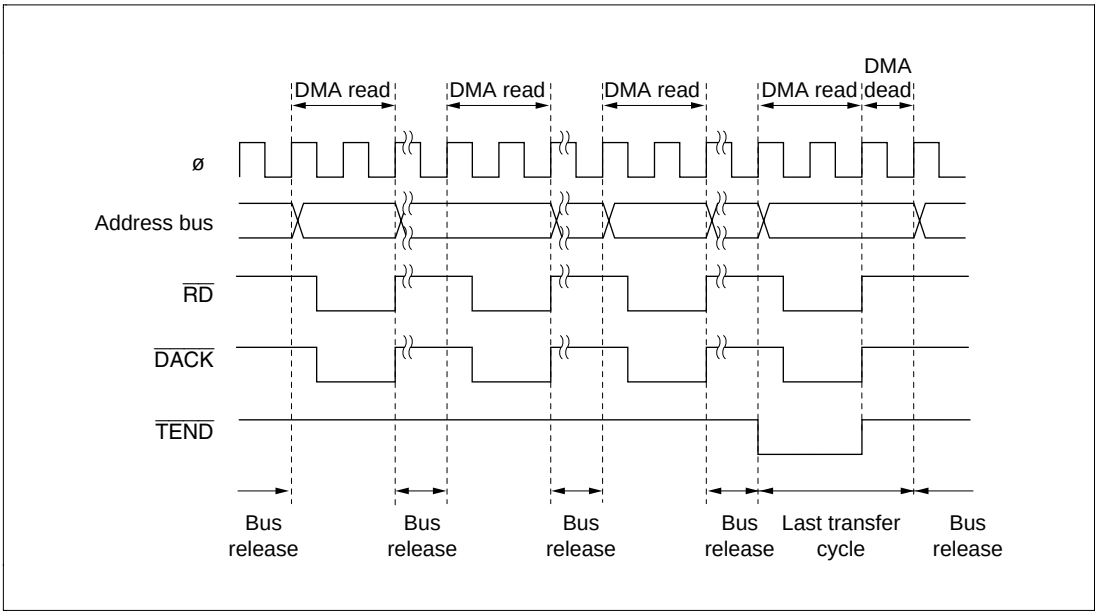


Figure 8-27 Example of Single Address Mode (Byte Read) Transfer

Figure 8-28 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

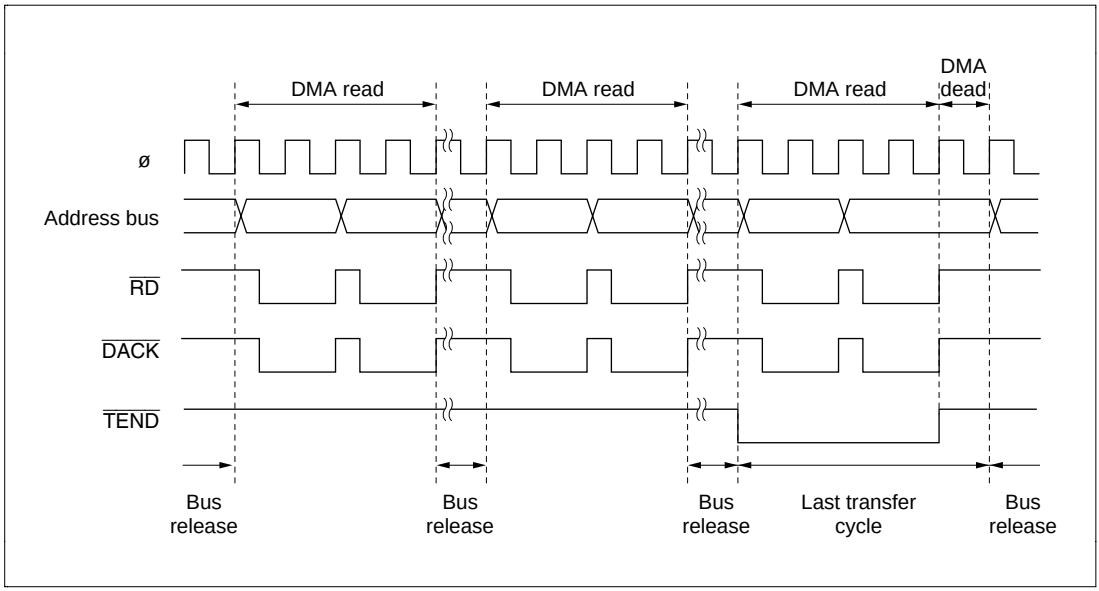


Figure 8-28 Example of Single Address Mode (Word Read) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Single Address Mode (Write): Figure 8-29 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and byte-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

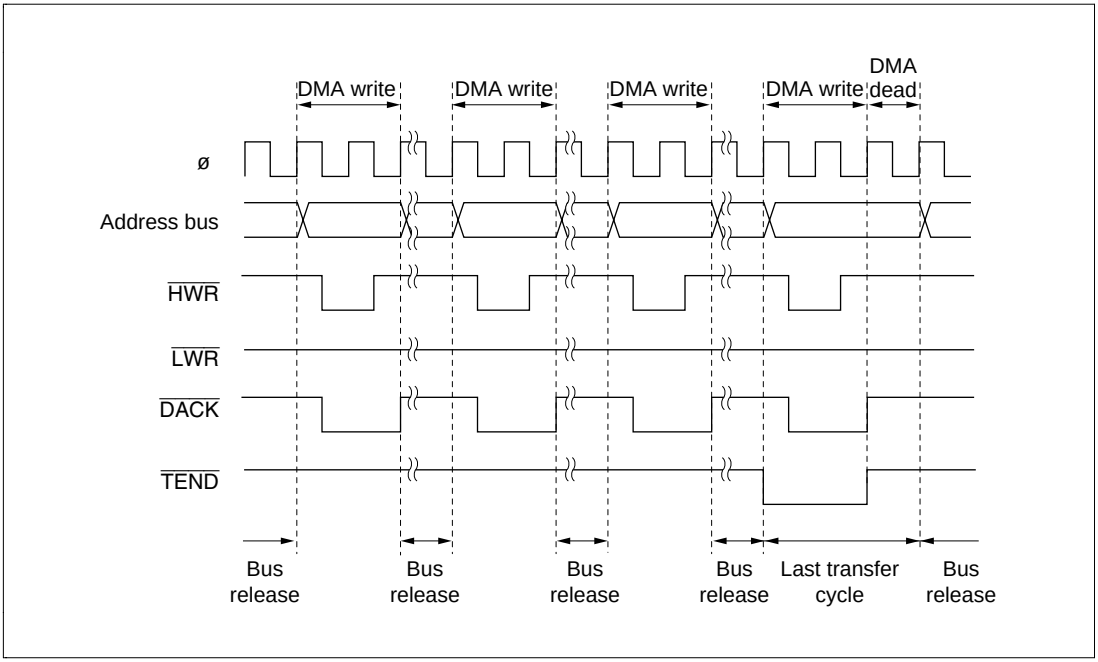


Figure 8-29 Example of Single Address Mode (Byte Write) Transfer

Figure 8-30 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.

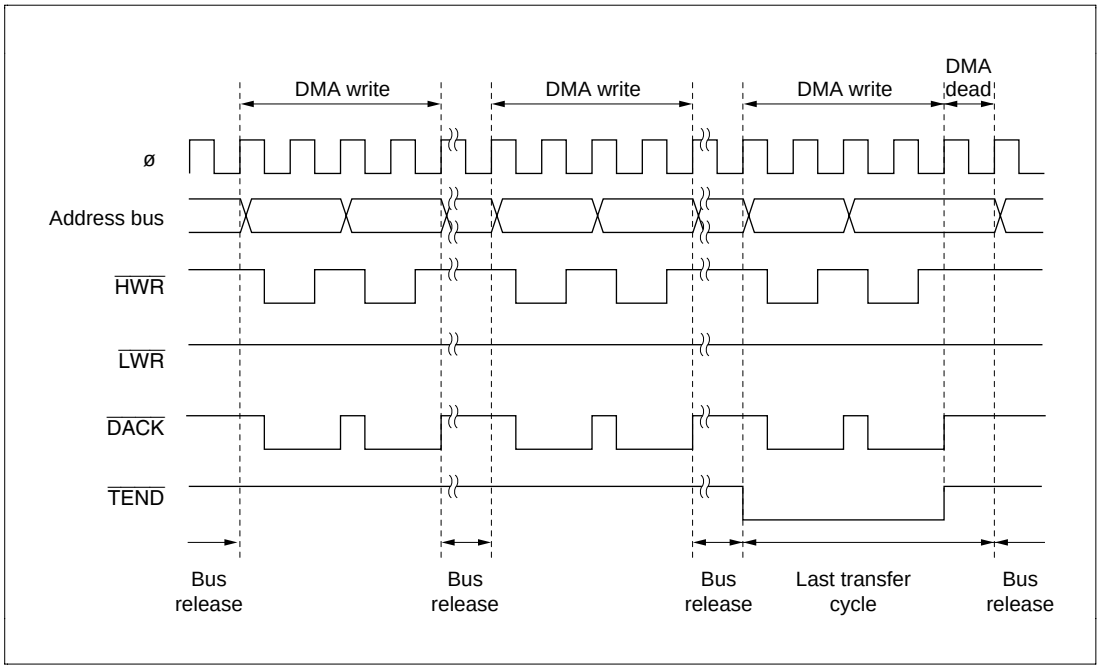


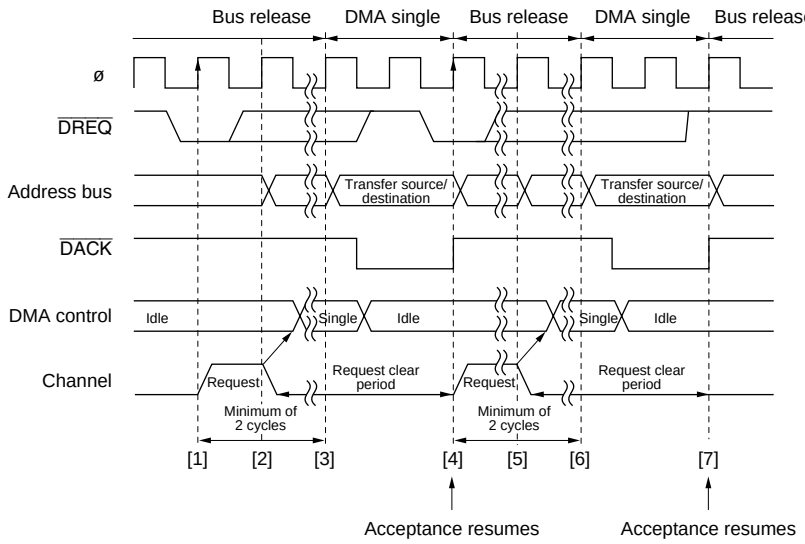
Figure 8-30 Example of Single Address Mode (Word Write) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released one or more bus cycles are inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

$\overline{\text{DREQ}}$ Pin Falling Edge Activation Timing: Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 8-31 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated single address mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of ø starts.
[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

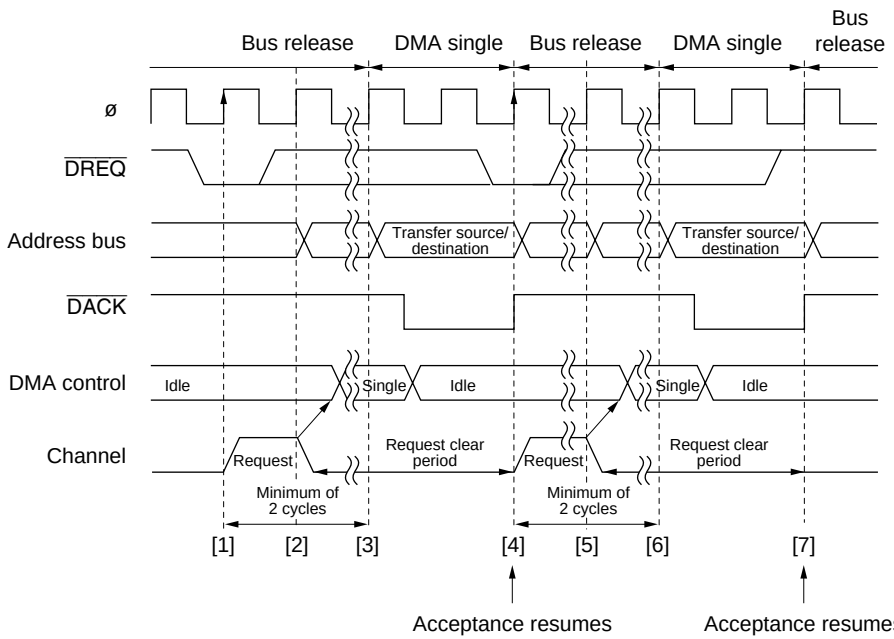
Figure 8-31 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Single Address Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA single cycle ends, acceptance resumes after the end of the single cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

$\overline{\text{DREQ}}$ Pin Low Level Activation Timing: Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 8-32 shows an example of $\overline{\text{DREQ}}$ pin low level activated single address mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMAC cycle is started.
[4] [7] Acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 8-32 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Single Address Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the single cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

8.5.12 Write Data Buffer Function

DMAC internal-to-external dual address transfers and single address transfers can be executed at high speed using the write data buffer function, enabling system throughput to be improved.

When the WDBE bit of BCRL in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel. Internal accesses are independent of the bus master, and DMAC dead cycles are regarded as internal accesses.

A low level can always be output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output is an external bus cycle. However, a low level is not output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle.

Figure 8-33 shows an example of burst mode transfer from on-chip RAM to external memory using the write data buffer function.

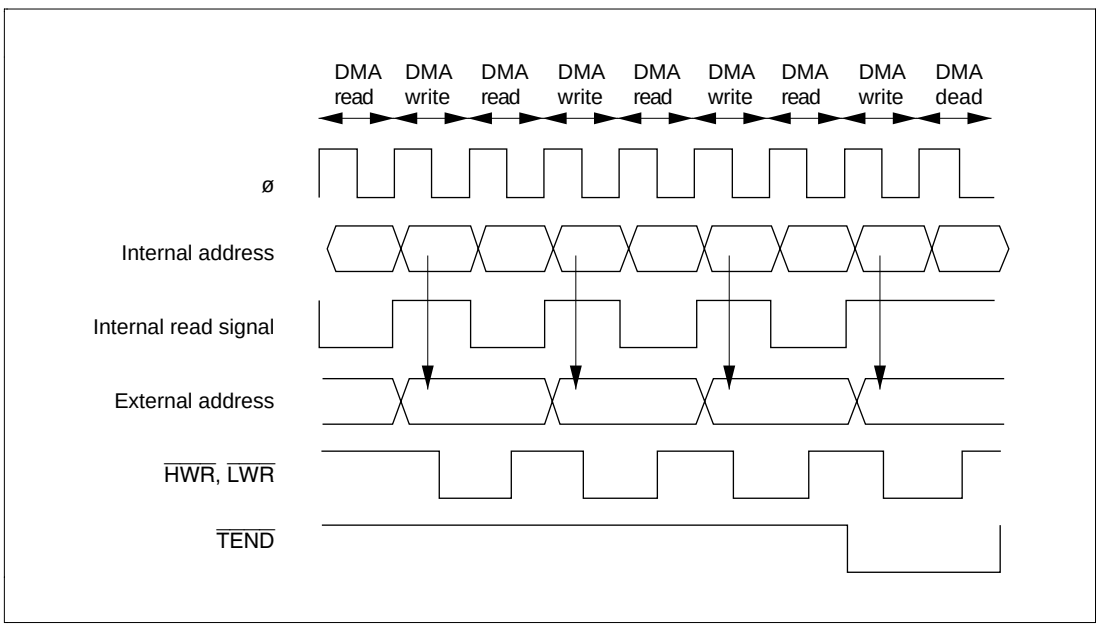


Figure 8-33 Example of Dual Address Transfer Using Write Data Buffer Function

Figure 8-34 shows an example of single address transfer using the write data buffer function. In this example, the CPU program area is in on-chip memory.

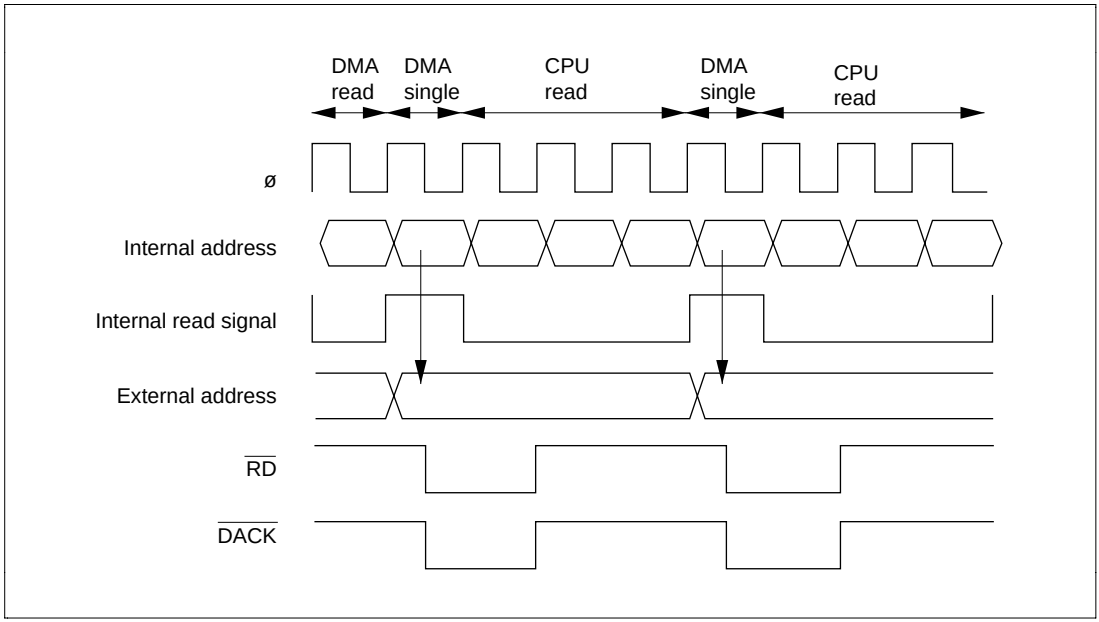


Figure 8-34 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

8.5.13 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 8-13 summarizes the priority order for DMAC channels.

Table 8-13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		
Channel 1A	Channel 1	
Channel 1B		Low

If transfer requests are issued simultaneously for more than one channel, or if a transfer request for another channel is issued during a transfer, when the bus is released the DMAC selects the highest-priority channel from among those issuing a request according to the priority order shown in table 8-13.

During burst transfer, or when one block is being transferred in block transfer, the channel will not be changed until the end of the transfer.

Figure 8-35 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

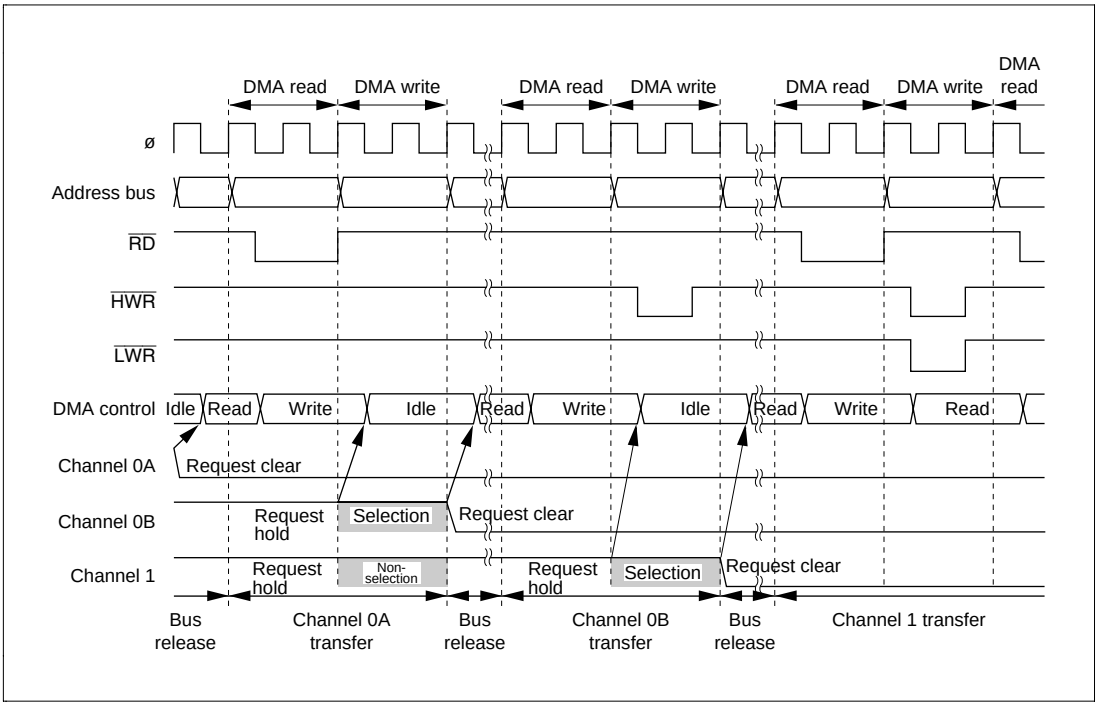


Figure 8-35 Example of Multi-Channel Transfer

8.5.14 Relation Between External Bus Requests, Refresh Cycles, the DTC, and the DMAC

There can be no break between a DMA cycle read and a DMA cycle write. This means that a refresh cycle, external bus release cycle, or DTC cycle is not generated between the external read and external write in a DMA cycle.

In the case of successive read and write cycles, such as in burst transfer or block transfer, a refresh or external bus released state may be inserted after a write cycle. Since the DTC has a lower priority than the DMAC, the DTC does not operate until the DMAC releases the bus.

When DMA cycle reads or writes are accesses to on-chip memory or internal I/O registers, these DMA cycles can be executed at the same time as refresh cycles or external bus release. However, simultaneous operation may not be possible when a write buffer is used.

8.5.15 NMI Interrupts and DMAC

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and the DTME bit are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 8-36 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

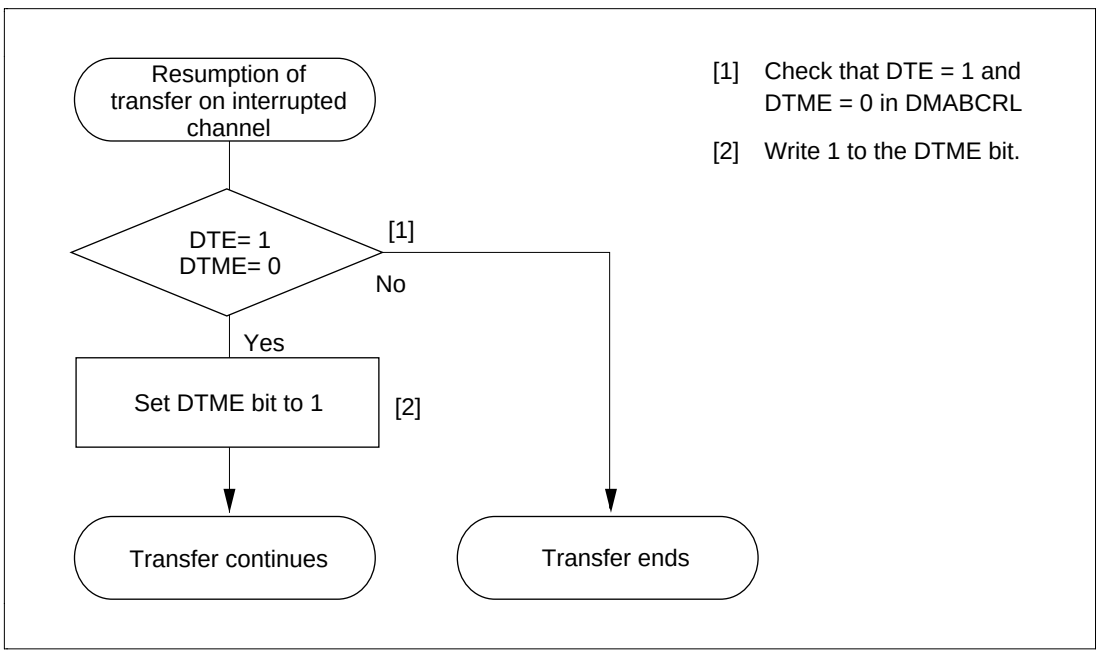


Figure 8-36 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

8.5.16 Forced Termination of DMAC Operation

If the DTE bit for the channel currently operating is cleared to 0, the DMAC stops on completion of the 1-byte or 1-word transfer in progress. DMAC operation resumes when the DTE bit is set to 1 again.

In full address mode, the same applies to the DTME bit.

Figure 8-37 shows the procedure for forcibly terminating DMAC operation by software.

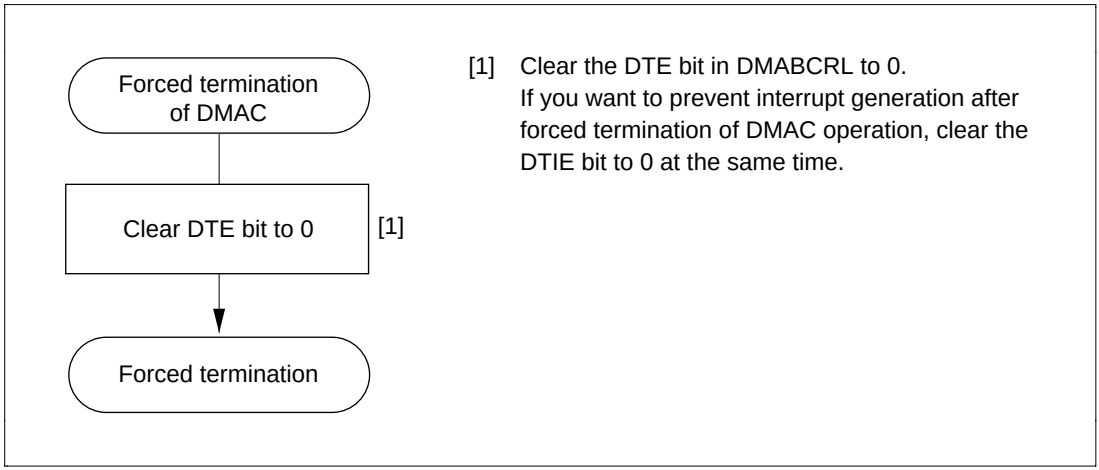
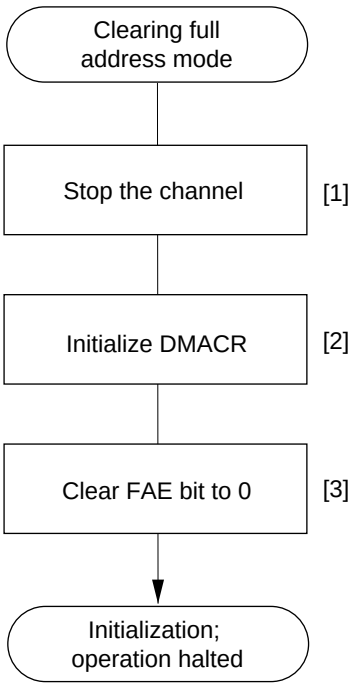


Figure 8-37 Example of Procedure for Forcibly Terminating DMAC Operation

8.5.17 Clearing Full Address Mode

Figure 8-38 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.



[1] Clear both the DTE bit and the DTME bit in DMABCRL to 0; or wait until the transfer ends and the DTE bit is cleared to 0, then clear the DTME bit to 0.
Also clear the corresponding DTIE bit to 0 at the same time.

[2] Clear all bits in DMACRA and DMACRB to 0.

[3] Clear the FAE bit in DMABCRH to 0.

Figure 8-38 Example of Procedure for Clearing Full Address Mode

8.6 Interrupts

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 8-14 shows the interrupt sources and their priority order.

Table 8-14 Interrupt Source Priority Order

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit for the corresponding channel in DMABCR, and interrupts from each source are sent to the interrupt controller independently.

The relative priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 8-14.

Figure 8-39 shows a block diagram of a transfer end/transfer break interrupt. An interrupt is always generated when the DTIE bit is set to 1 while DTE bit is cleared to 0.

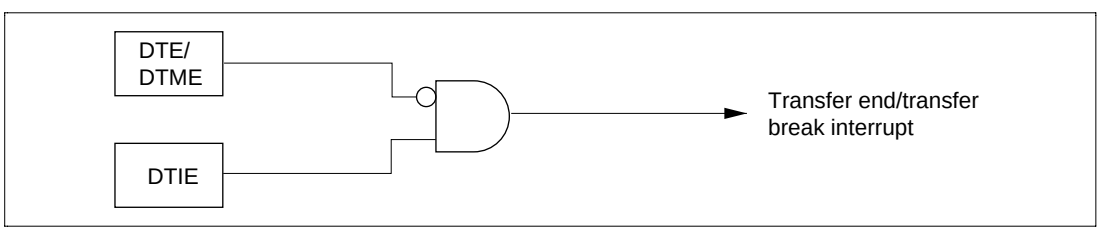


Figure 8-39 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to o while DTIEB bit is set to 1.

In both short address mode and full address mode, DMABCR should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

8.7 Usage Notes

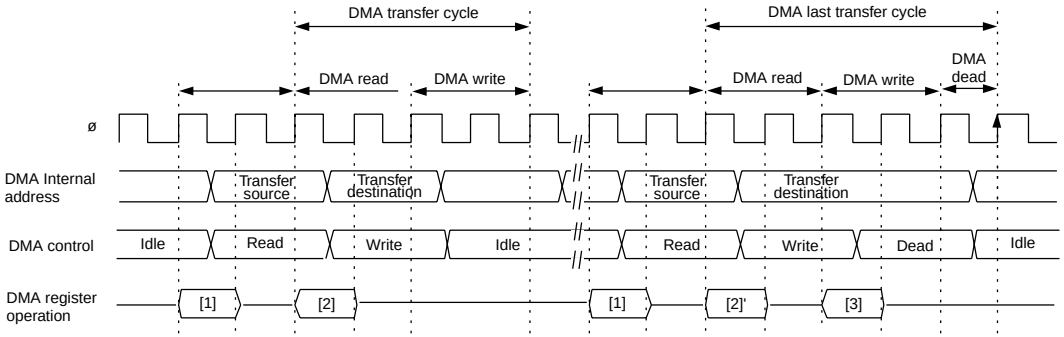
DMAC Register Access during Operation: Except for forced termination, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled.

Also, the DMAC register should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

(a) DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMAC transfer.

Figure 8-40 shows an example of the update timing for DMAC registers in dual address transfer mode.



[1] Transfer source address register MAR operation (incremented/decremented/fixed)
Transfer counter ETCR operation (decremented)
Block size counter ETCR operation (decremented in block transfer mode)

[2] Transfer destination address register MAR operation (incremented/decremented/fixed)

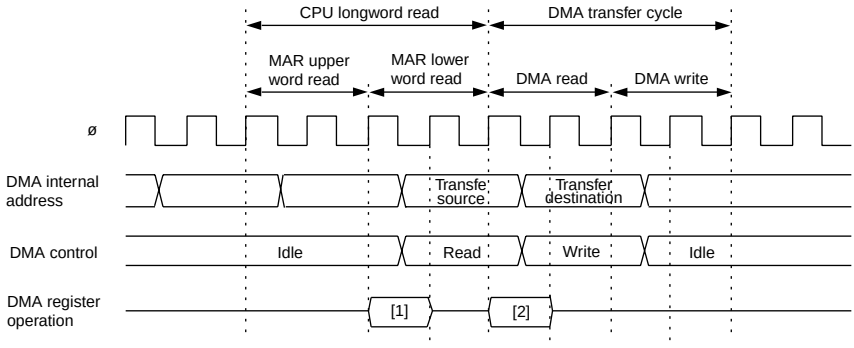
[2'] Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)

[3] Transfer address register MAR restore operation (in block or repeat transfer mode)
Transfer counter ETCR restore (in repeat transfer mode)
Block size counter ETCR restore (in block transfer mode)

Notes: 1. In single address transfer mode, the update timing is the same as [1].
2. The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 8-40 DMAC Register Update Timing

(b) If a DMAC transfer cycle occurs immediately after a DMAC register read cycle, the DMAC register is read as shown in figure 8-41.



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 8-41 Contention between DMAC Register Update and CPU Read

Module Stop: When the MSTPA7 bit in MSTPCR is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTPA7 bit if any of the DMAC channels is enabled. This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since the following DMAC register settings are valid even in the module stop state, they should be invalidated, if necessary, before a module stop.

- Transfer end/suspend interrupt (DTE = 0 and DTIE = 1)
- $\overline{\text{TEND}}$ pin enable (TEE = 1)
- $\overline{\text{DACK}}$ pin enable (FAE = 0 and SAE = 1)

Medium-Speed Mode: When the DTA bit is 0, internal interrupt signals specified as DMAC transfer sources are edge-detected.

In medium-speed mode, the DMAC operates on a medium-speed clock, while on-chip supporting modules operate on a high-speed clock. Consequently, if the period in which the relevant interrupt source is cleared by the CPU, DTC, or another DMAC channel, and the next interrupt is generated, is less than one state with respect to the DMAC clock (bus master clock), edge detection may not be possible and the interrupt may be ignored.

Also, in medium-speed mode, $\overline{\text{DREQ}}$ pin sampling is performed on the rising edge of the medium-speed clock.

Write Data Buffer Function: When the WDBE bit of BCRL in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel.

(a) Write Data Buffer Function and DMAC Register Setting

If the setting of is changed during execution of an external access by means of the write data buffer function, the external access may not be performed normally. The register that controls external accesses should only be manipulated when external reads, etc., are used with DMAC operation disabled, and the operation is not performed in parallel with external access.

(b) Write Data Buffer Function and DMAC Operation Timing

The DMAC can start its next operation during external access using the write data buffer function. Consequently, the $\overline{\text{DREQ}}$ pin sampling timing, $\overline{\text{TEND}}$ output timing, etc., are different from the case in which the write data buffer function is disabled. Also, internal bus cycles maybe hidden, and not visible.

(c) Write Data Buffer Function and $\overline{\text{TEND}}$ Output

A low level is not output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle. Note, for example, that a low level may not be output from the $\overline{\text{TEND}}$ pin if the write data buffer function is used when data transfer is performed between an internal I/O register and on-chip memory.

If at least one of the DMAC transfer addresses is an external address, a low level is output from the $\overline{\text{TEND}}$ pin.

Figure 8-42 shows an example in which a low level is not output at the $\overline{\text{TEND}}$ pin.

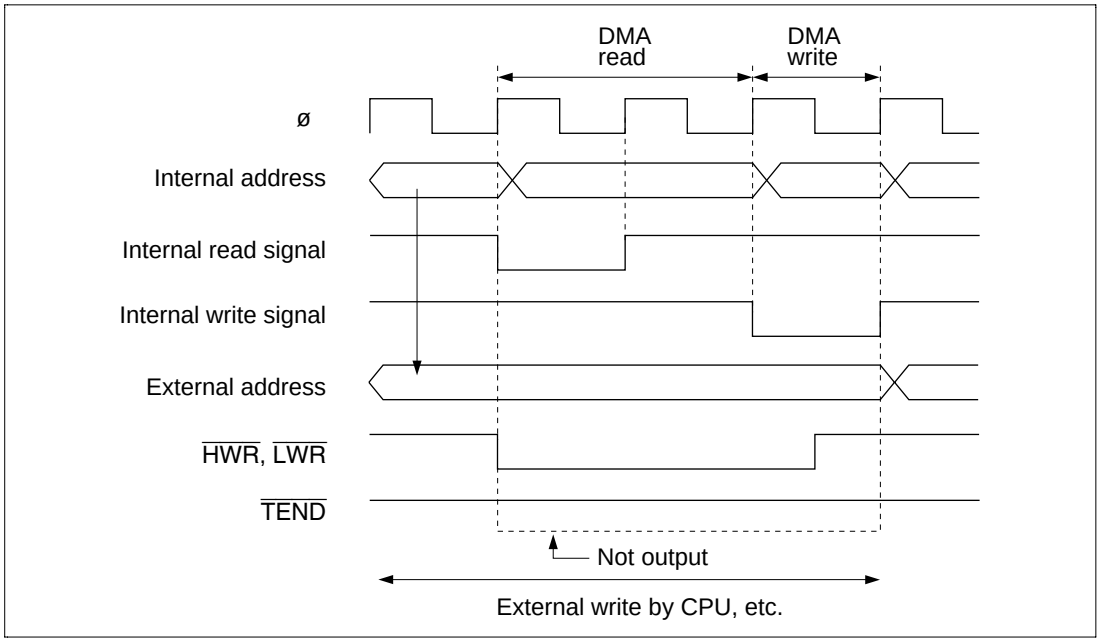


Figure 8-42 Example in Which Low Level is Not Output at $\overline{\text{TEND}}$ Pin

Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin: $\overline{\text{DREQ}}$ pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the $\overline{\text{DREQ}}$ pin, and switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].

[3] Activation request disabled state: Waits for detection of a high level on the $\overline{\text{DREQ}}$ pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed by detection of a low level.

Activation Source Acceptance: At the start of activation source acceptance, a low level is detected in both $\overline{\text{DREQ}}$ pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or $\overline{\text{DREQ}}$ pin low level that occurs before execution of the DMABCRL write to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or $\overline{\text{DREQ}}$ pin low level remaining from the end of the previous transfer, etc.

Internal Interrupt after End of Transfer: When the DTE bit is cleared to 0 by the end of transfer or an abort, the selected internal interrupt request will be sent to the CPU or DTC even if DTA is set to 1.

Also, if internal DMAC activation has already been initiated when operation is aborted, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if DTA is set to 1.

An internal interrupt request following the end of transfer or an abort should be handled by the CPU as necessary.

Channel Re-Setting: To reactivate a number of channels when multiple channels are enabled, use exclusive handling of transfer end interrupts, and perform DMABCR control bit operations exclusively.

Note, in particular, that in cases where multiple interrupts are generated between reading and writing of DMABCR, and a DMABCR operation is performed during new interrupt handling, the DMABCR write data in the original interrupt handling routine will be incorrect, and the write may invalidate the results of the operations by the multiple interrupts. Ensure that overlapping DMABCR operations are not performed by multiple interrupts, and that there is no separation between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, they must first be read while cleared to 0 before the CPU can write a 1 to them.

Section 9 Data Transfer Controller (DTC)

9.1 Overview

The H8S/2633 Series includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

9.1.1 Features

The features of the DTC are:

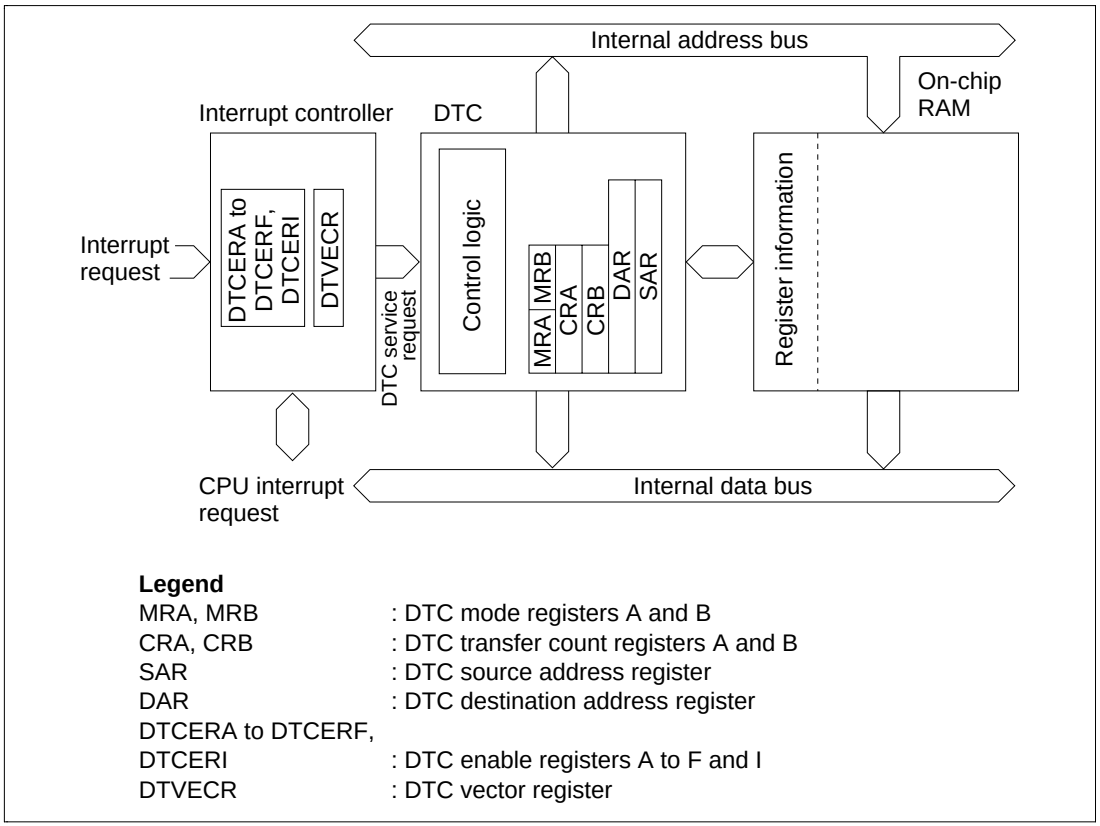
- Transfer possible over any number of channels
  - — Transfer information is stored in memory
  - — One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - — Normal, repeat, and block transfer modes available
  - — Incrementing, decrementing, and fixing of source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - — 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - — An interrupt request can be issued to the CPU after one data transfer ends
  - — An interrupt request can be issued to the CPU after the specified data transfers have completely ended
- Activation by software is possible
- Module stop mode can be set
  - — The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode.

9.1.2 Block Diagram

Figure 9-1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM*. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

Note: * When the DTC is used, the RAME bit in SYSCR must be set to 1.



Legend

MRA, MRB	: DTC mode registers A and B
CRA, CRB	: DTC transfer count registers A and B
SAR	: DTC source address register
DAR	: DTC destination address register
DTCERA to DTCERF, DTCERI	: DTC enable registers A to F and I
DTVECR	: DTC vector register

Figure 9-1 Block Diagram of DTC

9.1.3 Register Configuration

Table 9-1 summarizes the DTC registers.

Table 9-1 DTC Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
DTC mode register A	MRA	—*[2]	Undefined	—*[3]
DTC mode register B	MRB	—*[2]	Undefined	—*[3]
DTC source address register	SAR	—*[2]	Undefined	—*[3]
DTC destination address register	DAR	—*[2]	Undefined	—*[3]
DTC transfer count register A	CRA	—*[2]	Undefined	—*[3]
DTC transfer count register B	CRB	—*[2]	Undefined	—*[3]
DTC enable registers	DTCER	R/W	H'00	H'FE16 to H'FE1E
DTC vector register	DTVECR	R/W	H'00	H'FE1F
Module stop control register	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Registers within the DTC cannot be read or written to directly.

*3 Register information is located in on-chip RAM addresses H'EBC0 to H'EFBF. It cannot be located in external memory space. When the DTC is used, do not clear the RAME bit in SYSCR to 0.

9.2 Register Descriptions

9.2.1 DTC Mode Register A (MRA)

Bit	:	7	6	5	4	3	2	1	0
		SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—

MRA is an 8-bit register that controls the DTC operating mode.

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 3 and 2—DTC Mode (MD1, MD0): These bits specify the DTC transfer mode.

Bit 3	Bit 2	
MD1	**MD0**	**Description**
0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode.

Bit 1	
DTS	**Description**
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0	
Sz	**Description**
0	Byte-size transfer
1	Word-size transfer

9.2.2 DTC Mode Register B (MRB)

Bit	:	7	6	5	4	3	2	1	0
		CHNE	DISEL	—	—	—	—	—	—
Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain transfer, a number of data transfers can be performed consecutively in response to a single transfer request.

In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER is not performed.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferred)

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CPU are disabled or enabled after a data transfer.

Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not clear the interrupt source flag of the activating interrupt to 0)

Bits 5 to 0—Reserved: These bits have no effect on DTC operation in the H8S/2633 Series, and should always be written with 0.

9.2.3 DTC Source Address Register (SAR)

Bit	:	23	22	21	20	19	---	4	3	2	1	0

Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	---	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	---	—	—	—	—	—

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

9.2.4 DTC Destination Address Register (DAR)

Bit	:	23	22	21	20	19	---	4	3	2	1	0

Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	---	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	---	—	—	—	—	—

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

9.2.5 DTC Transfer Count Register A (CRA)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		← CRAH →								← CRAL →							

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00. This operation is repeated.

9.2.6 DTC Transfer Count Register B (CRB)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:		Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

9.2.7 DTC Enable Registers (DTCER)

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DTC enable registers comprise seven 8-bit readable/writable registers, DTCERA to DTCERF and DTCERI, with bits corresponding to the interrupt sources that can control enabling and disabling of DTC activation. These bits enable or disable DTC service for the corresponding interrupt sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby mode.

Bit n—DTC Activation Enable (DTCEn)

Bit n DTCEn	Description
0	DTC activation by this interrupt is disabled (Initial value) [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended

(n = 7 to 0)

A DTCE bit can be set for each interrupt source that can activate the DTC. The correspondence between interrupt sources and DTCE bits is shown in table 9-4, together with the vector number generated for each interrupt controller.

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

9.2.8 DTC Vector Register (DTVECR)

Bit	:	7	6	5	4	3	2	1	0
		SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/(W)*1	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: *1 Only 1 can be written to the SWDTE bit.
*2 Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

DTVECR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—DTC Software Activation Enable (SWDTE): Enables or disables DTC activation by software.

Bit 7

SWDTE	Description
0	DTC software activation is disabled (Initial value) [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 s written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): These bits specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a one-bit left-shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

9.2.9 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRA is a 8-bit readable/writable register that performs module stop mode control.

When the MSTPA6 bit in MSTPCRA is set to 1, the DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 6—Module Stop (MSTPA6): Specifies the DTC module stop mode.

Bit 6

MSTPA6	Description	
0	DTC module stop mode cleared	(Initial value)
1	DTC module stop mode set	

9.3 Operation

9.3.1 Overview

When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation.

Figure 9-2 shows a flowchart of DTC operation.

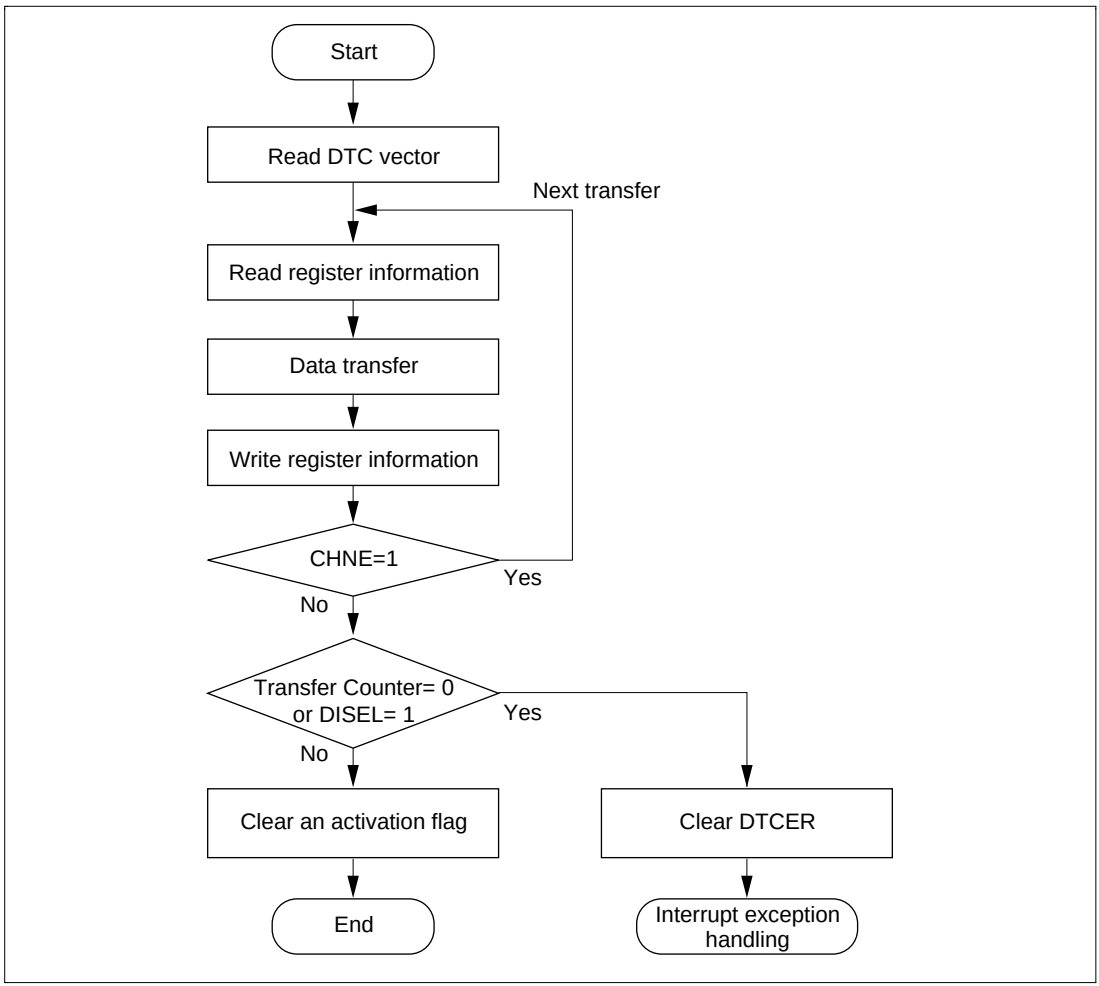


Figure 9-2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Table 9-2 outlines the functions of the DTC.

Table 9-2 DTC Functions

		Address Registers	
Transfer Mode	**Activation Source**	**Transfer Source**	**Transfer Destination**
• Normal mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — Up to 65,536 transfers possible • Repeat mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — After the specified number of transfers (1 to 256), the initial state resumes and operation continues • Block transfer mode — One transfer request transfers a block of the specified size — Block size is from 1 to 256 bytes or words — Up to 65,536 transfers possible — A block area can be designated at either the source or destination	• IRQ • TPU TGI • 8-bit timer CMI • SCI TXI or RXI • A/D converter ADI • DMAC DEND • Software	24 bits	24 bits

9.3.2 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. An interrupt becomes a DTC activation source when the corresponding bit is set to 1, and a CPU interrupt source when the bit is cleared to 0.

At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. Table 9-3 shows activation source and DTCER clearance. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI0.

Table 9-3 Activation Source and DTCER Clearance

Activation Source	When the DISEL Bit Is 0 and the Specified Number of Transfers Have Not Ended	When the DISEL Bit Is 1, or when the Specified Number of Transfers Have Ended
Software activation	The SWDTE bit is cleared to 0	The SWDTE bit remains set to 1 An interrupt is issued to the CPU
Interrupt activation	The corresponding DTCER bit remains set to 1 The activation source flag is cleared to 0	The corresponding DTCER bit is cleared to 0 The activation source flag remains set to 1 A request is issued to the CPU for the activation source interrupt

Figure 9-3 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

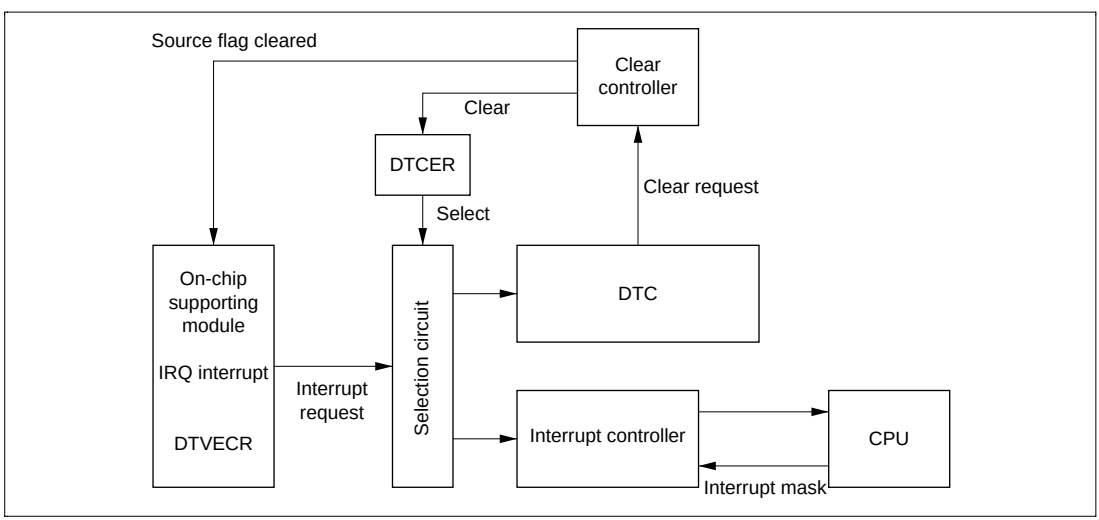


Figure 9-3 Block Diagram of DTC Activation Source Control

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

9.3.3 DTC Vector Table

Figure 9-4 shows the correspondence between DTC vector addresses and register information.

Table 9-4 shows the correspondence between activation and vector addresses. When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] << 1) (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the vector address is H'0420.

The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address. The register information can be placed at predetermined addresses in the on-chip RAM. The start address of the register information should be an integral multiple of four.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the address in the on-chip RAM.

Note: * Not available in the H8S/2633 Series.

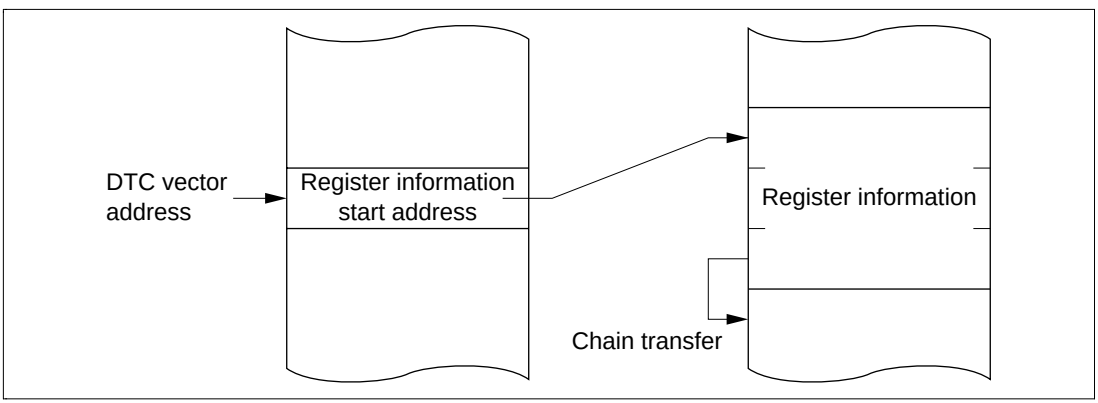


Figure 9-4 Correspondence between DTC Vector Address and Register Information

Table 9-4 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
Write to DTVECR	Software	DTVECR	H'0400+ (DTVECR [6:0] <<1)	—	High
IRQ0	External pin	16	H'0420	DTCEA7	
IRQ1		17	H'0422	DTCEA6	
IRQ2		18	H'0424	DTCEA5	
IRQ3		19	H'0426	DTCEA4	
IRQ4		20	H'0428	DTCEA3	
IRQ5		21	H'042A	DTCEA2	
IRQ6		22	H'042C	DTCEA1	
IRQ7		23	H'042E	DTCEA0	
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6	
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5	
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4	
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3	
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2	
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1	
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0	
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7	
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
TGI3A (GR3A compare match/ input capture)	TPU channel 3	48	H'0460	DTCEC5	High
TGI3B (GR3B compare match/ input capture)		49	H'0462	DTCEC4	
TGI3C (GR3C compare match/ input capture)		50	H'0464	DTCEC3	
TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2	
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1	
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0	
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5	
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4	
CMIA0 (compare match A0)	8-bit timer channel 0	64	H'0480	DTCED3	
CMIB0 (compare match B0)		65	H'0482	DTCED2	
CMIA1 (compare match A1)	8-bit timer channel 1	68	H'0488	DTCED1	
CMIB1 (compare match B1)		69	H'048A	DTCED0	
DEND0A (channel 0/channel 0A transfer end)	DMAC	72	H'0490	DTCEE7	
DEND0B (channel 0B transfer end)		73	H'0492	DTCEE6	
DEND1A (channel 1/channel 1A transfer end)		74	H'0494	DTCEE5	
DEND1B (channel 1B transfer end)		75	H'0496	DTCEE4	
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2	
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1	
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0	
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7	
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
CMIA2 (compare match A2)	8-bit timer channel 2	92	H'04B8	DTCEF5	High
CMIB2 (compare match B2)		93	H'04BA	DTCEF4	
CMIA3 (compare match A3)	8-bit timer channel 3	96	H'04C0	DTCEF3	
CMIB3 (compare match B3)		97	H'04C2	DTCEF2	
IICI0 (1-byte transmit/reception complete)	IIC channel 0 (option)	100	H'04C8	DTCEF1	
IICI1 (1-byte transmit/reception complete)	IIC channel 1 (option)	102	H'04CC	DTCEF0	
RXI3 (reception complete 3)	SCI channel 3	121	H'04F2	DTCEI7	
TXI3 (transmit data empty 3)		122	H'04F4	DTCEI6	
RXI4 (reception complete 4)	SCI channel 4	125	H'04FA	DTCEI5	
TXI4 (transmit data empty 4)		126	H'04FC	DTCEI4	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

9.3.4 Location of Register Information in Address Space

Figure 9-5 shows how the register information should be located in the address space.

Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information (contents of the vector address). In the case of chain transfer, register information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF).

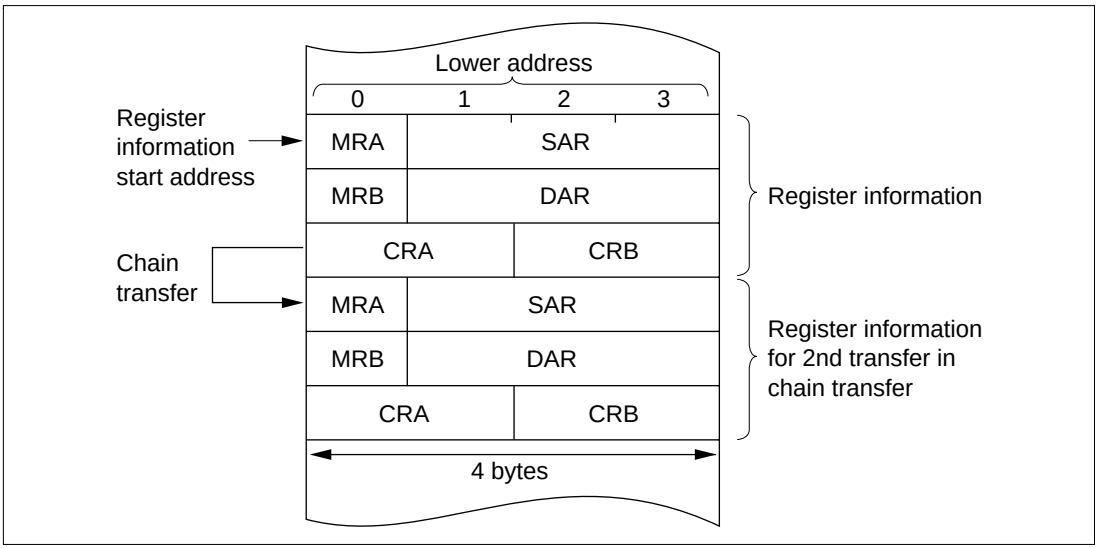


Figure 9-5 Location of Register Information in Address Space

9.3.5 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 9-5 lists the register information in normal mode and figure 9-6 shows memory mapping in normal mode.

Table 9-5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

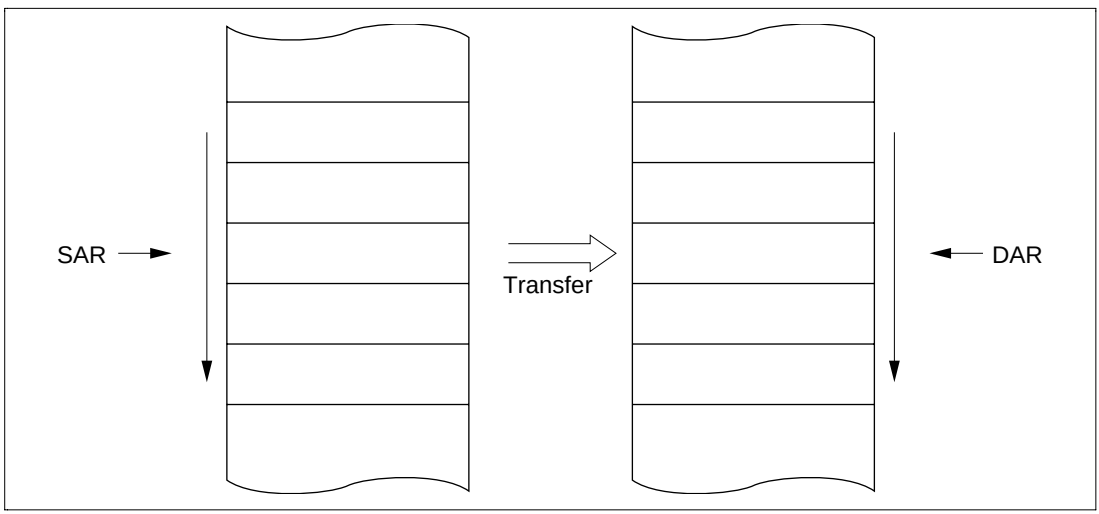


Figure 9-6 Memory Mapping in Normal Mode

9.3.6 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 9-6 lists the register information in repeat mode and figure 9-7 shows memory mapping in repeat mode.

Table 9-6 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

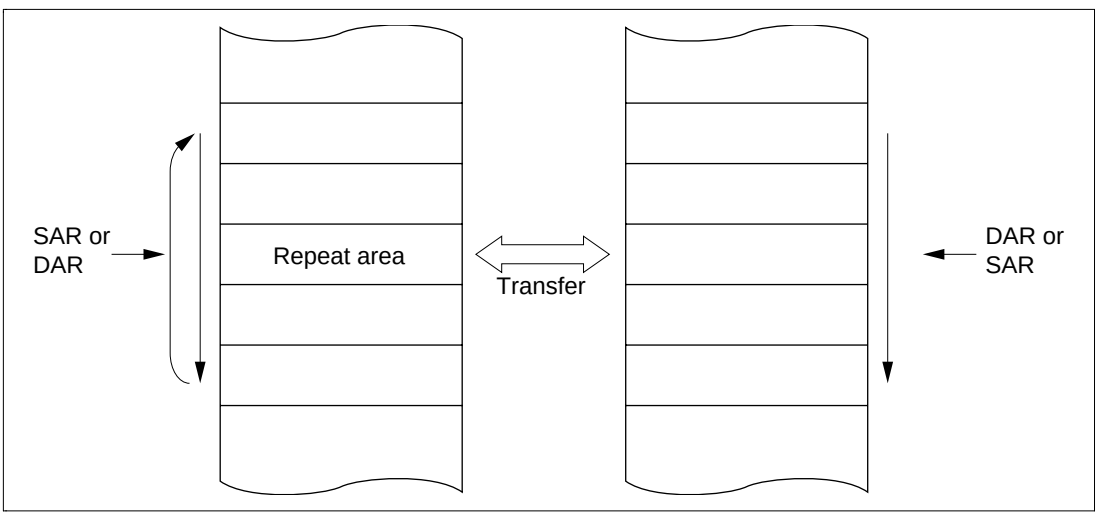


Figure 9-7 Memory Mapping in Repeat Mode

9.3.7 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt is requested.

Table 9-7 lists the register information in block transfer mode and figure 9-8 shows memory mapping in block transfer mode.

Table 9-7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Transfer count

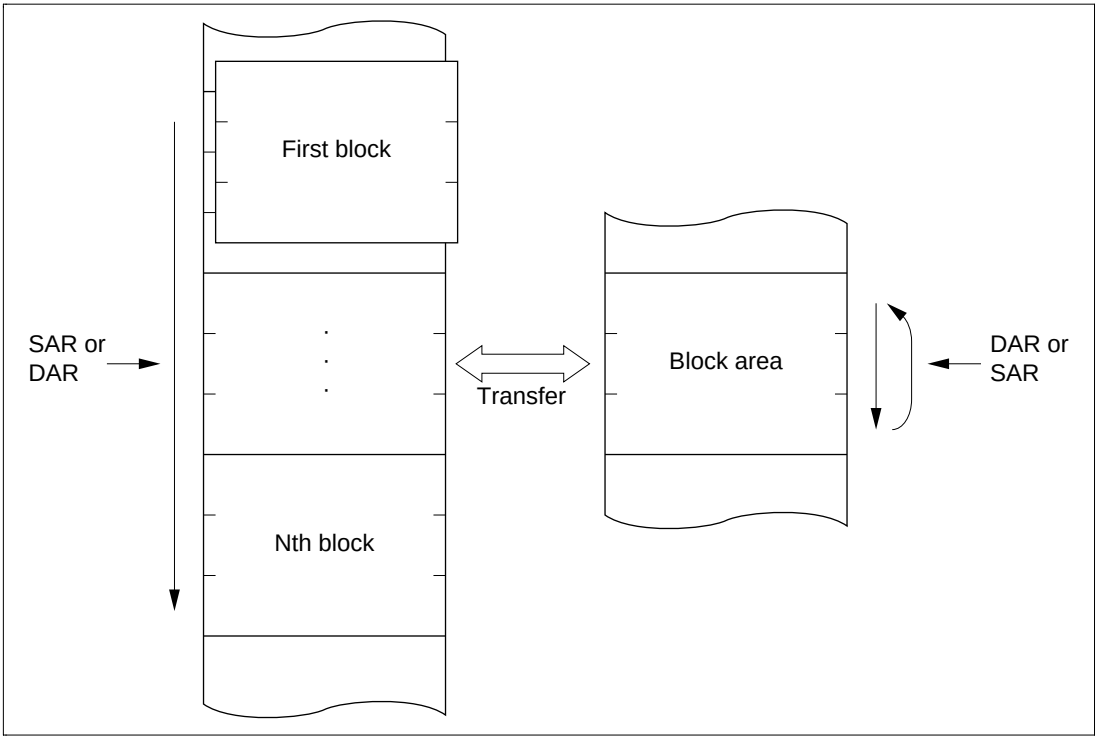


Figure 9-8 Memory Mapping in Block Transfer Mode

9.3.8 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consectutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 9-9 shows the memory map for chain transfer.

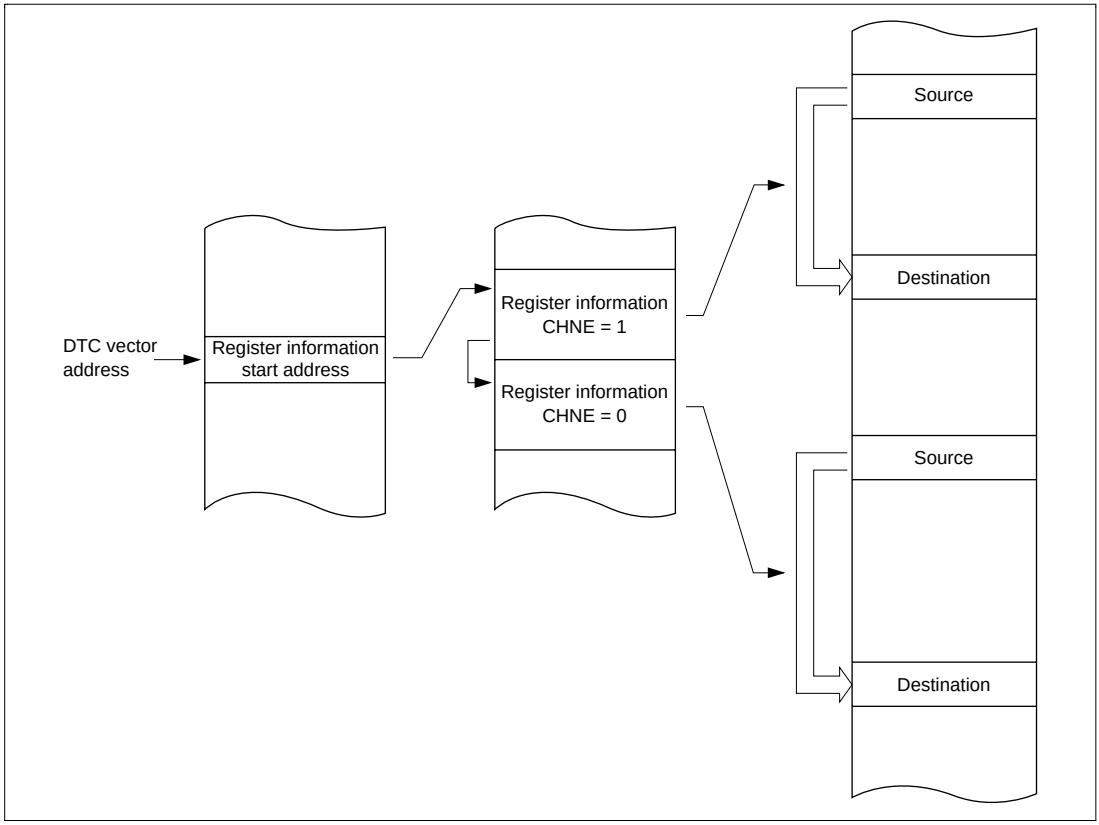


Figure 9-9 Chain Transfer Memory Map

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

9.3.9 Operation Timing

Figures 9-10 to 9-12 show an example of DTC operation timing.

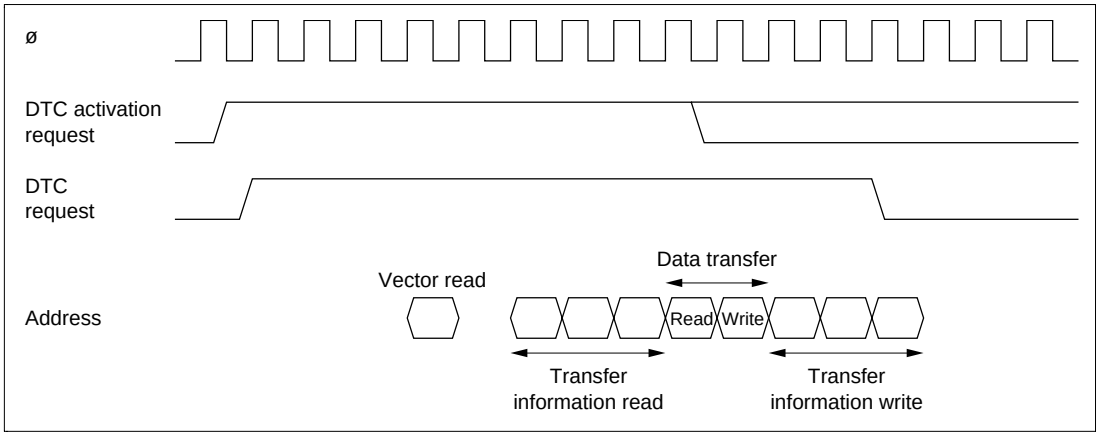


Figure 9-10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

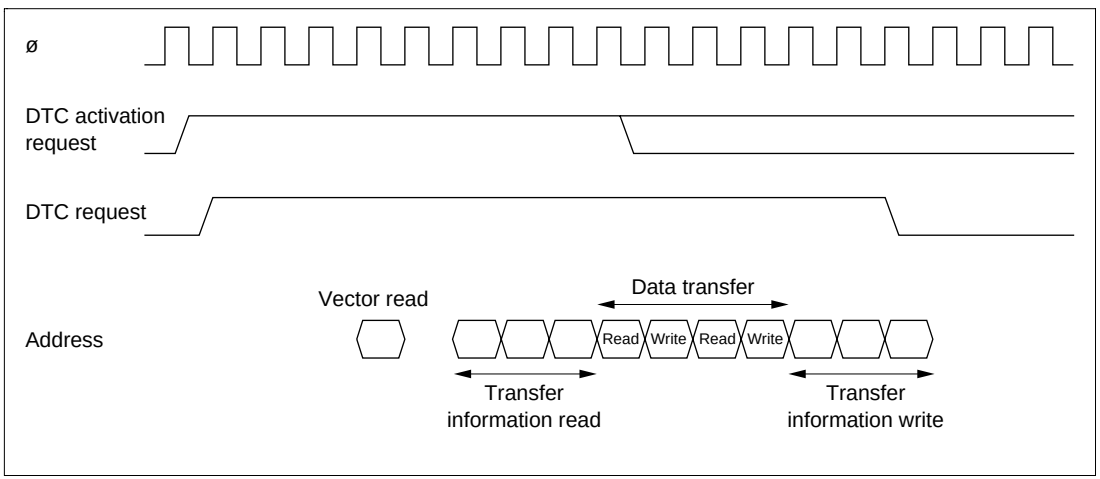


Figure 9-11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

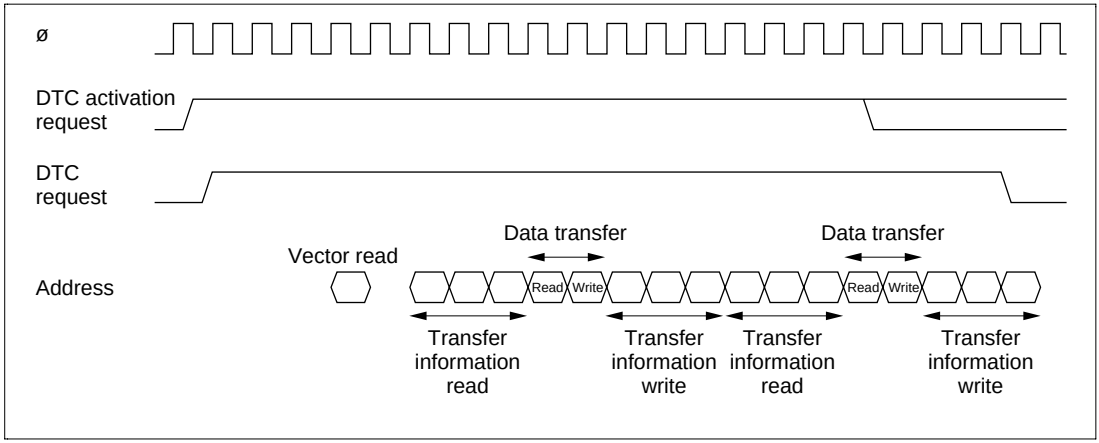


Figure 9-12 DTC Operation Timing (Example of Chain Transfer)

9.3.10 Number of DTC Execution States

Table 9-8 lists execution statuses for a single DTC data transfer, and table 9-9 shows the number of states required for each execution status.

Table 9-8 DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

N: Block size (initial setting of CRAH and CRAL)

Table 9-9 Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8		16	
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1							

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

Number of execution states = $I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from the on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

9.3.11 Procedures for Using DTC

Activation by Interrupt: The procedure for using the DTC with interrupt activation is as follows:

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.

[5] After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

Activation by Software: The procedure for using the DTC with software activation is as follows:

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Check that the SWDTE bit is 0.

[4] Write 1 to SWDTE bit and the vector number to DTVECR.

[5] Check the vector number written to DTVECR.

[6] After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

9.3.12 Examples of Use of the DTC

(1) Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

[1] Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.

[2] Set the start address of the register information at the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.

[5] Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.

[6] When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

(2) Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

[1] Perform settings for transfer to the PPG's NDR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.

[2] Perform settings for transfer to the TPU's TGR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.

[3] Locate the TPU transfer register information consecutively after the NDR transfer register information.

[4] Set the start address of the NDR transfer register information to the DTC vector address.

[5] Set the bit corresponding to TGIA in DTCER to 1.

[6] Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.

[7] Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.

[8] Set the CST bit in TSTR to 1, and start the TCNT count operation.

[9] Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.

[10] When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

(3) Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

[1] Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[2] Set the start address of the register information at the DTC vector address (H'04C0).

[3] Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.

[6] If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.

[7] After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

9.4 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

9.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DMAC Transfer End Interrupt: When the DTC is activated with a DMAC transfer end interrupt, the DMAC's DTE bit is not controlled by the DTC regardless of the transfer counter and DISEL bit, and write data takes precedence.

DTCE Bit Setting: For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

Section 10 I/O Ports

10.1 Overview

The H8S/2633 Series has 10 I/O ports (ports 1, 3, 7 and A to G), and two input-only port (ports 4 and 9).

Table 10-1 summarizes the port functions. The pins of each port also have other functions.

Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function, and in addition to DR and DDR, have a MOS input pull-up control register (PCR) to control the on/off state of MOS input pull-up.

Ports 3, and A to C include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

When ports 10 to 13, 70 to 73, and A to G are used as the output pins for expanded bus control signals, they can drive one TTL load plus a 50pF capacitance load. Those ports in other cases, and ports 14 to 17, 3, and 74 to 77, can drive one TTL load and a 30pF capacitance load. All IO ports can drive Darlington transistors when set to output.

See appendix C, I/O Port Block Diagrams, for a block diagram of each port.

Table 10-1 Port Functions

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Schmitt-triggered input (P16, P14)	P17/PO15/TIOCB2/ PWM3/TCLKD P16/PO14/TIOCA2/ PWM2/$\overline{\text{IRQ1}}$ P15/PO13/TIOCB1/ TCLKC P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$ P13/PO11/TIOCD0/ TCLKB/A23 P12/PO10/TIOCC0/ TCLKA/A22 P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$/A21 P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$/A20	8-bit I/O port also functioning as DMA controller output pins ($\overline{\text{DACK0}}$, $\overline{\text{DACK1}}$), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2), PPG output pins (PO15 to PO8), interrupt input pins ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$), 14-bit PWM output pins (PWM2, PWM3), and address outputs (A20 to A23)			8-bit I/O port also functioning as DMA controller output pins ($\overline{\text{DACK0}}$, $\overline{\text{DACK1}}$), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2), PPG output pins (PO15 to PO8), interrupt input pins ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$), and 14-bit PWM output pins (PWM2, PWM3)
Port 3	• 8-bit I/O port • Open-drain output capability • Schmitt-triggered input (P35, P32)	P37 /TxD4 P36/RxD4 P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$ P34 /RxD1/SDA0 P33 /TxD1/SCL1 P32 /SCK0/SDA1/$\overline{\text{IRQ4}}$ P31 /RxD0/IrRxD P30 /TxD0/IrTxD	8-bit I/O port also functioning as SCI (channel 0, 1, and 4) I/O pins (TxD0, RxD0, SCK0, IrTxD, IrRxD, TxD1, RxD1, SCK1, TxD4, RxD4, SCK4), interrupt input pins ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$), and IIC (channel 0 and 1) I/O pins (SCL0, SDA0, SCL1, SDA1)			
Port 4	• 8-bit input port	P47 /AN7/DA1 P46 /AN6/DA0 P45 /AN5 P44 /AN4 P43 /AN3 P42 /AN2 P41 /AN1 P40/AN0	8-bit input port also functioning as A/D converter analog inputs (AN7 to AN0) and D/A converter analog outputs (DA1, DA0)			

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port 7	• 8-bit I/O port	P77/TxD3 P76/RxD3 P75/TMO3/SCK3 P74/TMO2/$\overline{\text{MRES}}$ P73/TMO1/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$ P72/TMO0/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$/SYNCI P71/TMRI23/TMCI23/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$ P70/TMRI01/TMCI01/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	8-bit I/O port also functioning as 8-bit timer I/O pins (TMRI01, TMCI01, TMRI23, TMCI23, TMO0, TMO1, TMO2, TMO3), DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), the IIC input pin (SYNCI), SCI I/O pins (SCK3, RxD3, TxD3), and the manual reset input pin ($\overline{\text{MRES}}$)			8-bit I/O port also functioning as 8-bit timer I/O pins (TMRI01, TMCI01, TMRI23, TMCI23, TMO0, TMO1, TMO2, TMO3), DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), the IIC input pin (SYNCI), SCI I/O pins (SCK3, RxD3, TxD3), and the manual reset input pin ($\overline{\text{MRES}}$)
Port 9	• 8-bit input port	P97/AN15/DA3 P96/AN14/DA2 P95/AN13 P94/AN12 P93/AN11 P92/AN10 P91/AN9 P90/AN8	8-bit input port also functioning as A/D converter analog inputs (AN15 to AN8) and D/A converter analog outputs (DA3, DA2)			
Port A	• 4-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PA3/A19/SCK2 PA2/A18/RxD2 PA1/A17/TxD2 PA0/A16	4-bit I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2) and address outputs (A19 to A16)			4-bit I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2)

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port B	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PB7/A15/TIOCB5 PB6/A14/TIOCA5 PB5/A13/TIOCB4 PB4/A12/TIOCA4 PB3/A11/TIOCD3 PB2/A10/TIOCC3 PB1/A9/TIOCB3 PB0/A8/TIOCA3	8-bit I/O port also functioning as TPU I/O pins (TIOCB5, TIOCA5, TIOCB4, TIOCA4, TIOCD3, TIOCC3, TIOCB3, TIIOCA3) and address outputs (A15 to A8)			8-bit I/O port also functioning as TPU I/O pins (TIOCB5, TIOCA5, TIOCB4, TIOCA4, TIOCD3, TIOCC3, TIOCB3, TIIOCA3)
Port C	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PC7/A7/PWM1 PC6/A6/PWM0 PC5/A5 PC4/A4 PC3/A3 PC2/A2 PC1/A1 PC0 /A0	8-bit I/O port also functioning as 14-bit PWM (channel 1 and 0) output pins (PWM1, PWM0) and address outputs (A7 to A0)			8-bit I/O port also functioning as 14-bit PWM (channel 1 and 0) output pins (PWM1, PWM0)
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD7 /D15 PD6/D14 PD5/D13 PD4/D12 PD3/D11 PD2/D10 PD1/D9 PD0 /D8	Data bus input/output			I/O port
Port E	• 8-bit I/O port • Built-in MOS input pull-up	PE7/D7 PE6/D6 PE5/D5 PE4/D4 PE3/D3 PE2/D2 PE1/D1 PE0 /D0	In 8-bit-bus mode: I/O port In 16-bit-bus mode: data bus input/output			I/O port

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port F	• 8-bit I/O port	PF7 /ø	When DDR = 0: input port When DDR = 1 (after reset): ø output			When DDR = 0 (after reset): input port When DDR = 1: ø output
		PF6 /$\overline{\text{AS}}$/$\overline{\text{LCAS}}$ PF5 /$\overline{\text{RD}}$ PF4 /$\overline{\text{HWR}}$ PF3/$\overline{\text{LWR}}$/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	$\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$ outputs $\overline{\text{ADTRG}}$, $\overline{\text{IRQ3}}$ input When LCASS = 0: $\overline{\text{AS}}$ output When RMTS2 to RMTS0 = B'001 to B'011, CW2 = 0, and LCASS = 1: $\overline{\text{LCAS}}$ output			I/O port $\overline{\text{ADTRG}}$, $\overline{\text{IRQ3}}$ input
		PF2/$\overline{\text{LCAS}}$/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	When WAITE = 0 and BREQOE = 0 (after reset): I/O port When WAITE = 1 and BREQOE = 0: $\overline{\text{WAIT}}$ input When WAITE = 0 and BREQOE = 1: $\overline{\text{BREQO}}$ input When RMTS2 to RMTS0 = B'001 to B'011, CW2 = 0, and LCASS = 0: $\overline{\text{LCAS}}$ output			I/O port
		PF1/$\overline{\text{BACK}}$/BUZZ PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{\text{BREQ}}$ input, $\overline{\text{BACK}}$ output BUZZ output, $\overline{\text{IRQ2}}$ input			BUZZ output $\overline{\text{IRQ2}}$ input I/O port
Port G	• 5-bit I/O port	PG4 /$\overline{\text{CS0}}$	When DDR = 0*1: input port When DDR = 1*2: $\overline{\text{CS0}}$ output			I/O port
		PG3 /$\overline{\text{CS1}}$ PG2 /$\overline{\text{CS2}}$ PG1 /$\overline{\text{CS3}}$/$\overline{\text{OE}}$/$\overline{\text{IRQ7}}$	When DDR = 0 (after reset): input port When DDR = 1: $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$ outputs $\overline{\text{OE}}$ output, $\overline{\text{IRQ7}}$ input			I/O port, $\overline{\text{IRQ7}}$ input
		PG0 /$\overline{\text{CAS}}$/$\overline{\text{IRQ6}}$	DRAM space set: $\overline{\text{CAS}}$ output Otherwise (after reset): I/O port $\overline{\text{IRQ6}}$ input			I/O port, $\overline{\text{IRQ6}}$ input

Notes: *1 After a reset in mode 6

*2 After a reset in mode 4 or 5

10.2 Port 1

10.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 pins also function as PPG output pins (PO15 to PO8), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), DMAC output pins ($\overline{\text{DACK0}}$, $\overline{\text{DACK1}}$), 14-bit PWM output pins (PWM2, PWM3) external interrupt pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), and address bus output pins (A23 to A20). Port 1 pin functions change according to the operating mode.

Figure 10-1 shows the port 1 pin configuration.

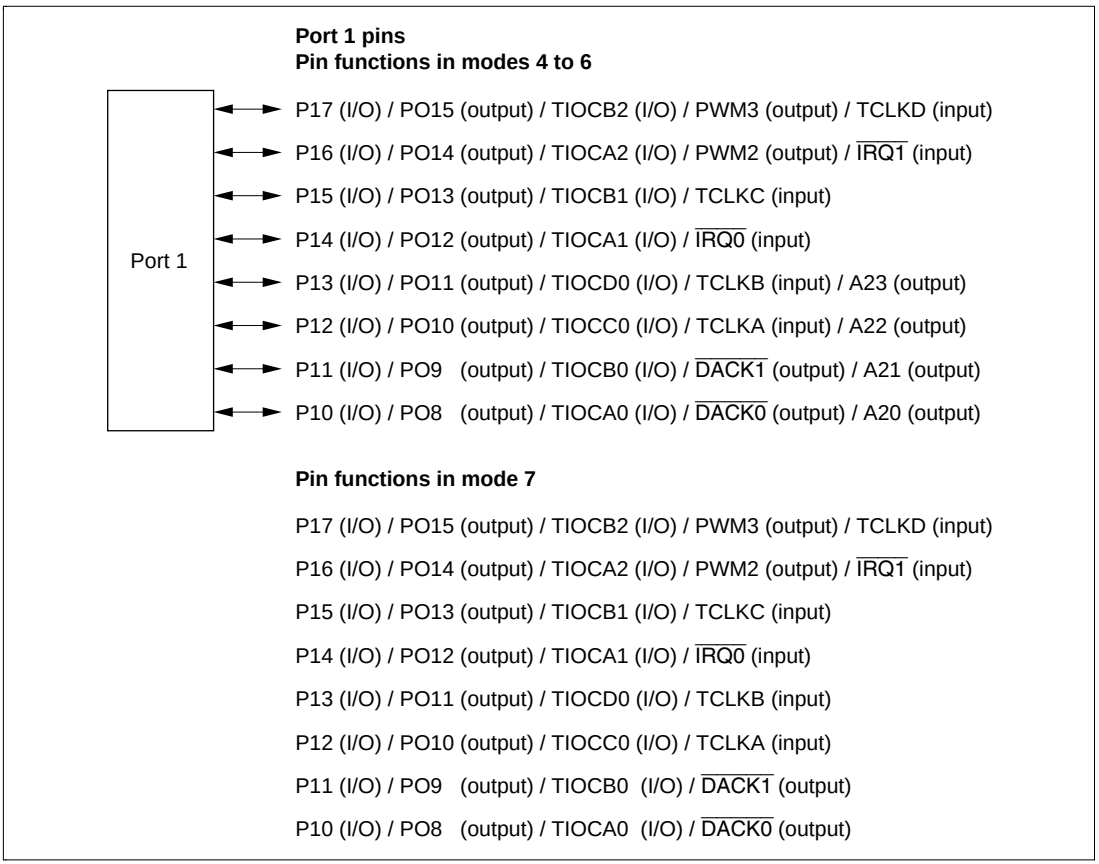


Figure 10-1 Port 1 Pin Functions

10.2.2 Register Configuration

Table 10-2 shows the port 1 register configuration.

Table 10-2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 1 data direction register	P1DDR	W	H'00	H'FE30
Port 1 data register	P1DR	R/W	H'00	H'FF00
Port 1 register	PORT1	R	Undefined	H'FFB0

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P1DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode. Because PPG, TPU, and DMAC are initialized at a manual reset, pin states are determined by P1DDR and P1DR.

Port 1 Data Register (P1DR)

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P17 to P10).

P1DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins P17 to P10.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 1 pins (P17 to P10) must always be performed on P1DR.

If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT1 contents are determined by the pin states, as P1DDR and P1DR are initialized. PORT1 retains its prior state by a manual reset or in software standby mode.

10.2.3 Pin Functions

Port 1 pins also function as PPG output pins (PO15 to PO8), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), DMAC output pins ($\overline{\text{DACK0}}$ and $\overline{\text{DACK1}}$), external interrupt input pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), 14-bit PWM output pins (PWM2 and PWM3), and address bus output pins (A23 to A20). Port 1 pin functions are shown in table 10-3.

Table 10-3 Port 1 Pin Functions

Pin | **Selection Method and Pin Functions**

P17/PO15/ TIOCB2/PWM3/ TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, OEB bit in DACR3, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)			
OEB	—	0	0	0	1
P17DDR	—	0	1	1	—
NDER15	—	—	0	1	—
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output	PWM3 output
		TIOCB2 input *1			
	TCLKD input *2				

Notes: *1 TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1.

*2 TCLKD input when the setting for either TCR0 or TCR5 is: TPSC2 to TPSC0 = B'111.
TCLKD input when channels 2 and 4 are set to phase counting mode.

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
P16/PO14/ TIOCA2/PWM2/ $\overline{\text{IRQ1}}$	The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOA3 to IOA0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), OEA bit in DACR3, bit NDER14 in NDERH, and bit P16DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)			
OEA	—	0	0	0	1
P16DDR	—	0	1	1	—
NDER14	—	—	0	1	—
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output	PWM2 output
		TIOCA2 input *1			
	$\overline{\text{IRQ1}}$ input				

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output *2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1.

*2 TIOCB2 output is disabled.

Pin	Selection Method and Pin Functions
P15/PO13/ TIOCB1/TCLKC	The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, bit NDER13 in NDERH, and bit P15DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output
		TIOCB1 input *1		
	TCLKC input *2			

Notes: *1 TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.

*2 TCLKC input when the setting for either TCR0 or TCR2 is: TPSC2 to TPSC0 = B'110; or when the setting for either TCR4 or TCR5 is TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase counting mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
P14/PO12/ TIOCA1/$\overline{\text{IRQ0}}$	The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bit NDER12 in NDERH, and bit P14DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output
		TIOCA1 input *1		
	$\overline{\text{IRQ0}}$ input			

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx.

*2 TIOCB1 output is disabled.

Pin	Selection Method and Pin Functions
P13/PO11/ TIOCD0/TCLKB/ A23	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bits AE3 to AE0 in PFCR, bit NDER11 in NDERH, and bit P13DDR.

Operating mode	Modes 4 to 6				
AE3 to AE0	B'0000 to B'1110				B'1111
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P13DDR	—	0	1	1	—
NDER11	—	—	0	1	—
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output	A23 output
		TIOCD0 input *1			
	TCLKB input *2				

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input *1		
	TCLKB input *2			

Notes: *1 TIOCD0 input when MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx.

*2 TCLKB input when the setting for TCR0 to TCR2 is: TPSC2 to TPSC0 = B'101.

TCLKB input when channels 1 and 5 are set to phase counting mode.

Pin	Selection Method and Pin Functions						
P13/PO11/ TIOCD0/TCLKB/ A23 (cont)	TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
	MD3 to MD0	B'0000		B'0010	B'0011		
	IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
	CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
	Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

Pin	Selection Method and Pin Functions
P12/PO10/ TIOCC0/TCLKA/ A22	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bits AE3 to AE0 in PFCR, bit NDER10 in NDERH, and bit P12DDR.

Operating mode	Modes 4 to 6				
AE3 to AE0	B'0000 to B'1110				B'1111
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P12DDR	—	0	1	1	—
NDER10	—	—	0	1	—
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output	A22 output
		TIOCC0 input *1			
	TCLKA input *2				

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input *1		
	TCLKA input *2			

Pin	Selection Method and Pin Functions

P12/PO10/ TIOCC0/TCLKA/ A22 (cont)

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCC0 input when MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx.

*2 TCLKA input when the setting for TCR0 to TCR5 is: TPSC2 to TPSC0 = B'100.

TCLKA input when channels 1 and 5 are set to phase counting mode.

*3 TIOCD0 output is disabled.
When BFA = 1 or BFB = 1 in TMDR0, output is disabled and setting (2) applies.

Pin	Selection Method and Pin Functions
P11/PO9/TIOCB0/$\overline{\text{DACK1}}$/A21	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, and bits IOB3 to IOB0 in TIOR0H), bits AE3 to AE0 in PFCR, bit NDER9 in NDERH, SAE1 bit in DMABCRH, and bit P11DDR.

Operating mode	Modes 4 to 6					
AE3 to AE0	B'0000 to B'1101					B'1110 to B'1111
SAE1	0				1	—
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—	—
P11DDR	—	0	1	1	—	—
NDER9	—	—	0	1	—	—
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output	$\overline{\text{DACK1}}$ output	A21 output
		TIOCB0 input *1				

Operating mode	Mode 7				
AE3 to AE0	—				
SAE1	0				1
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P11DDR	—	0	1	1	—
NDER9	—	—	0	1	—
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output	$\overline{\text{DACK1}}$ output
		TIOCB0 input *1			

Note: *1 TIOCB0 input when MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx.

Pin	Selection Method and Pin Functions						
P11/PO9/TIOCB0/ $\overline{\text{DACK1}}$/A21 (cont)	TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
	MD3 to MD0	B'0000		B'0010	B'0011		
	IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
	CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
	Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
P10/PO8/TIOCA0/$\overline{\text{DACK0}}$/A20	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bits AE3 to AE0 in PFCR, bit NDER8 in NDERH, SAE0 bit in DMABCRH, and bit P10DDR.

Operating mode	Modes 4 to 6					
AE3 to AE0	B'0000 to B'1110					B'1101 to B'1111
SAE0	0				1	—
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—	—
P10DDR	—	0	1	1	—	—
NDER8	—	—	0	1	—	—
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output	$\overline{\text{DACK0}}$ output	A20 output
		TIOCA0 input *[1]				

Operating mode	Mode 7				
AE3 to AE0	—				
SAE0	0				1
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P10DDR	—	0	1	1	—
NDER8	—	—	0	1	—
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output	$\overline{\text{DACK0}}$ output
		TIOCA0 input *[1]			

Pin	Selection Method and Pin Functions						
P10/PO8/TIOCA0/ $\overline{\text{DACK0}}$/A20 (cont)	TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
	MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
	IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
	CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
	Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA0 input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

*2 TIOCB0 output is disabled.

10.3 Port 3

10.3.1 Overview

Port 3 is an 8-bit I/O port. Port 3 is a multi-purpose port for SCI I/O pins (TxD0, RxD0, SCK0, IrTxD, IrRxD, TxD1, RxD1, SCK1, TxD4, RxD4, SCK4), external interrupt input pins ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$) and IIC I/O pins (SCL0, SDA0, SCL1, SDA1). All of the port 3 pin functions have the same operating mode. The configuration for each of the port 3 pins is shown in figure 10-2.

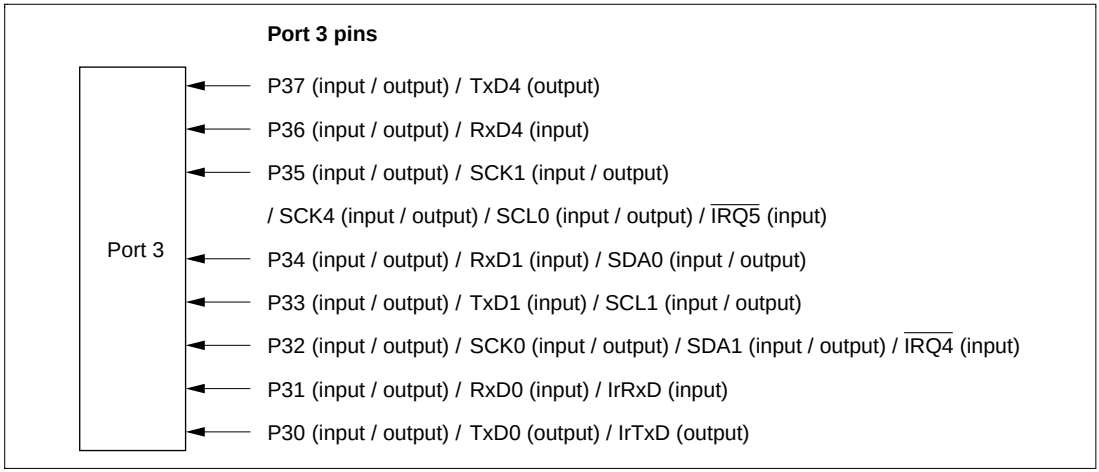


Figure 10-2 Port 3 Pin Functions

10.3.2 Register Configuration

Table 10-4 shows the configuration of port 3 registers.

Table 10-4 Port 3 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*
Port 3 data direction register	P3DDR	W	H'00	H'FE32
Port 3 data register	P3DR	R/W	H'00	H'FF02
Port 3 register	PORT3	R	Undefined	H'FFB2
Port 3 open drain control register	P3ODR	R/W	H'00	H'FE46

Note: * Indicates the lower-place 16 bits.

Port 3 Data Direction Register (P3DDR)

Bit	:	7	6	5	4	3	2	1	0
		P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P3DDR is an 8-bit write-dedicated register, which specifies the I/O for each port 3 pin by bit. Read is disenabled. If a read is carried out, undefined values are read out.

By setting P3DDR to 1, the corresponding port 3 pins become output, and be clearing to 0 they become input.

P3DDR is initialized to H'00 by a power-on reset and in hardware standby mode. The previous state is maintained by a manual reset and in software standby mode. SCI and IIC are initialized, so the pin state is determined by the specification of P3DDR and P3DR.

Port 3 Data Register (P3DR)

Bit	:	7	6	5	4	3	2	1	0
		P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register, which stores the output data of port 3 pins (P35 to P30).

P3DR is initialized to H'00 by a power-on reset and in hardware standby mode. The previous state is maintained by a manual reset and in software standby mode.

Port 3 Register (PORT3)

Bit	:	7	6	5	4	3	2	1	0
		P37	P36	P35	P34	P33	P32	P31	P30
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P37 to P30.

PORT3 is an 8-bit read-dedicated register, which reflects the state of pins. Write is disenabled. Always carry out writing off output data of port 3 pins (P37 to P30) to P3DR without fail.

When P3DDR is set to 1, if port 3 is read, the values of P3DR are read. When P3DDR is cleared to 0, if port 3 is read, the states of pins are read out.

P3DDR and P3DR are initialized by a power-on reset and in hardware standby mode, so PORT3 is determined by the state of the pins. The previous state is maintained by a manual reset and in software standby mode.

Port 3 Open Drain Control Register (P3ODR)

Bit	:	7	6	5	4	3	2	1	0
		P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P3ODR is an 8-bit readable/writable register, which controls the on/off of port 3 pins (P37 to P30).

By setting P3ODR to 1, the port 3 pins become an open drain out, and when cleared to 0 they become CMOS output.

P3ODR is initialized to H'00 by a power-on reset and in hardware standby mode. The previous state is maintained by a manual reset and in software standby mode.

10.3.3 Pin Functions

The port 3 pins double as SCI I/O input pins (TxD0, RxD0, SCK0, TxD1, RxD1, and SCK1). The functions of port 3 pins are shown in Table 10-5.

Table 10-5 Port 3 Pin Functions

<table>
<tr><th>Pin</th><th>Selection Method and Pin Functions</th></tr>
<tr><td>P37/TxD4</td><td>Switches as follows according to combinations of SCR TE bit of SCI4 and the P37DDR bit.
<table>
<tr><td>TE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>P37DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>P37 input pin</td><td>P37 output pin*</td><td>TxD4 output pin</td></tr>
</table>
Note: * When P37ODR = 1, it becomes NMOS open drain output.</td></tr>
<tr><td>P36/RxD4</td><td>Switches as follows according to combinations of SCR RE bit of SCI4 and the P36DDR bit.
<table>
<tr><td>RE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>P36DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>P36 input pin</td><td>P36 output pin*</td><td>RxD4 input pin</td></tr>
</table>
Note: * When P36ODR = 1, it becomes NMOS open drain output.</td></tr>
<tr><td>P35/SCK1/
SCK4/SCL0/
$\overline{\text{IRQ5}}$</td><td>Switches as follows according to combinations of ICCR0 ICE bit of IIC0, SMR C/$\overline{\text{A}}$ bit of SCI1 or SCI4, SCR CKE0 and CKE1 bits, and the P35DDR bit.
When used as a SCL0 I/O pin, always be sure to clear the following bits to 0: SMR C/$\overline{\text{A}}$ bits of SCI1 or SCI4, and SCR CKE0 and CKE1 bits. Do not set SCK1 and SCK4 to simultaneous output.
The SCL0 output format is NMOS open drain output, enabling direct bus driving.
<table>
<tr><td>ICE</td><td colspan="5">0</td><td>1</td></tr>
<tr><td>CKE1</td><td colspan="4">0</td><td>1</td><td>0</td></tr>
<tr><td>C/$\overline{\text{A}}$</td><td colspan="3">0</td><td>1</td><td>—</td><td>0</td></tr>
<tr><td>CKE0</td><td colspan="2">0</td><td>1</td><td>—</td><td>—</td><td>0</td></tr>
<tr><td>P35DDR</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="2">Pin function</td><td>P35
input pin</td><td>P35
output pin*</td><td>SCK1/
SCK4
output pin*</td><td>SCK1/
SCK4
output pin*</td><td>SCK1/
SCK4
input pin</td><td>SCL0
I/O pin</td></tr>
<tr><td colspan="6">$\overline{\text{IRQ5}}$ input</td></tr>
</table>
Note: * Output type is NMOS push-pull. When P35ODR = 1, it becomes NMOS open drain output.</td></tr>
</table>

Pin	Selection Method and Pin Functions
P34/RxD1/ SDA0	Switches as follows according to combinations of ICCR0 ICE bit of IIC0, SCR RE bit of SCI1, and the P34DDR bit. The SDA0 output format becomes NMOS open drain output, enabling direct bus driving.

ICE	0			1
RE	0		1	—
P34DDR	0	1	—	—
Pin function	P34 input pin	P34 output pin*	RxD1 input pin	SDA0 I/O pin

Note: * Output type is NMOS push-pull. When P34ODR = 1, it becomes NMOS open drain tray.

P33/TxD1/ SCL1	Switches as follows according to combinations of ICCR1 ICE bit of IIC1, SCR TE bit of SCI1 and the P33DDR bit. The SCL1 output format becomes NMOS open drain output, enabling direct bus driving.

ICE	0			1
TE	0		1	—
P33DDR	0	1	—	—
Pin function	P33 input pin	P33 output pin*	TxD1 output pin*	SCL1 I/O pin*

Note: * When P33ODR = 1, it becomes NMOS open drain output.

P32/SCK0/ SDA1/$\overline{\text{IRQ4}}$	Switches as follows according to combinations of ICCR1 ICE bit of IIC1, SMR C/$\overline{\text{A}}$ bit of SCI0, SCR CKE0 and CKE1 bits, and the P32DDR bit. If using as an SDA1 input pin, always set SMR C/$\overline{\text{A}}$ bit of SCI0 and SCR CKE0 and CKE1 bits to 0 without fail. The SDA1 output format becomes NMOS open drain output, enabling direct bus driving.

ICE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P32DDR	0	1	—	—	—	—
Pin function	P32 input pin	P32 output pin	SCK0 output pin*	SCK0 output pin*	SCK0 input pin	SDA1 I/O pin
	$\overline{\text{IRQ4}}$ input					

Note: * When P32ODR = 1, it becomes NMOS open drain output.

Pin	Selection Method and Pin Functions
P31/RxD0/ IrRxD	Switches as follows according to combinations of SCR RE bit of SCI0 and the P31DDR bit.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input pin	P31 output pin*	RxD0/IrRxD input pin

Note: * When P31ODR = 1, it becomes NMOS open drain output.

P30/TxD0/ IrTxD	Switches as follows according to combinations of SCR TE bit of SCI0 and the P30DDR bit.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input pin	P30 output pin*	TxD0/IrTxD output pin*

Note: * When P30ODR = 1, it becomes NMOS open drain output.

10.4 Port 4

10.4.1 Overview

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins (AN0 to AN7) and D/A converter analog output pins (DA0, DA1). Port 4 pin functions are the same in all operating modes. Figure 10-3 shows the port 4 pin configuration.

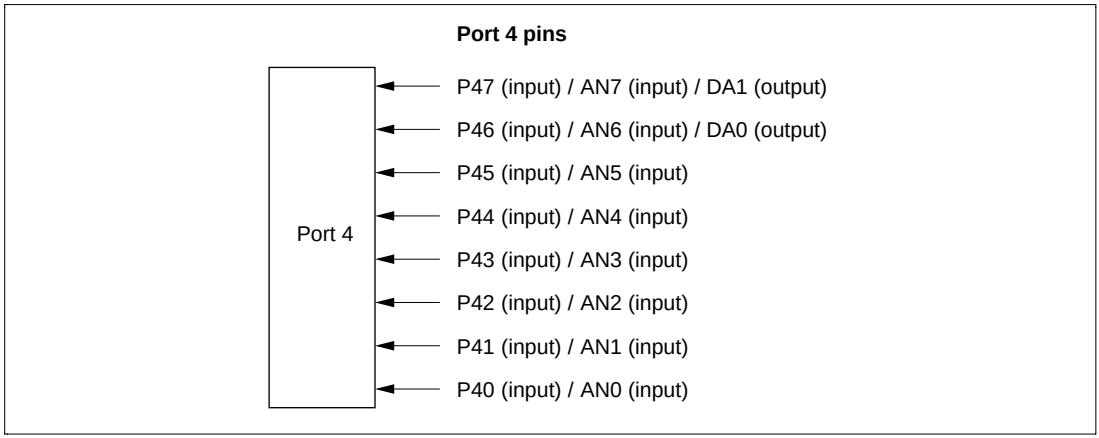


Figure 10-3 Port 4 Pin Functions

10.4.2 Register Configuration

Table 10-6 shows the port 4 register configuration. Port 4 is an input-only port, and does not have a data direction register or data register.

Table 10-6 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 4 register	PORT4	R	Undefined	H'FFB3

Note: * Lower 16 bits of the address.

Port 4 Register (PORT4): The pin states are always read when a port 4 read is performed.

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins P47 to P40.

10.4.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7) and D/A converter analog output pins (DA0 and DA1).

10.5 Port 7

10.5.1 Overview

Port 7 is an 8-bit I/O port. Port 7 is a multipurpose port for the 8-bit timer I/O pins (TMRI01, TMCI01, TMRI23, TMCI23, TMO0, TMO1, TMO2, TMO3), DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), IIC input pin (SYNCI), SCI I/O pins (SCK3, RxD3, TxD3) and manual reset input pin ($\overline{\text{MRES}}$). The pin functions for P77 to P74 are the same in all operating modes. P73 to P70 pin functions are switched according to operating mode.

Figure 10-4 shows the configuration for port 7 pins.

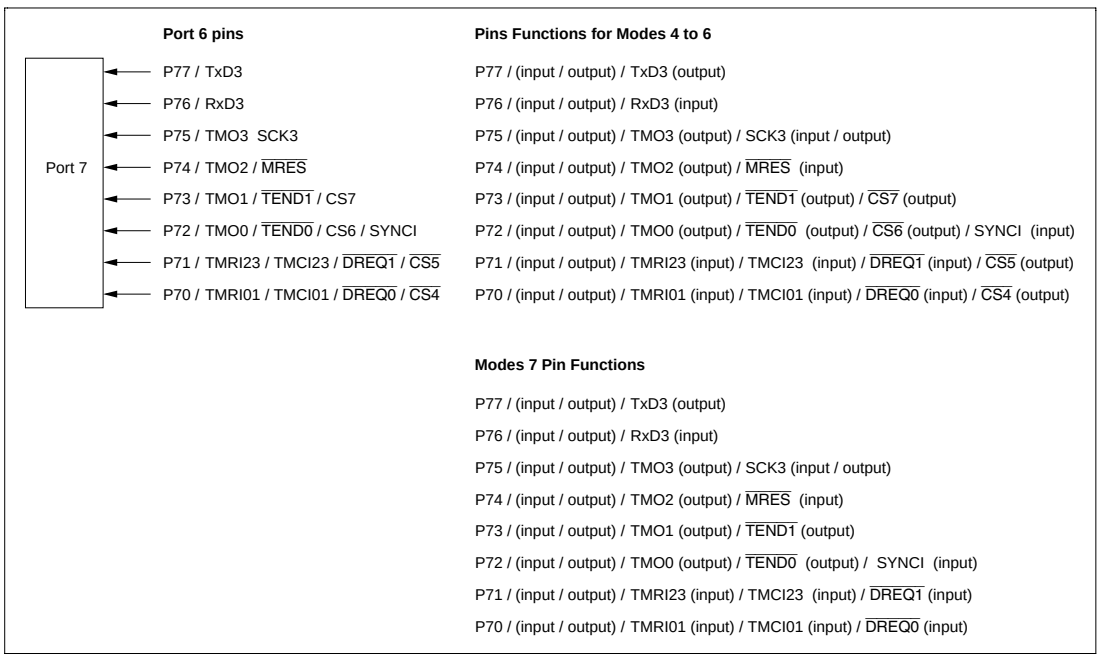


Figure 10-4 Port 6 Pin Functions

10.5.2 Register Configuration

Table 10-7 shows the port 7 register configuration.

Table 10-7 Port 7 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*
Port 7 data direction register	P7DDR	W	H'00	H'FE36
Port 7 data register	P7DR	R/W	H'00	H'FF06
Port 7 register	PORT7	R	Undefined	H'FFB6

Note: * Indicates the lower-place 16 bits of the address.

Port 7 Data Direction Register (P7DDR)

Bit	:	7	6	5	4	3	2	1	0
		P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P7DDR is an 8-bit write-dedicated register, which specifies the I/O for each port 7 pin by bit. Read is disenabled. If a read is carried out, undefined values are read out.

By setting P7DDR to 1, the corresponding port 7 pins become output, and by clearing to 0 they become input.

P7DDR is initialized to H'00 by a power-on reset and in hardware standby mode. The previous state is maintained by a manual reset and in software standby mode. DMAC, 8-bit timer and SCI are initialized by a manual reset, so the pin state is determined by the specification of P7DDR and P7DR.

Port 7 Data Register (P7DR)

Bit	:	7	6	5	4	3	2	1	0
		P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7DR is an 8-bit readable/writable register, which stores the output data of port 7 pins (P77 to P70).

P7DR is initialized to H'00 by a power-on reset and in hardware standby mode. The previous state is maintained by a manual reset and in software standby mode.

Port 7 Register (PORT7)

Bit	:	7	6	5	4	3	2	1	0
		P77	P76	P75	P74	P73	P72	P71	P70
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P77 to P70.

PORT7 is an 8-bit read-dedicated register, which reflects the state of pins. Write is disenabled. Always carry out writing off output data of port 7 pins (P77 to P70) to P7DR without fail.

When P7DDR is set to 1, if port 7 is read, the values of P7DR are read. When P7DDR is cleared to 0, if port 7 is read, the states of pins are read out.

P7DDR and P7DR are initialized by a power-on reset and in hardware standby mode, so PORT7 is determined by the state of the pins. The previous state is maintained by a manual reset and in software standby mode.

10.5.3 Pin Functions

The pins of port 7 are multipurpose pins which function as 8-bit timer I/O pins, (TMRI01, TMCI01, TMRI23, TMCI23, TMO0, TMO1, TMO2, TMO3), DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), IIC input pin (SYNCI), SCI I/O pins (SCK3, RxD3, TxD3) and manual reset input pin ($\overline{\text{MRES}}$). Table 10-8 shows the functions of port 7 pins.

Table 10-8 Port 7 Pin Functions

<table>
<tr><th>Pin</th><th>Selection Method and Pin Functions</th></tr>
<tr><td>P77/TxD3</td><td>Switches as follows according to combinations of SCR TE bit of SCI3, and the P77DDR bit.
<table>
<tr><td>TE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>P77DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>P77 input pin</td><td>P77 output pin</td><td>TxD3 output pin</td></tr>
</table></td></tr>
<tr><td>P76/RxD3</td><td>Switches as follows according to combinations of SCR RE bit of SCI3 and the P76DDR bit.
<table>
<tr><td>RE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>P76DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>P76 input pin</td><td>P76 output pin</td><td>RxD3 I/O pin</td></tr>
</table></td></tr>
<tr><td>P75/TMO3/
SCK3</td><td>Switches as follows according to combinations of SMR C/$\overline{\text{A}}$ bit of SCI3, SCR CKE0 and CKE1 bits, TCSR3 OS3 to OS0 bits of the 8-bit timer, and the P75DDR bit.
<table>
<tr><td>OS3 to OS0</td><td colspan="5">All 0</td><td>Any is 1</td></tr>
<tr><td>CKE1</td><td colspan="4">0</td><td>1</td><td>—</td></tr>
<tr><td>C/$\overline{\text{A}}$</td><td colspan="3">0</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>CKE0</td><td colspan="2">0</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>P75DDR</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>Pin function</td><td>P75
input pin</td><td>P75
output pin</td><td>SCK3
output pin</td><td>SCK3
output pin</td><td>SCK3
input pin</td><td>TMO3
Output</td></tr>
</table></td></tr>
</table>

Pin	Selection Method and Pin Functions

P74/TMO2/$\overline{\text{MRES}}$

Switches as follows according to combinations of TCSR2 OS3 to OS0 bits of the 8-bit timer, SYSCR MRESE bit and the P74DDR bit.

MRESE	0			1
OS3 to OS0	All 0		Any is 1	—
P74DDR	0	1	—	—
Pin function	P74 input pin	P74 output pin	TMO2 output	$\overline{\text{MRES}}$ input pin

P73/TMO1/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$

Switches according to combinations of DMATCR TEE1 bit of DMAC, TCSR1 OS3 to OS0 bits of the 8-bit timer, and the P73DDR bit.

Operating Mode	Modes 4 to 6				Mode 7			
TEE1	0			1	0			1
OS3 to OS0	All 0		Any is 1	—	All 0		Any is 1	—
P73DDR	0	1	—	—	0	1	—	—
Pin function	P73 input pin	$\overline{\text{CS7}}$ output pin	TMO1 output	$\overline{\text{TEND1}}$ output	P73 input pin	P73 output pin	TMO1 output	$\overline{\text{TEND1}}$ output

P72/TMO0/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$/SYNCI

Switches as follows according to combinations of operating mode and DMATCR TEE0 bit of DMAC, OS3 to OS0 bits of 8-bit timer TCSR0, and the P72DDR bit.

Operating Mode	Modes 4 to 6				Mode 7			
TEE0	0			1	0			1
OS3 to OS0	All 0		Any is 1	—	All 0		Any is 1	—
P72DDR	0	1	—	—	0	1	—	—
Pin function	P72 input pin	$\overline{\text{CS6}}$ output pin	TMO0 output	$\overline{\text{TEND1}}$ output	P72 input pin	P72 output pin	TMO0 output	$\overline{\text{TEND1}}$ output
	SYNCI input	—	SYNCI input					

Pin	Selection Method and Pin Functions

P71/TMRI23/ TMCI23/ $\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$

Switches as follows according to operating mode and P71DDR.

Operating Mode	Modes 4 to 6		Mode 7	
P71DDR	0	1	0	1
Pin function	P71 input Pin	$\overline{\text{CS5}}$ output	P71 input pin	P71 output pin
	$\overline{\text{DREQ0}}$, TMRI23, TMCI23 input	—	$\overline{\text{DREQ0}}$, TMRI23, TMCI23 input	

P70/TMRI01/ TMCI01/ $\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$

Switches as follows according to operating mode and P70DDR.

Operating Mode	Modes 4 to 6		Mode 7	
P70DDR	0	1	0	1
Pin function	P70 input pin	$\overline{\text{CS4}}$ output	P70 input pin	P70 output pin
	$\overline{\text{DREQ0}}$, TMRI01, TMCI01 input	—	$\overline{\text{DREQ0}}$, TMRI01, TMCI01 input	

10.6 Port 9

10.6.1 Overview

Port 9 is an 8-bit input-only port. Port 9 pins also function as A/D converter analog input pins (AN8 to AN15) and D/A converter analog output pins (DA2, DA3). Port 9 pin functions are the same in all operating modes. Figure 10-5 shows the port 9 pin configuration.

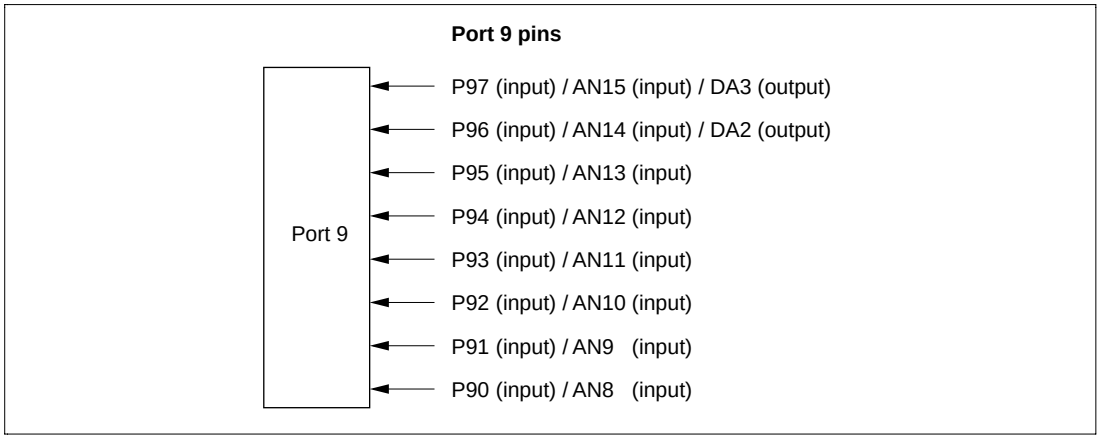


Figure 10-5 Port 9 Pin Functions

10.6.2 Register Configuration

Table 10-9 shows the port 9 register configuration. Port 9 is an input-only port, and does not have a data direction register or data register.

Table 10-9 Port 9 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 9 register	PORT9	R	Undefined	H'FFB8

Note: * Lower 16 bits of the address.

Port 9 Register (PORT9): The pin states are always read when a port 9 read is performed.

Bit	:	7	6	5	4	3	2	1	0
		P97	P96	P95	P94	P93	P92	P91	P90
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins P97 to P90.

10.6.3 Pin Functions

Port 9 pins are multipurpose pins which function as A/D converter analog input pins (AN8 to AN15) and D/A converter analog output pins (DA2, DA3).

10.7 Port A

10.7.1 Overview

Port A is a 6-bit I/O port. Port A pins also function as address bus outputs and SCI2 I/O pins (SCK2, RxD2, and TxD2). The pin functions change according to the operating mode.

Port A has a built-in MOS input pull-up function that can be controlled by software.

Figure 10-6 shows the port A pin configuration.

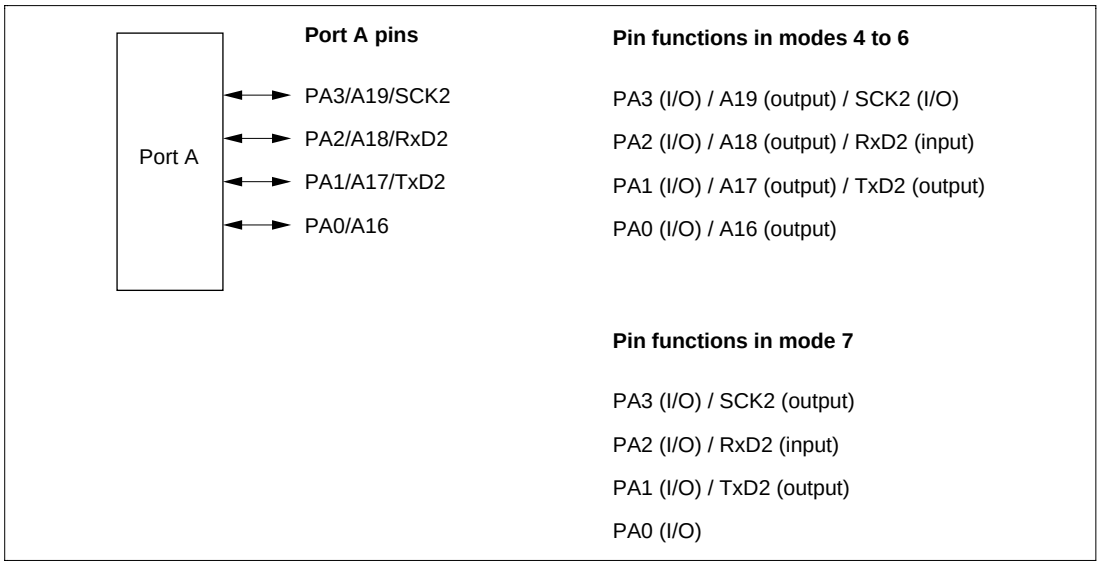


Figure 10-6 Port A Pin Functions

10.7.2 Register Configuration

Table 10-10 shows the port A register configuration.

Table 10-10 Port A Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port A data direction register	PADDR	W	H'0	H'FE39
Port A data register	PADR	R/W	H'0	H'FF09
Port A register	PORTA	R	Undefined	H'FFB9
Port A MOS pull-up control register	PAPCR	R/W	H'0	H'FE40
Port A open-drain control register	PAODR	R/W	H'0	H'FE47

Notes: *1 Lower 16 bits of the address.
*2 Value of bits 3 to 0.

Port A Data Direction Register (PADDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	W	W	W	W

PADDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bits 7 and 6 are reserved; they return an undetermined value if read.

PADDR is initialized to H'0 (bits 3 to 0) by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  The corresponding port A pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, irrespective of the value of bits PA4DDR to PA0DDR. When pins are not used as address outputs, setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

Port A Data Register (PADR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PADR is an 8-bit readable/writable register that stores output data for the port A pins (PA7 to PA0).

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

PADR is initialized to H'0 (bits 3 to 0) by a powr-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3	PA2	PA1	PA0
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	—	—	—	—	R	R	R	R

Note: * Determined by state of pins PA3 to PA0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port A pins (PA7 to PA0) must always be performed on PADR.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTA contents are determined by the pin states, as PADDR and PADR are initialized. PORTA retains its prior state by a manual reset or in software standby mode.

Port A MOS Pull-Up Control Register (PAPCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port A on an individual bit basis.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified. In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

PAPCR is initialized by a manual reset or to H'0 (bits 3 to 0) by a power-on reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port A Open Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off for each port A pin (PA7 to PA0).

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PAODR bit makes the corresponding port A pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

PAODR is initialized to H'0 (bits 3 to 0) by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

10.7.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, port A pins function as address outputs according to the setting of AE3 to AE0 in PFCR; when they do not function as address outputs, the pins function as SCI I/O pins and I/O ports.

Port A pin functions in modes 4 to 6 are shown in figure 10-7.

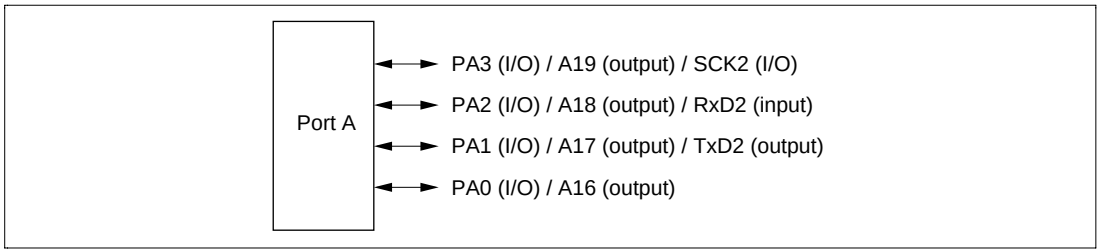


Figure 10-7 Port A Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port A pins function as I/O ports and SCI2 I/O pins (SCK2, TxD2, RxD2). Input or output can be specified for each pin on an individual bit basis. Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

Port A pin functions are shown in figure 10-8.

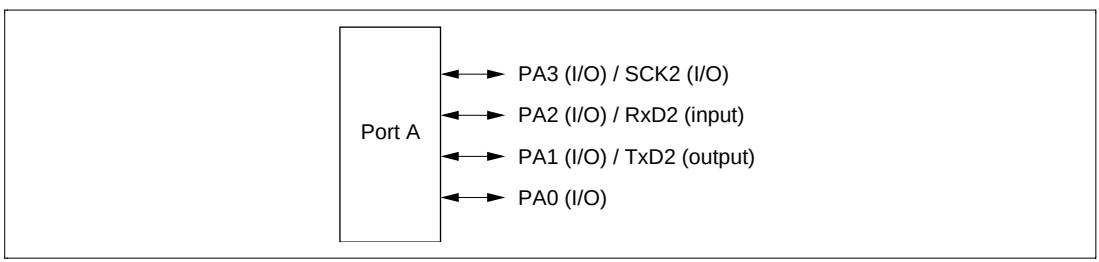


Figure 10-8 Port A Pin Functions (Mode 7)

10.7.4 MOS Input Pull-Up Function

Port A has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10-11 summarizes the MOS input pull-up states.

Table 10-11 MOS Input Pull-Up States (Port A)

Pin States	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output or SCI output	OFF	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF	ON/OFF

Legend:

OFF : MOS input pull-up is always off.

ON/OFF : On when PADDR = 0 and PAPCR = 1; otherwise off.

10.8 Port B

10.8.1 Overview

Port B is an 8-bit I/O port. Port B pins also function as TPU I/O pins (TIOCA3, TIOCB3, TIOCC3, TIOCD3, TIOCA4, TIOCB4, TIOCA5, TIOCB5) and as address outputs; the pin functions change according to the operating mode.

Port B has a built-in MOS input pull-up function that can be controlled by software.

Figure 10-9 shows the port B pin configuration.

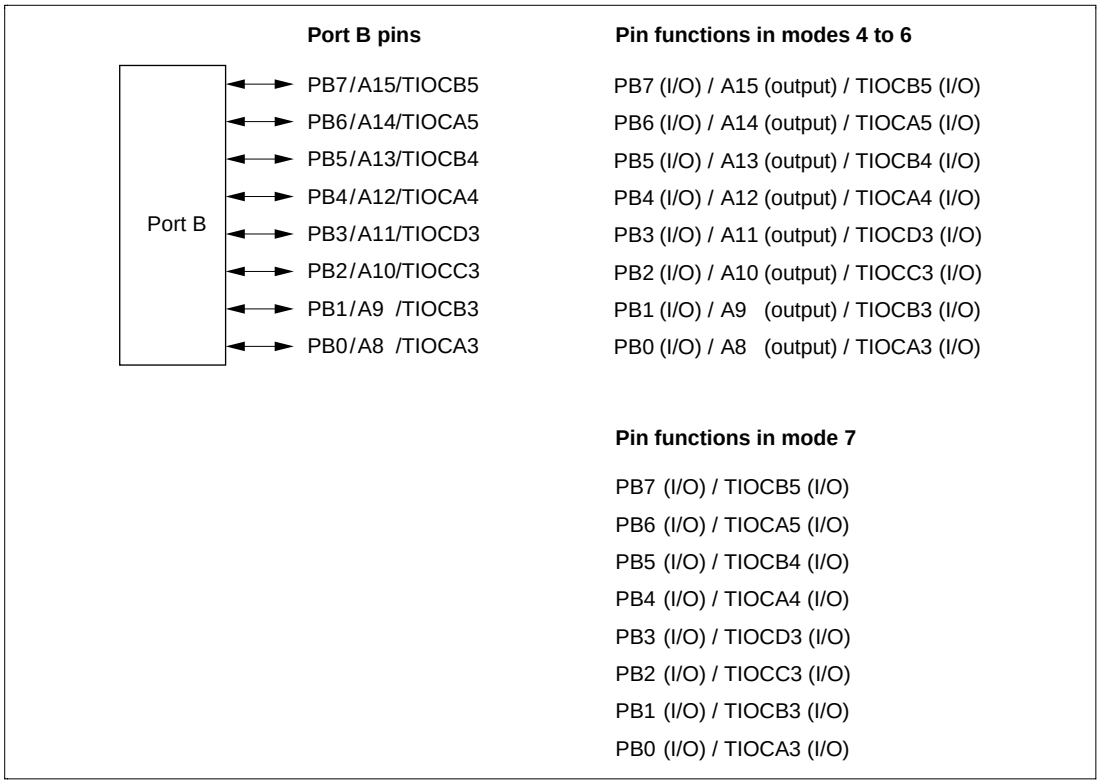


Figure 10-9 Port B Pin Functions

10.8.2 Register Configuration

Table 10-12 shows the port B register configuration.

Table 10-12 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port B data direction register	PBDDR	W	H'00	H'FE3A
Port B data register	PBDR	R/W	H'00	H'FF0A
Port B register	PORTB	R	Undefined	H'FFBA
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FE41
Port B open-drain control register	PBODR	R/W	H'00	H'FE48

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  The corresponding port B pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, irrespective of the value of the PBDDR bits. When pins are not used as address outputs, setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

Port B Data Register (PBDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBDR is an 8-bit readable/writable register that stores output data for the port B pins (PB7 to PB0). PBDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PB7 to PB0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port B pins (PB7 to PB0) must always be performed on PBDR.

If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTB contents are determined by the pin states, as PBDDR and PBDR are initialized. PORTB retains its prior state in software standby mode.

Port B MOS Pull-Up Control Register (PBPCR)

Bit	:	7	6	5	4	3	2	1	0
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port B on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the TPU's TIOR, and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIOR and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

PBPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port B Open Drain Control Register (PBODR)

Bit	:	7	6	5	4	3	2	1	0
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBODR is an 8-bit readable/writable register that controls the PMOS on/off state for each port B pin (PB7 to PB0).

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PBODR bit makes the corresponding port B pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

PBODR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

10.8.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, the corresponding port B pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR. When pins are not used as address outputs, they function as TPU I/O pins and I/O ports.

Port B pin functions in modes 4 to 6 are shown in figure 10-10.

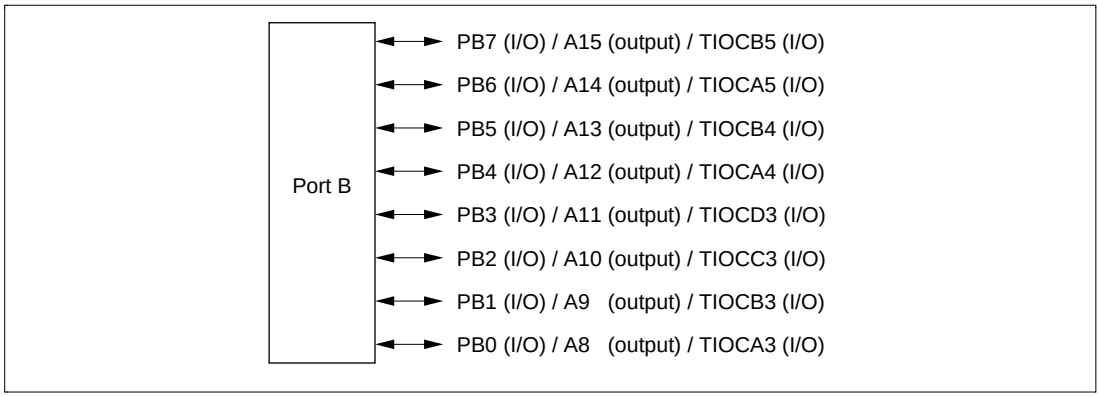


Figure 10-10 Port B Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port B pins function as I/O ports and TPU I/O pins (TIOCA3, TIOCB3, TIOCC3, TIOCD3, TIOCA4, TIOCB4, TIOCA5, and TIOCB5). Input or output can be specified for each pin on an individual bit basis. Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

Port B pin functions in mode 7 are shown in figure 10-11.

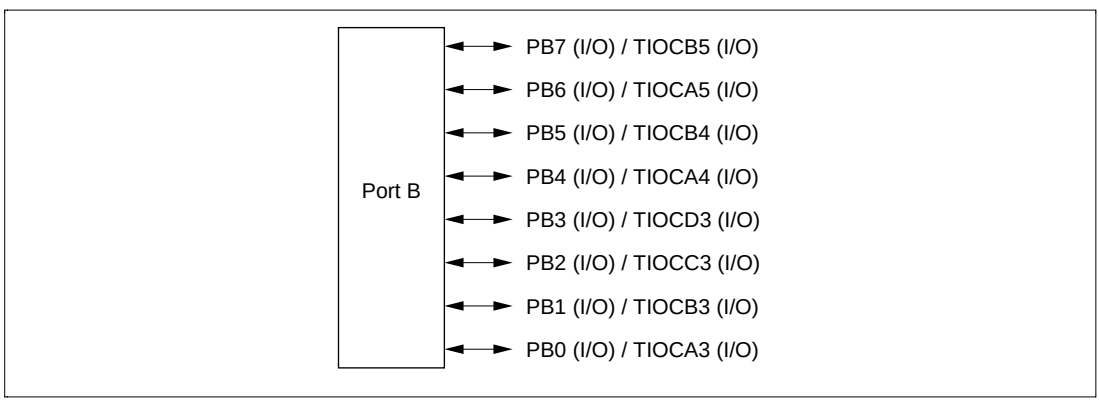


Figure 10-11 Port B Pin Functions (Mode 7)

10.8.4 MOS Input Pull-Up Function

Port B has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the TPU's TIOR, and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIOR and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10-13 summarizes the MOS input pull-up states.

Table 10-13 MOS Input Pull-Up States (Port B)

Pin States	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output or TPU output	OFF	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF	ON/OFF

Legend:

OFF : MOS input pull-up is always off.

ON/OFF : On when PBDDR = 0 and PBPCR = 1; otherwise off.

10.9 Port C

10.9.1 Overview

Port C is an 8-bit I/O port. Port C has a 14-bit PWM output (PWM0, PWM1) and an address bus output function. The pin functions change according to the operating mode.

Port C has a built-in MOS input pull-up function that can be controlled by software.

Figure 10-12 shows the port C pin configuration.

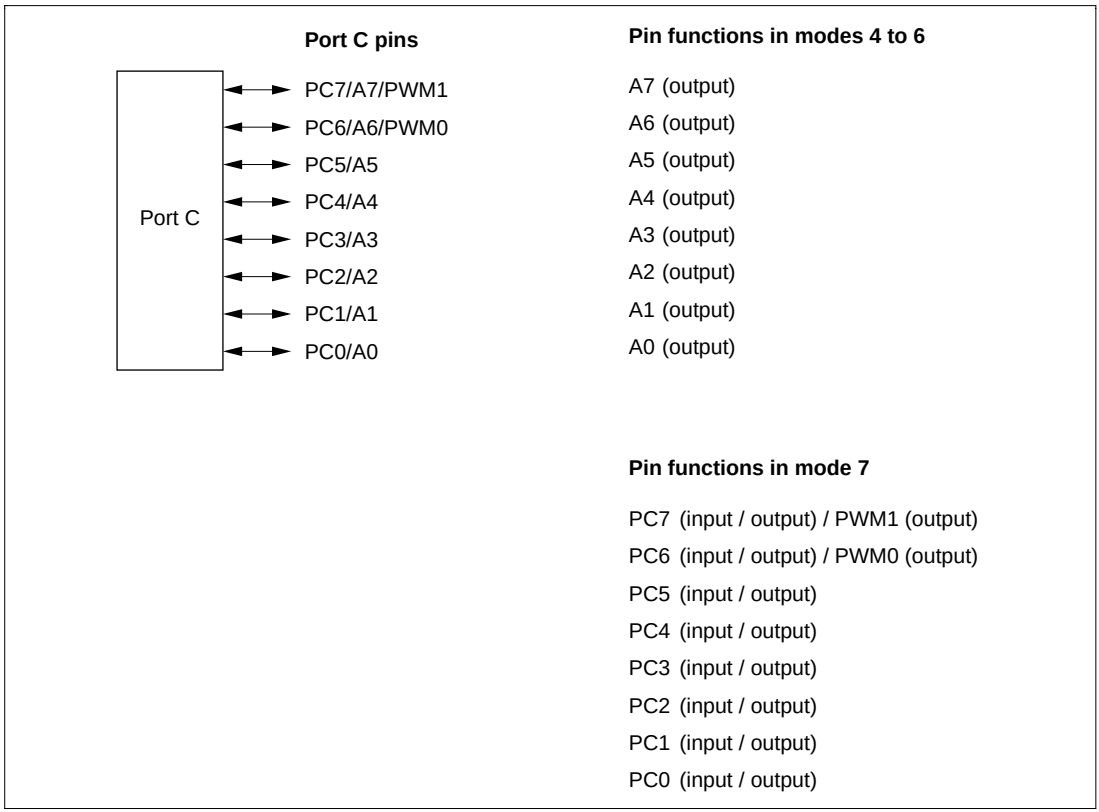


Figure 10-12 Port C Pin Functions

10.9.2 Register Configuration

Table 10-14 shows the port C register configuration.

Table 10-14 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port C data direction register	PCDDR	W	H'00	H'FE3B
Port C data register	PCDR	R/W	H'00	H'FF0B
Port C register	PORTC	R	Undefined	H'FFBB
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'FE42
Port C open-drain control register	PCODR	R/W	H'00	H'FE49

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

PCDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when the mode is changed to software standby mode.

- Modes 4 and 5
  The corresponding port C pins are address outputs irrespective of the value of the PCDDR bits.
- Mode 6
  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while clearing the bit to 0 makes the pin an input port.
- Mode 7
  Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

Port C Data Register (PCDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDR is an 8-bit readable/writable register that stores output data for the port C pins (PC7 to PC0).

PCDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PC7 to PC0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port C pins (PC7 to PC0) must always be performed on PCDR.

If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTC contents are determined by the pin states, as PCDDR and PCDR are initialized. PORTC retains its prior state by a manual reset or in software standby mode.

Port C MOS Pull-Up Control Register (PCPCR)

Bit	:	7	6	5	4	3	2	1	0
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port C on an individual bit basis.

In modes 6 and 7, if PCPCR is set to 1 when the port is in the input state in accordance with the settings of DACR and PCDDR in PWM, the MOS input pull-up is set to ON.

PCPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port C Open Drain Control Register (PCODR)

Bit	:	7	6	5	4	3	2	1	0
		PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDDR is an 8-bit Read/Write register and controls PMOS On/Off of each pin (PC7 to PC0) of port C.

If PCODR is set to 1 by setting AE3 to AE0 in PFCR in mode other than address output mode, port C pins function as NMOS open drain outputs and when the setting is cleared to 0, the pins function as CMOS outputs.

PCODR is initialized to H'00 in power-on reset mode or hardware standby mode. PCODR retains the last state in manual reset mode or software standby mode.

10.9.3 Pin Functions for Each Mode

(1) Modes 4 and 5

In mode 4 and 5, port C pins function as address outputs automatically.

Figure 10-13 shows the port C pin functions.

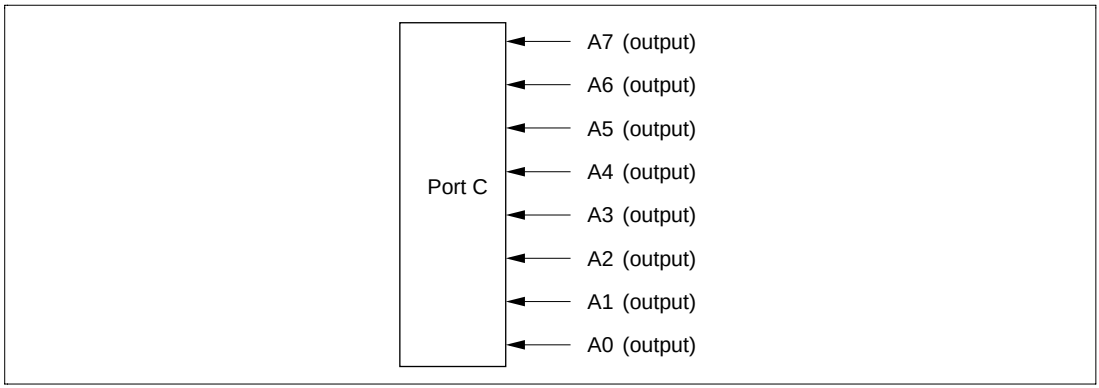


Figure 10-13 Port C Pin Functions (Modes 4 and 5)

(2) Mode 6

In mode 6, port C pints function as address outputs or input ports and I/O can be specified in bit units. When each bit in PCDDR is set to 1, the corresponding pin functions as an address output and when the bit cleared to 0, the pin functions as a PWM output and an input port.

Figure 10-14 shows the port C pin functions.

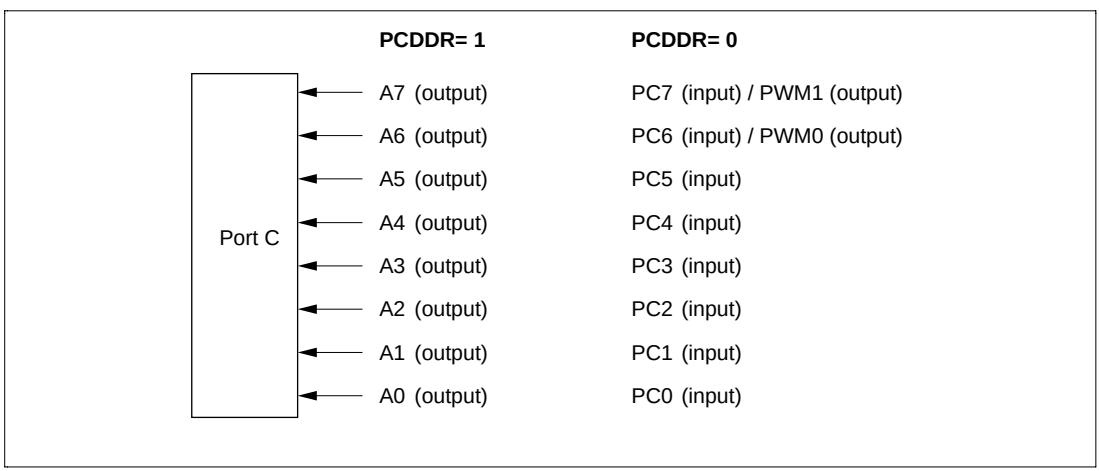


Figure 10-14 Port C Pin Functions (Mode 6)

(3) Mode 7

In mode 7, port C pins function as PWM outputs and I/O ports and I/O can be specified for each pin in bit units. When each bit in PCDDR is set to 1, the corresponding pin functions as an output port and when the bit is cleared to 0, the pin functions as an input port.

Figure 10-15 shows the port C pin functions.

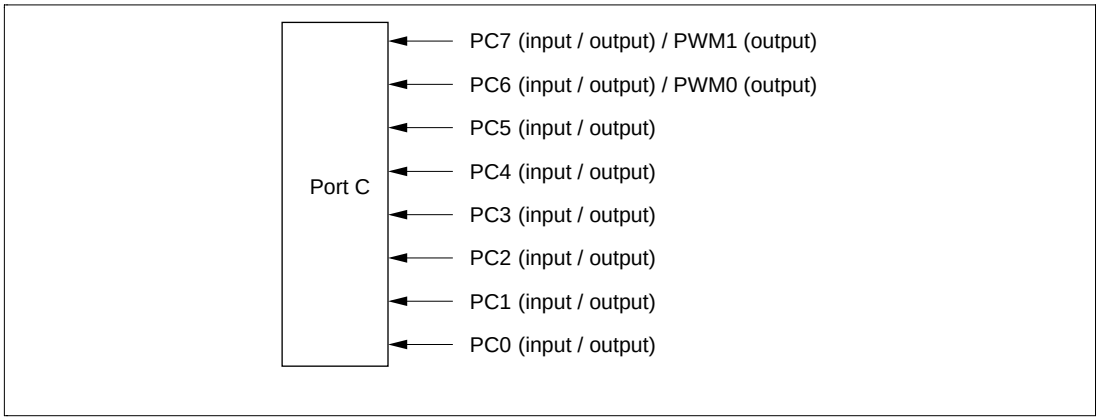


Figure 10-15 Port C Pin Functions (Mode 7)

10.9.4 MOS Input Pull-Up Function

Port C has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 6 and 7, and can be specified as on or off on an individual bit basis.

In modes 6 and 7, when PCPCR is set to 1 in the input state by setting of DACR and PCDDR, the MOS input pull-up is set to ON.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10-15 summarizes the MOS input pull-up states.

Table 10-15 MOS Input Pull-Up States (Port C)

Pin States	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output or PWM output	OFF	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF	ON/OFF

Legend:

OFF : MOS input pull-up is always off.

ON/OFF : On when PCDDR = 0 and PCPCR = 1; otherwise off.

10.10 Port D

10.10.1 Overview

Port D is an 8-bit I/O port. Port D has a data bus I/O function, and the pin functions change according to the operating mode.

Port D has a built-in MOS input pull-up function that can be controlled by software.

Figure 10-16 shows the port D pin configuration.

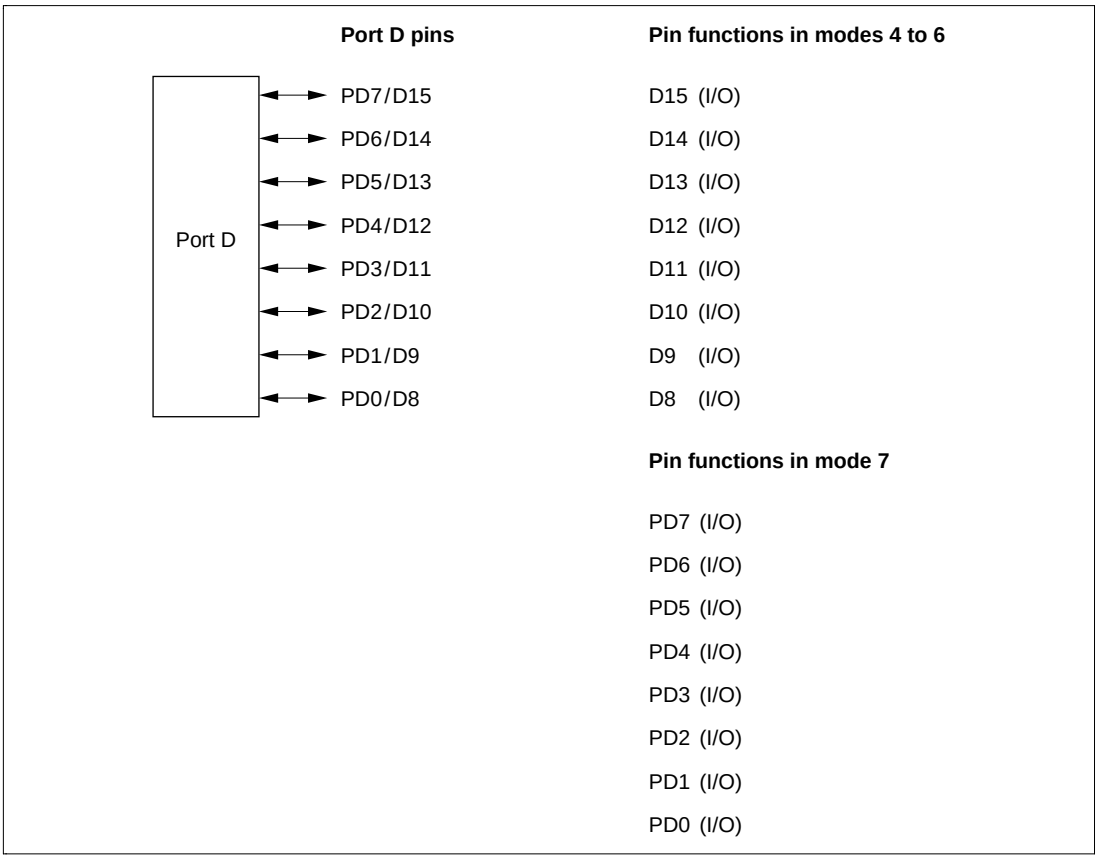


Figure 10-16 Port D Pin Functions

10.10.2 Register Configuration

Table 10-16 shows the port D register configuration.

Table 10-16 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FE3C
Port D data register	PDDR	R/W	H'00	H'FF0C
Port D register	PORTD	R	Undefined	H'FFBC
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FE43

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

PDDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

- Modes 4 to 6

  The input/output direction specification by PDDDR is ignored, and port D is automatically designated for data I/O.

- Mode 7

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Port D Data Register (PDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDDR is an 8-bit readable/writable register that stores output data for the port D pins (PD7 to PD0).

PDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1	0
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PD7 to PD0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port D pins (PD7 to PD0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTD contents are determined by the pin states, as PDDDR and PDDR are initialized. PORTD retains its prior state by a manual reset or in software standby mode.

Port D MOS Pull-Up Control Register (PDPCR)

Bit	:	7	6	5	4	3	2	1	0
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

10.10.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, port D pins are automatically designated as data I/O pins.

Port D pin functions in modes 4 to 6 are shown in figure 10-17.

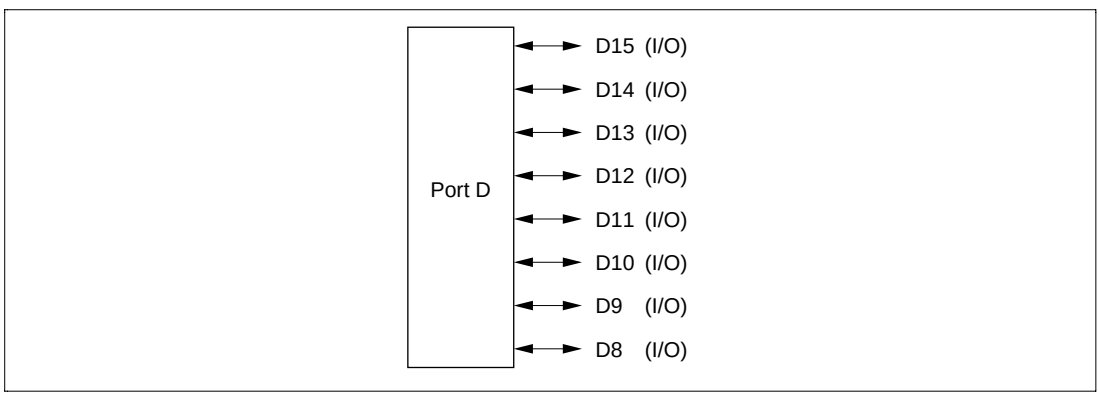


Figure 10-17 Port D Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port D pins function as I/O ports. Input or output can be specified for each pin on an individual bit basis. Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Port D pin functions in mode 7 are shown in figure 10-18.

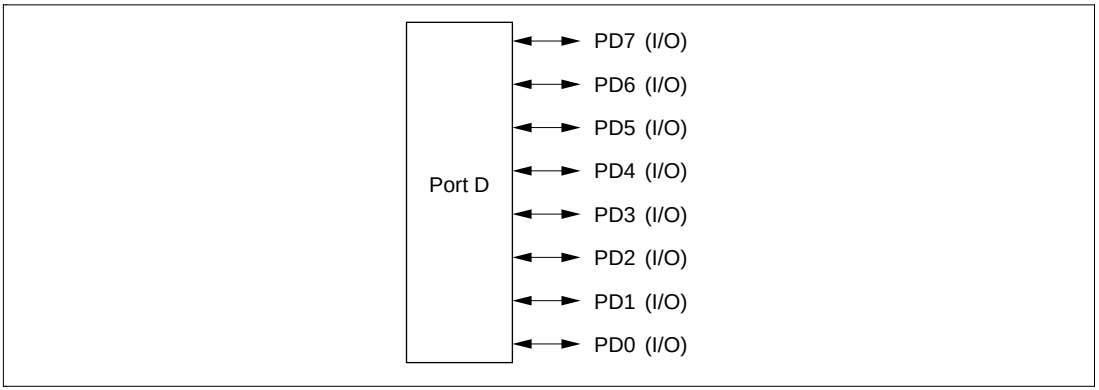


Figure 10-18 Port D Pin Functions (Mode 7)

10.10.4 MOS Input Pull-Up Function

Port D has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in mode 7, and can be specified as on or off on an individual bit basis.

When a PDDDR bit is cleared to 0 in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10-17 summarizes the MOS input pull-up states.

Table 10-17 MOS Input Pull-Up States (Port D)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
4 to 6	OFF	OFF	OFF	OFF	OFF
7			ON/OFF	ON/OFF	ON/OFF

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PDDDR = 0 and PDPCR = 1; otherwise off.

10.11 Port E

10.11.1 Overview

Port E is an 8-bit I/O port. Port E has a data bus I/O function, and the pin functions change according to the operating mode and whether 8-bit or 16-bit bus mode is selected.

Port E has a built-in MOS input pull-up function that can be controlled by software.

Figure 10-19 shows the port E pin configuration.

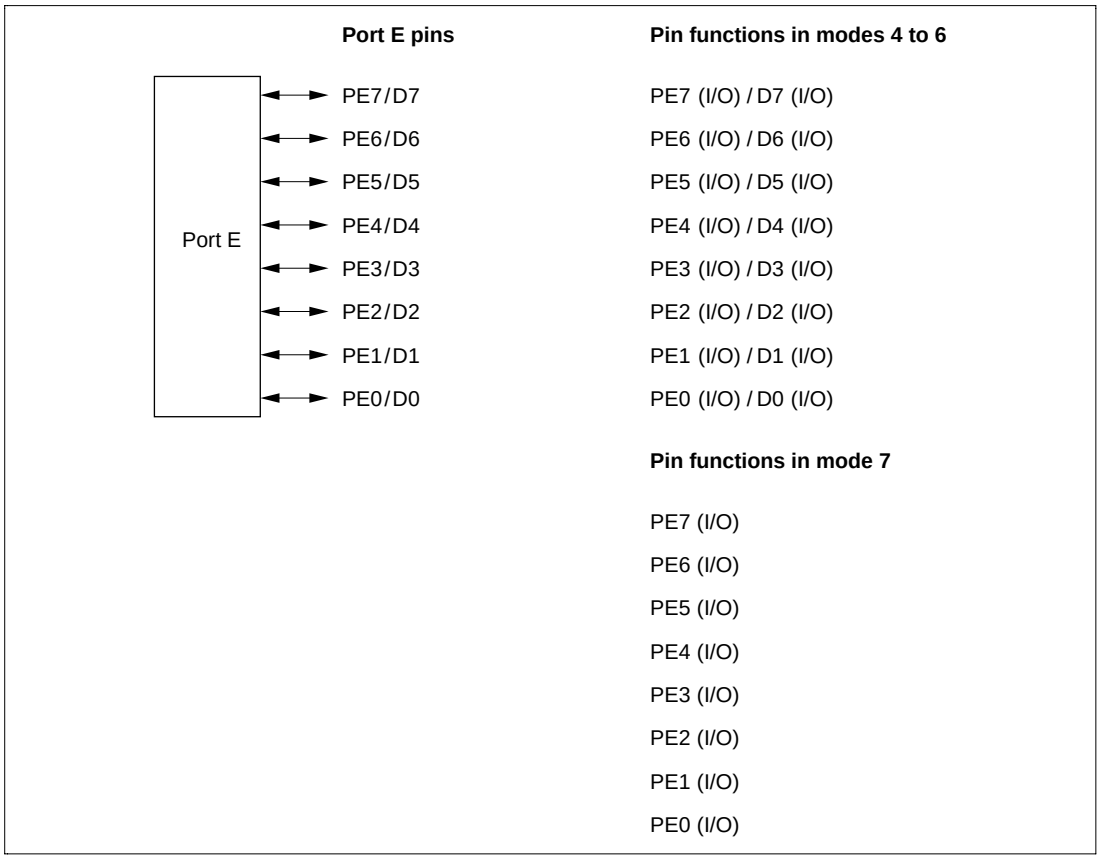


Figure 10-19 Port E Pin Functions

10.11.2 Register Configuration

Table 10-18 shows the port E register configuration.

Table 10-18 Port E Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port E data direction register	PEDDR	W	H'00	H'FE3D
Port E data register	PEDR	R/W	H'00	H'FF0D
Port E register	PORTE	R	Undefined	H'FFBD
Port E MOS pull-up control register	PEPCR	R/W	H'00	H'FE44

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

PEDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

- Modes 4 to 6

  When 8-bit bus mode has been selected, port E pins function as I/O ports. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 7, Bus Controller.

- Mode 7

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E Data Register (PEDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEDR is an 8-bit readable/writable register that stores output data for the port E pins (PE7 to PE0).

PEDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PE7 to PE0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port E pins (PE7 to PE0) must always be performed on PEDR.

If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTE contents are determined by the pin states, as PEDDR and PEDR are initialized. PORTE retains its prior state by a manual reset or in software standby mode.

Port E MOS Pull-Up Control Register (PEPCR)

Bit	:	7	6	5	4	3	2	1	0
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port E on an individual bit basis.

When a PEDDR bit is cleared to 0 (input port setting) with 8-bit bus mode selected in mode 4, 5, or 6, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PEPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

10.11.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, when 8-bit access is designated and 8-bit bus mode is selected, port E pins are automatically designated as I/O ports. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

When 16-bit bus mode is selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

Port E pin functions in modes 4 to 6 are shown in figure 10-20.

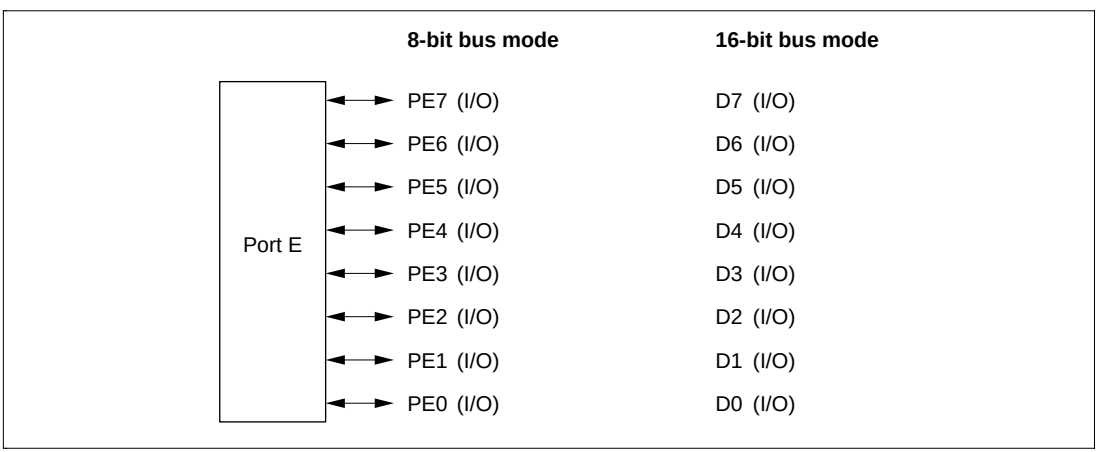


Figure 10-20 Port E Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port E pins function as I/O ports. Input or output can be specified for each pin on a bit-by-bit basis. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E pin functions in mode 7 are shown in figure 10-21.

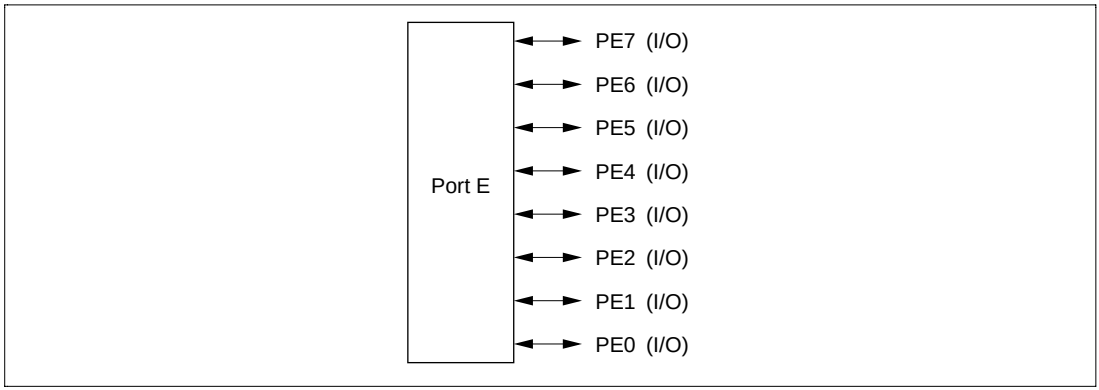


Figure 10-21 Port E Pin Functions (Mode 7)

10.11.4 MOS Input Pull-Up Function

Port E has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 4 to 6 when 8-bit bus mode is selected, or in mode 7, and can be specified as on or off on an individual bit basis.

When a PEDDR bit is cleared to 0 in mode 4 to 6 when 8-bit bus mode is selected, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10-19 summarizes the MOS input pull-up states.

Table 10-19 MOS Input Pull-Up States (Port E)

Modes		Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
7		OFF	OFF	ON/OFF	ON/OFF	ON/OFF
4 to 6	8-bit bus					
	16-bit bus			OFF	OFF	OFF

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PEDDR = 0 and PEPCR = 1; otherwise off.

10.12 Port F

10.12.1 Overview

Port F is an 8-bit I/O port. Port F pins also function as external interrupt input pins ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ3}}$), BUZZ output pin, A/D trigger input pin ($\overline{\text{ADTRG}}$), bus control signal input/output pins ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, $\overline{\text{LCAS}}$, $\overline{\text{WAIT}}$, $\overline{\text{BREQO}}$, $\overline{\text{BREQ}}$, and $\overline{\text{BACK}}$) and the system clock (ø) output pin.

Figure 10-22 shows the port F pin configuration.

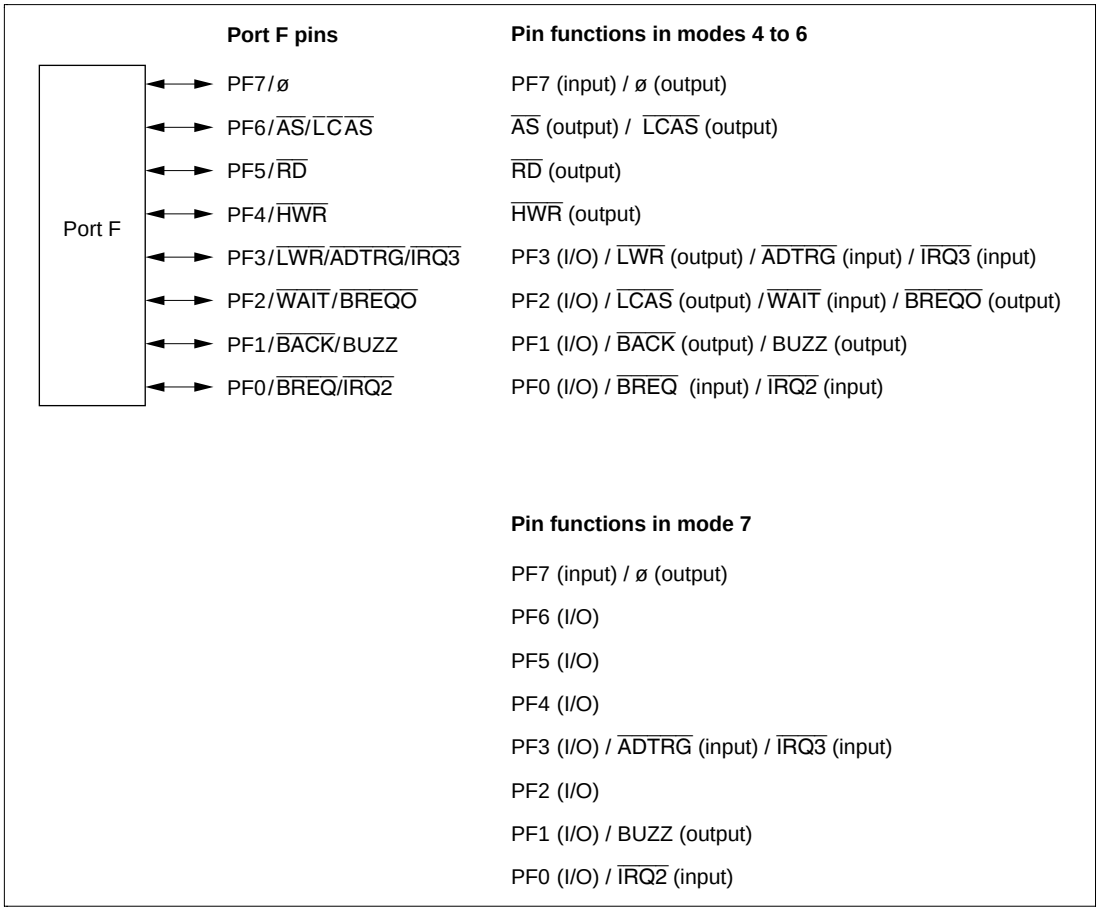


Figure 10-22 Port F Pin Functions

10.12.2 Register Configuration

Table 10-20 shows the port F register configuration.

Table 10-20 Port F Registers

Name	Abbreviation	R/W	Initial Value	Address*1
Port F data direction register	PFDDR	W	H'80/H'00*2	H'FE3E
Port F data register	PFDR	R/W	H'00	H'FF0E
Port F register	PORTF	R	Undefined	H'FFBE

Notes: *1 Lower 16 bits of the address.
*2 Initial value depends on the mode.

Port F Data Direction Register (PFDDR)

Bit :	7	6	5	4	3	2	1	0
	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Modes 4 to 6								
Initial value :	1	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W
Mode 7								
Initial value :	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

PFDDR is initialized by a power-on reset, and in hardware standby mode, to H'80 in modes 4 to 6, and to H'00 in mode 7. It retains its prior state by a manual reset or in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  Pin PF7 functions as the ø output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0.

  The input/output direction specified by PFDDR is ignored for pins PF6 to PF3, which are automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$). PF6 functions as a bus control output ($\overline{LCAS}$) by setting of the bus controller.

Pins PF2 to PF0 are designated as bus control input/output pins ($\overline{\text{LCAS}}$, $\overline{\text{WAIT}}$, $\overline{\text{BREQO}}$, $\overline{\text{BACK}}$, $\overline{\text{BREQ}}$) by means of bus controller settings. At other times, setting a PFDDR bit to 1 makes the corresponding port F pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PFDDR bit to 1 makes the corresponding port F pin PF6 to PF0 an output port, or in the case of pin PF7, the ø output pin. Clearing the bit to 0 makes the pin an input port.

Port F Data Register (PFDR)

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (PF7 to PF0).

PFDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PF7 to PF0.

PORTF is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port F pins (PF7 to PF0) must always be performed on PFDR.

If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTF contents are determined by the pin states, as PFDDR and PFDR are initialized. PORTF retains its prior state by a manual reset or in software standby mode.

10.12.3 Pin Functions

Port F pins also function as external interrupt input pins ($\overline{IRQ2}$ and $\overline{IRQ3}$), BUZZ output pin, A/D trigger input pin ($\overline{ADTRG}$), bus control signal input/output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{LCAS}$, $\overline{WAIT}$, $\overline{BREQO}$, $\overline{BREQ}$, and $\overline{BACK}$) and the system clock (ø) output pin. The pin functions differ between modes 4 to 6, and mode 7. Port F pin functions are shown in table 10-21.

Table 10-21 Port F Pin Functions

PF7/ø

The pin function is switched as shown below according to bit PF7DDR.

PF7DDR	0	1
Pin function	PF7 input pin	ø output pin

PF6/$\overline{AS}$/$\overline{LCAS}$

The pin function is switched as shown below according to the combination of the operating mode and bits RMTS2 to RMTS0, LCASS, BREQOE, WAITE, ABW5 to ABW2, and PF2DDR.

Operating Mode	Modes 4 to 6		Mode 7	
LCASS	0	1*	—	
PF6DDR	—	—	0	1
Pin function	$\overline{AS}$ output pin	$\overline{LCAS}$ output pin	PF6 input pin	PF6 output pin

Note: * Restricted to RMTS2 to TMTS0=B'001 to B'011, DRAM space 16-bit access in modes 4 to 6 only

PF5/$\overline{RD}$

The pin function is switched as shown below according to the operating mode and bit PF5DDR.

Operating Mode	Modes 4 to 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output pin	PF5 input pin	PF5 output pin

PF4/$\overline{HWR}$

The pin function is switched as shown below according to the operating mode and bit PF4DDR.

Operating Mode	Modes 4 to 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output pin	PF4 input pin	PF4 output pin

Pin	Selection Method and Pin Functions

PF3/$\overline{\text{LWR}}$/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$

The pin function is switched as shown below according to the operating mode, the bus mode, A/D converter bits TRGS1 and TRGS0, and bit PF3DDR.

Operating mode	Modes 4 to 6			Mode 7	
Bus mode	16-bit bus mode	8-bit bus mode		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{\text{LWR}}$ output pin	PF3 input pin	PF3 output pin	PF3 input pin	PF3 output pin
		$\overline{\text{ADTRG}}$ input pin*1			
		$\overline{\text{IRQ3}}$ input pin*2			

Notes: *1 $\overline{\text{ADTRG}}$ input when TRGS0 = TRGS1 = 1.

*2 When used as an external interrupt input pin, do not use as an I/O pin for another function.

PF2/$\overline{\text{LCAS}}$/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$

The pin function is switched as shown below according to the combination of the operating mode and bits RMTS2 to RMTS0, LCASS, BREQOE, WAITE, ABW5 to ABW2, and PF2DDR.

Operating Mode	Modes 4 to 6					Mode 7	
LCASS	0*	1				—	
BREQOE	—	0			1	—	
WAITE	—	0		1	—	—	
PF2DDR	—	0	1	—	—	0	1
Pin function	$\overline{\text{LCAS}}$ output pin	PF2 input pin	PF2 output pin	$\overline{\text{WAIT}}$ input pin	$\overline{\text{BREQO}}$ output pin	PF2 input pin	PF2 output pin

Note: * Restricted to RMTS2 to TMTS0=B'001 to B'011, DRAM space 16-bit access in modes 4 to 6 only.

PF1/$\overline{\text{BACK}}$/BUZZ

The pin function is switched as shown below according to the combination of the operating mode and bits BRLE, BUZZE, and PF1DDR.

Operating Mode	Modes 4 to 6				Mode 7		
BRLE	0			1	—		
BUZZE	0		1	—	0		1
PF1DDR	0	1	—	—	0	1	—
Pin function	PF1 input pin	PF1 output pin	BUZZ output pin	$\overline{\text{BACK}}$ output pin	PF1 input pin	PF1 output pin	BUZZ output pin

Pin	Selection Method and Pin Functions
PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	The pin function is switched as shown below according to the combination of the operating mode, and bits BRLE and PF0DDR.

Operating Mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
Pin function	PF0 input pin	PF0 output pin	$\overline{\text{BREQ}}$ input pin	PF0 input pin	PF0 output pin
	$\overline{\text{IRQ2}}$ input pin				

10.13 Port G

10.13.1 Overview

Port G is a 5-bit I/O port and also used as external interrupt input pins ($\overline{\text{IRQ6}}$, $\overline{\text{IRQ7}}$) and bus control signal output pins ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{CAS}}$, $\overline{\text{OE}}$).

Figure 10-23 shows the configuration of port G pins.

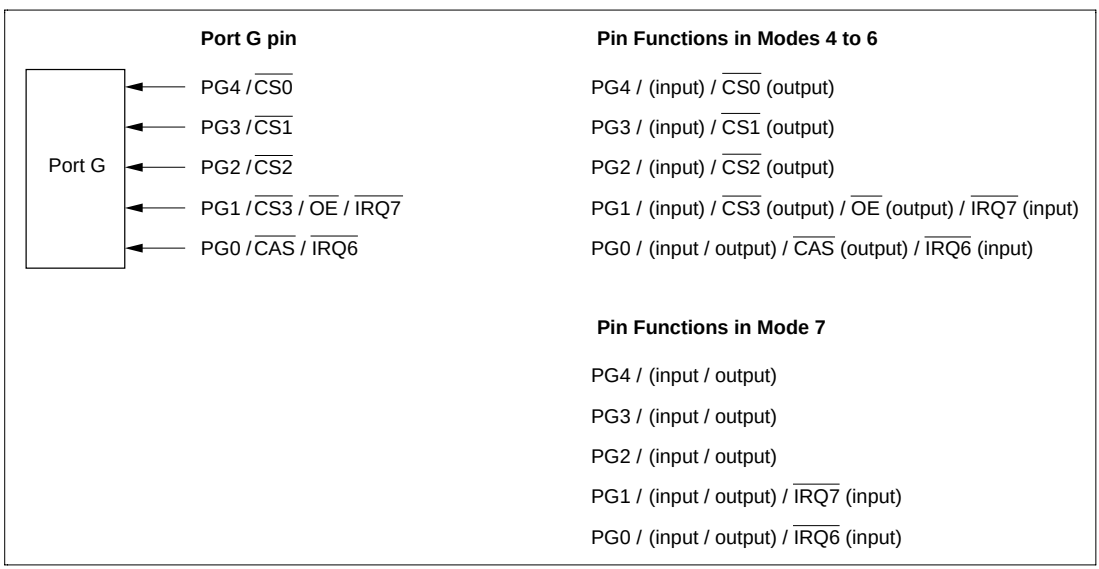


Figure 10-23 Port G Pin Functions

10.13.2 Register Configuration

Table 10-22 shows the port G register configuration.

Table 10-22 Port G Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port G data direction register	PGDDR	W	H'10/H'00*3	H'FE3F
Port G data register	PGDR	RW	H'00	H'FF0F
Port G register	PORTG	R	Undefined	H'FFBF

Notes: *1 Indicates the low order 16 bits of the address
*2 Value of bits 4 to 0
*3 The initial value varies according to the mode.

Port G Data Direction Register (PGDDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Modes 4 and 5									
Initial value	:	Undefined	Undefined	Undefined	1	0	0	0	0
R/W	:	—	—	—	W	W	W	W	W
Modes 6 and 7									
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	W	W	W	W	W

PGDDR is an 8-bit write only register and specifies I/O of each pin of port G in bit units. Read processing is invalid. Bits 7 to 5 are reserved bits. When the contents are read, undefined values are read.

In modes 4 and 5, the PG4DDR bits are initialized to H'10 (bit 4 to 0) in power-on reset or hardware standby mode, in modes 6 and 7, the bits are initialized to H'00 (bit 4 to 0). In manual reset or software standby mode, PGDDR retains the last status. Use the OPE bit of SBYCR to select whether the bus control output pin retains the output state or becomes the high-impedance when the mode is changed to a software standby mode.

- Modes 4 to 6

  When PGDDR is set to 1, pins PG4 to PG1 function as bus control signal output pins ($\overline{CS0}$ to $\overline{CS3}$, $\overline{OE}$). When PGDDR is cleared to 0, the pins function as input ports.

  When the DRAM interface is set, pin PG0 functions as the $\overline{CAS}$ output pin. When PGDDR is set to 1, the pin functions as an output port. When PGDDR is cleared to 0, the pin functions as an input port.

See Chapter 7 for the DRAM interface.

- Mode 7

PGDDR to 1 it becomes an output port, and by clearing it to 0 it becomes an input port.

Port G Data Register (PGDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W	R/W

PGDR is an 8-bit read/write register and stores output data of port G output pins (PG4 to PG0).

Bits 7 to 5 are reserved bits. When the contents are read, undefined values are read. Write processing is invalid.

In power-on reset or hardware standby mode, PGDR is initialized to H'00 (bits 4 to 0). In manual reset or software standby mode, PGDR retains the last state.

(3) Port G Register (PORTG)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R	R

Note: * Determined by the state of PG4 to PG0

PORTG is an 8-bit read only register and reflects the pin state. Write processing is invalid. Write processing of output data of port G pins (PG4 to PG0) must be performed for PGDR.

Bits 7 to 5 are reserved bits. When the contents are read, undefined values are read. Write processing is invalid.

If port G is read when PGDDR is set to 1, the value in PGDR is read. If port G is read when PGDDR is cleared to 0, the pin state is read.

In power-on reset or hardware standby mode, port G is determined by the pin state because PGDDR and PGDR are initialized. In manual reset or software standby mode, the last state is retained.

10.13.3 Pin Functions

Port G is used also as external interrupt input pins ($\overline{IRQ6}$, $\overline{IRQ7}$) and bus control signal output pins ($\overline{CS0}$ to $\overline{CS3}$, $\overline{CAS}$, $\overline{OE}$). The pin functions are different between modes 4 and 6, and mode 7. Table 10-23 shows the port G pin functions.

Table 10-23 Port G Pin Functions

Pin | **Selection Method and Pin Functions**

PG4/$\overline{CS0}$

The pin function is switched as shown below according to the operating mode and bit PG4DDR.

Operating Mode	Modes 4 to 6		Mode 7	
PG4DDR	0	1	0	1
Pin function	PG4 input pin	$\overline{CS0}$ output pin	PG4 input pin	PG4 output pin

PG3/$\overline{CS1}$

The pin function is switched as shown below according to the operating mode and bit PG3DDR.

Operating Mode	Modes 4 to 6		Mode 7	
PG3DDR	0	1	0	1
Pin function	PG3 input pin	$\overline{CS1}$ output pin	PG3 input pin	PG3 output pin

PG2/$\overline{CS2}$

The pin function is switched as shown below according to the operating mode and bit PG2DDR.

Operating Mode	Modes 4 to 6		Mode 7	
PG2DDR	0	1	0	1
Pin function	PG2 input pin	$\overline{CS2}$ output pin	PG2 input pin	PG2 output pin

Pin	Selection Method and Pin Functions
PG1/$\overline{\text{CS3}}$/$\overline{\text{OE}}$/$\overline{\text{IRQ7}}$	The pin function is switched as shown below according to the operating mode and bits OES and PG1DDR in BCRL.

Operating Mode	Modes 4 to 6			Mode 7	
PG1DDR	0	1		0	1
OES	—	0	1	—	—
Pin function	PG1 input pin	$\overline{\text{CS3}}$ output pin	$\overline{\text{OE}}$ output pin	PG1 input pin	PG1 output pin
	$\overline{\text{IRQ7}}$ input				

PG0/$\overline{\text{CAS}}$/$\overline{\text{IRQ6}}$	The pin function is switched as shown below according to the operating mode and bits RMTS2 to RMTS0 in BCRH.

Operating Mode	Modes 4 to 6			Mode 7	
RMTS2 to RMTS0	B'000		B'001 to B'011	—	
PG0DDR	0	1	—	0	1
Pin function	PG0 input pin	PG0 output pin	$\overline{\text{CAS}}$ output pin	PG0 input pin	PG0 output pin
	$\overline{\text{IRQ6}}$ input				

Section 11 16-Bit Timer Pulse Unit (TPU)

11.1 Overview

The H8S/2633 Series has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels.

11.1.1 Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture possible
  - Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set
  - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 1 (channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

- 26 interrupt sources
  - For channels 0 and 3, four compare match/input capture dual-function interrupts and one overflow interrupt can be requested independently
  - For channels 1, 2, 4, and 5, two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently
- Automatic transfer of register data
  - Block transfer, 1-word data transfer, and 1-byte data transfer possible by data transfer controller (DTC) or DMA controller (DMAC)
- Programmable pulse generator (PPG) output trigger can be generated
  - Channel 0 to 3 compare match/input capture signals can be used as PPG output trigger
- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signals can be used as A/D converter conversion start trigger
- Module stop mode can be set
  - As the initial setting, TPU operation is halted. Register access is enabled by exiting module stop mode.

Table 11-1 lists the functions of the TPU.

Table 11-1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		ø/1 ø/4 ø/16 ø/64 TCLKA TCLKB TCLKC TCLKD	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKB	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKB TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 ø/1024 ø/4096 TCLKA	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKC TCLKD
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B	TGR5A TGR5B
General registers/ buffer registers		TGR0C TGR0D	—	—	TGR3C TGR3D	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DMAC activation	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
A/D converter trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
PPG trigger	TGR0A/ TGR0B compare match or input capture	TGR1A/ TGR1B compare match or input capture	TGR2A/ TGR2B compare match or input capture	TGR3A/ TGR3B compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend

◯ : Possible

— : Not possible

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the TPU.

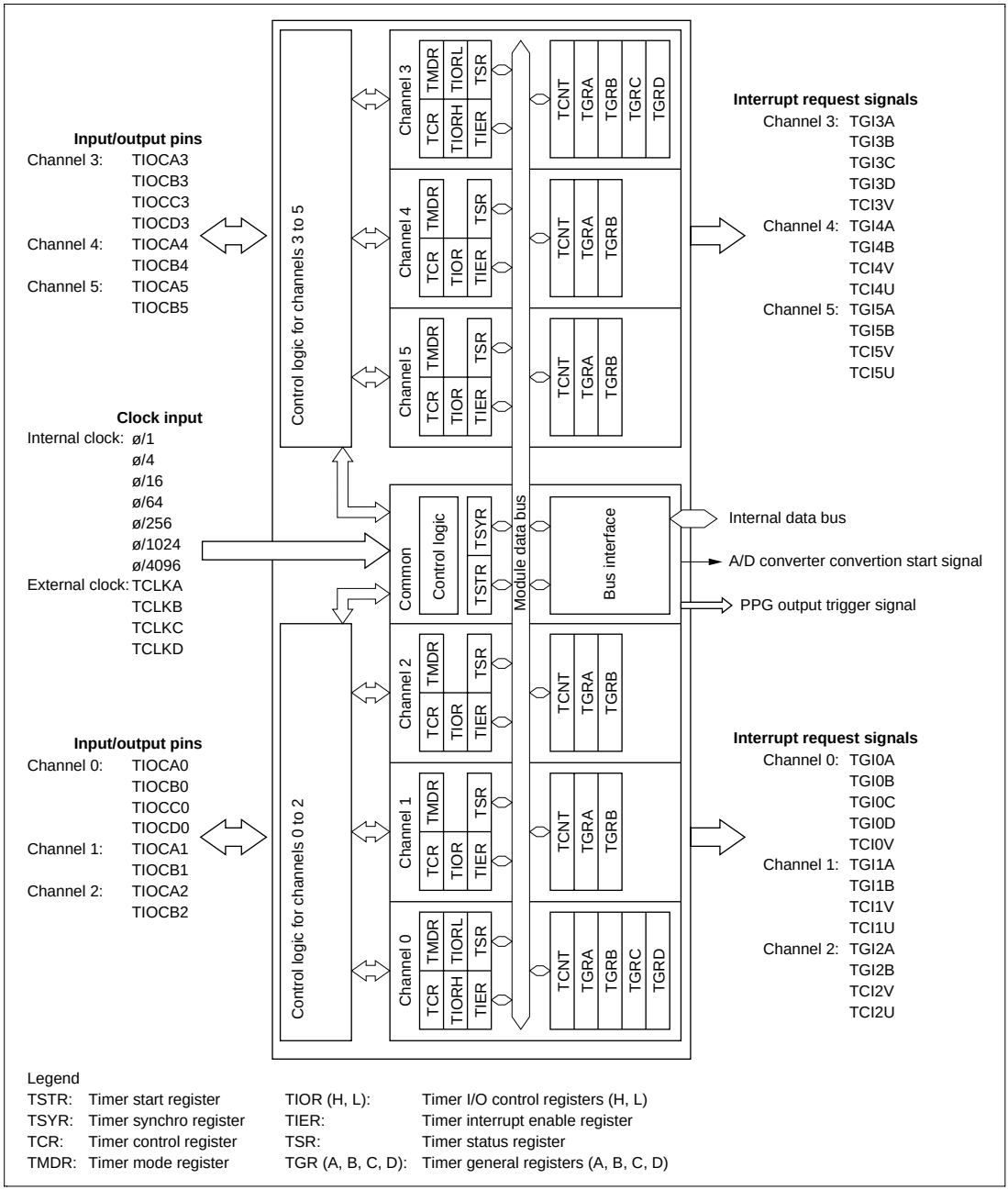


Figure 11-1 Block Diagram of TPU

11.1.3 Pin Configuration

Table 11-2 summarizes the TPU pins.

Table 11-2 TPU Pins

Channel	Name	Symbol	I/O	Function
All	Clock input A	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output compare output/PWM output pin

Channel	Name	Symbol	I/O	Function
3	Input capture/out compare match A3	TIOCA3	I/O	TGR3A input capture input/output compare output/PWM output pin
	Input capture/out compare match B3	TIOCB3	I/O	TGR3B input capture input/output compare output/PWM output pin
	Input capture/out compare match C3	TIOCC3	I/O	TGR3C input capture input/output compare output/PWM output pin
	Input capture/out compare match D3	TIOCD3	I/O	TGR3D input capture input/output compare output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/output compare output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/output compare output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/output compare output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/output compare output/PWM output pin

11.1.4 Register Configuration

Table 11-3 summarizes the TPU registers.

Table 11-3 TPU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address *1
0	Timer control register 0	TCR0	R/W	H'00	H'FF10
	Timer mode register 0	TMDR0	R/W	H'C0	H'FF11
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FF12
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FF13
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FF14
	Timer status register 0	TSR0	R/(W)*2	H'C0	H'FF15
	Timer counter 0	TCNT0	R/W	H'0000	H'FF16
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FF18
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FF1A
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FF1C
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FF1E
1	Timer control register 1	TCR1	R/W	H'00	H'FF20
	Timer mode register 1	TMDR1	R/W	H'C0	H'FF21
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FF22
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FF24
	Timer status register 1	TSR1	R/(W) *2	H'C0	H'FF25
	Timer counter 1	TCNT1	R/W	H'0000	H'FF26
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FF28
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FF2A
2	Timer control register 2	TCR2	R/W	H'00	H'FF30
	Timer mode register 2	TMDR2	R/W	H'C0	H'FF31
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FF32
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FF34
	Timer status register 2	TSR2	R/(W) *2	H'C0	H'FF35
	Timer counter 2	TCNT2	R/W	H'0000	H'FF36
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FF38
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FF3A

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
3	Timer control register 3	TCR3	R/W	H'00	H'FE80
	Timer mode register 3	TMDR3	R/W	H'C0	H'FE81
	Timer I/O control register 3H	TIOR3H	R/W	H'00	H'FE82
	Timer I/O control register 3L	TIOR3L	R/W	H'00	H'FE83
	Timer interrupt enable register 3	TIER3	R/W	H'40	H'FE84
	Timer status register 3	TSR3	R/(W)*[2]	H'C0	H'FE85
	Timer counter 3	TCNT3	R/W	H'0000	H'FE86
	Timer general register 3A	TGR3A	R/W	H'FFFF	H'FE88
	Timer general register 3B	TGR3B	R/W	H'FFFF	H'FE8A
	Timer general register 3C	TGR3C	R/W	H'FFFF	H'FE8C
	Timer general register 3D	TGR3D	R/W	H'FFFF	H'FE8E
4	Timer control register 4	TCR4	R/W	H'00	H'FE90
	Timer mode register 4	TMDR4	R/W	H'C0	H'FE91
	Timer I/O control register 4	TIOR4	R/W	H'00	H'FE92
	Timer interrupt enable register 4	TIER4	R/W	H'40	H'FE94
	Timer status register 4	TSR4	R/(W) *[2]	H'C0	H'FE95
	Timer counter 4	TCNT4	R/W	H'0000	H'FE96
	Timer general register 4A	TGR4A	R/W	H'FFFF	H'FE98
	Timer general register 4B	TGR4B	R/W	H'FFFF	H'FE9A
5	Timer control register 5	TCR5	R/W	H'00	H'FEA0
	Timer mode register 5	TMDR5	R/W	H'C0	H'FEA1
	Timer I/O control register 5	TIOR5	R/W	H'00	H'FEA2
	Timer interrupt enable register 5	TIER5	R/W	H'40	H'FEA4
	Timer status register 5	TSR5	R/(W) *[2]	H'C0	H'FEA5
	Timer counter 5	TCNT5	R/W	H'0000	H'FEA6
	Timer general register 5A	TGR5A	R/W	H'FFFF	H'FEA8
	Timer general register 5B	TGR5B	R/W	H'FFFF	H'FEAA
All	Timer start register	TSTR	R/W	H'00	H'FEB0
	Timer synchro register	TSYR	R/W	H'00	H'FEB1
	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Only 0 can be written, for flag clearing.

11.2 Register Descriptions

11.2.1 Timer Control Register (TCR)

Channel 0: TCR0

Channel 3: TCR3

Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1

Channel 2: TCR2

Channel 4: TCR4

Channel 5: TCR5

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has six TCR registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a reset, and in hardware standby mode.

TCR register settings should be made only when TCNT operation is stopped.

Bits 7, 6, 5—Counter Clear 2, 1, and 0 (CCLR2, CCLR1, CCLR0): These bits select the TCNT counter clearing source.

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture *2
		1	0	TCNT cleared by TGRD compare match/input capture *2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1

Notes: *1 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

*2 When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

*3 Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ø/4 both edges = ø/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority.

Bit 4	Bit 3		
CKEG1	**CKEG0**	**Description**	
0	0	Count at rising edge	(Initial value)
	1	Count at falling edge	
1	—	Count at both edges	

Note: Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Bits 2, 1, and 0—Time Prescaler 2, 1, and 0 (TPSC2 to TPSC0): These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 11-4 shows the clock sources that can be set for each channel.

Table 11-4 TPU Clock Sources

	Internal Clock							External Clock				Overflow/ Underflow on Another
Channel	**ø/1**	**ø/4**	**ø/16**	**ø/64**	**ø/256**	**ø/1024**	**ø/4096**	**TCLKA**	**TCLKB**	**TCLKC**	**TCLKD**	**Channel**
0	○	○	○	○				○	○	○	○	
1	○	○	○	○	○			○	○			○
2	○	○	○	○		○		○	○	○		
3	○	○	○	○	○	○	○	○				
4	○	○	○	○		○		○		○		○
5	○	○	○	○	○			○		○	○	

Legend

○ : Setting

Blank : No setting

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
0	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	External clock: counts on TCLKD pin input	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	Internal clock: counts on ø/256	
			1	Counts on TCNT2 overflow/underflow	

Note: This setting is ignored when channel 1 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
2	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	Internal clock: counts on ø/1024	

Note: This setting is ignored when channel 2 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
3	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	Internal clock: counts on ø/1024	
		1	0	Internal clock: counts on ø/256	
			1	Internal clock: counts on ø/4096	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
4	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		1	0	Internal clock: counts on ø/1024	
			1	Counts on TCNT5 overflow/underflow	

Note: This setting is ignored when channel 4 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
5	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		1	0	Internal clock: counts on ø/256	
			1	External clock: counts on TCLKD pin input	

Note: This setting is ignored when channel 5 is in phase counting mode.

11.2.2 Timer Mode Register (TMDR)

Channel 0: TMDR0

Channel 3: TMDR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1

Channel 2: TMDR2

Channel 4: TMDR4

Channel 5: TMDR5

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. The TMDR registers are initialized to H'C0 by a reset, and in hardware standby mode.

TMDR register settings should be made only when TCNT operation is stopped.

Bits 7 and 6—Reserved: These bits are always read as 1 and cannot be modified.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

BFB	Description	
0	TGRB operates normally	(Initial value)
1	TGRB and TGRD used together for buffer operation	

Bit 4—Buffer Operation A (BFA): Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.

Bit 4		
BFA	**Description**	
0	TGRA operates normally	(Initial value)
1	TGRA and TGRC used together for buffer operation	

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operating mode.

Bit 3	Bit 2	Bit 1	Bit 0		
MD3*[1]	**MD2*[2]**	**MD1**	**MD0**	**Description**	
0	0	0	0	Normal operation	(Initial value)
			1	Reserved	
		1	0	PWM mode 1	
			1	PWM mode 2	
	1	0	0	Phase counting mode 1	
			1	Phase counting mode 2	
		1	0	Phase counting mode 3	
			1	Phase counting mode 4	
1	*	*	*	—	

*: Don't care

Notes: *1 MD3 is a reserved bit. In a write, it should always be written with 0.

*2 Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

11.2.3 Timer I/O Control Register (TIOR)

Channel 0: TIOR0H

Channel 1: TIOR1

Channel 2: TIOR2

Channel 3: TIOR3H

Channel 4: TIOR4

Channel 5: TIOR5

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L

Channel 3: TIOR3L

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The TIOR registers are initialized to H'00 by a reset, and in hardware standby mode.

Care is required since TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

Bits 7 to 4— I/O Control B3 to B0 (IOB3 to IOB0)
I/O Control D3 to D0 (IOD3 to IOD0):

Bits IOB3 to IOB0 specify the function of TGRB.
Bits IOD3 to IOD0 specify the function of TGRD.

	Bit 7	Bit 6	Bit 5	Bit 4			
Channel	**IOB3**	**IOB2**	**IOB1**	**IOB0**	**Description**		
0	0	0	0	0	TGR0B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count- up/count-down*[1]

*: Don’t care

Note: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

Channel	Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
0	0	0	0	0	TGR0D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
3	0	0	0	0	TGR3B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down*[1]

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

Channel	Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
3	0	0	0	0	TGR3D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
4	0	0	0	0	TGR4B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/ input capture

*: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
5	0	0	0	0	TGR5B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

Bits 3 to 0— I/O Control A3 to A0 (IOA3 to IOA0)
I/O Control C3 to C0 (IOC3 to IOC0):

IOA3 to IOA0 specify the function of TGRA.
IOC3 to IOC0 specify the function of TGRC.

	Bit 3	Bit 2	Bit 1	Bit 0			
Channel	**IOA3**	**IOA2**	**IOA1**	**IOA0**	**Description**		
0	0	0	0	0	TGR0A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/ count clock	Input capture at TCNT1 count-up/count-down

*: Don't care

Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
0	0	0	0	0	TGR0C is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
1	0	0	0	0	TGR1A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/input capture

*: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
3	0	0	0	0	TGR3A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

*: Don't care

Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
3	0	0	0	0	TGR3C is output compare register*[1]	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3C is input capture register*[1]	Capture input source is TIOCC3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

*: Don't care

Note: *1 When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
4	0	0	0	0	TGR4A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

*: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
5	0	0	0	0	TGR5A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

11.2.4 Timer Interrupt Enable Register (TIER)

Channel 0: TIER0

Channel 3: TIER3

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TIER1

Channel 2: TIER2

Channel 4: TIER4

Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel. The TIER registers are initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

Bit 7

TTGE	Description	
0	A/D conversion start request generation disabled	(Initial value)
1	A/D conversion start request generation enabled	

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

TCIEU	Description	
0	Interrupt requests (TCIU) by TCFU disabled	(Initial value)
1	Interrupt requests (TCIU) by TCFU enabled	

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4

TCIEV	Description	
0	Interrupt requests (TCIV) by TCFV disabled	(Initial value)
1	Interrupt requests (TCIV) by TCFV enabled	

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3

TGIED	Description	
0	Interrupt requests (TGID) by TGFD bit disabled	(Initial value)
1	Interrupt requests (TGID) by TGFD bit enabled	

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2

TGIEC	Description	
0	Interrupt requests (TGIC) by TGFC bit disabled	(Initial value)
1	Interrupt requests (TGIC) by TGFC bit enabled	

Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.

Bit 1

TGIEB	Description	
0	Interrupt requests (TGIB) by TGFB bit disabled	(Initial value)
1	Interrupt requests (TGIB) by TGFB bit enabled	

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0

TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(Initial value)
1	Interrupt requests (TGIA) by TGFA bit enabled	

11.2.5 Timer Status Register (TSR)

Channel 0: TSR0

Channel 3: TSR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, for flag clearing.

Channel 1: TSR1

Channel 2: TSR2

Channel 4: TSR4

Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Note: * Only 0 can be written, for flag clearing.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU has six TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, and in hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7

TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Flag (TCFU): Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

TCFU	Description	
0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1	(Initial value)
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)	

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occurred.

Bit 4

TCFV	Description	
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1	(Initial value)
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)	

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3

TGFD	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Bit 2—Input Capture/Output Compare Flag C (TGFC): Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2

TGFC	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the occurrence of TGRB input capture or compare match.

Bit 1

TGFB	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Bit 0—Input Capture/Output Compare Flag A (TGFA): Status flag that indicates the occurrence of TGRA input capture or compare match.

Bit 0

TGFA	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

11.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)

Channel 1: TCNT1 (up/down-counter*)

Channel 2: TCNT2 (up/down-counter*)

Channel 3: TCNT3 (up-counter)

Channel 4: TCNT4 (up/down-counter*)

Channel 5: TCNT5 (up/down-counter*)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	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: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

11.2.7 Timer General Register (TGR)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	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

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardware standby mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

11.2.8 Timer Start Register (TSTR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. TSTR is initialized to H'00 by a reset, and in hardware standby mode. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stoppage for TCNT.

Bit n	
CSTn	**Description**
0	TCNTn count operation is stopped (Initial value)
1	TCNTn performs count operation

n = 5 to 0

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

11.2.9 Timer Synchro Register (TSYR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Timer Synchro 5 to 0 (SYNC5 to SYNC0): These bits select whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels*[1], and synchronous clearing through counter clearing on another channel*[2] are possible.

Bit n

SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) (Initial value)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

n = 5 to 0

Notes: *1 To set synchronous operation, the SYNC bits for at least two channels must be set to 1.

*2 To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.

11.2.10 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA5 bit in MSTPCRA is set to 1, TPU operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 5—Module Stop (MSTPA5): Specifies the TPU module stop mode.

Bit 5

MSTPA5	Description	
0	TPU module stop mode cleared	
1	TPU module stop mode set	(Initial value)

11.3 Interface to Bus Master

11.3.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 11-2.

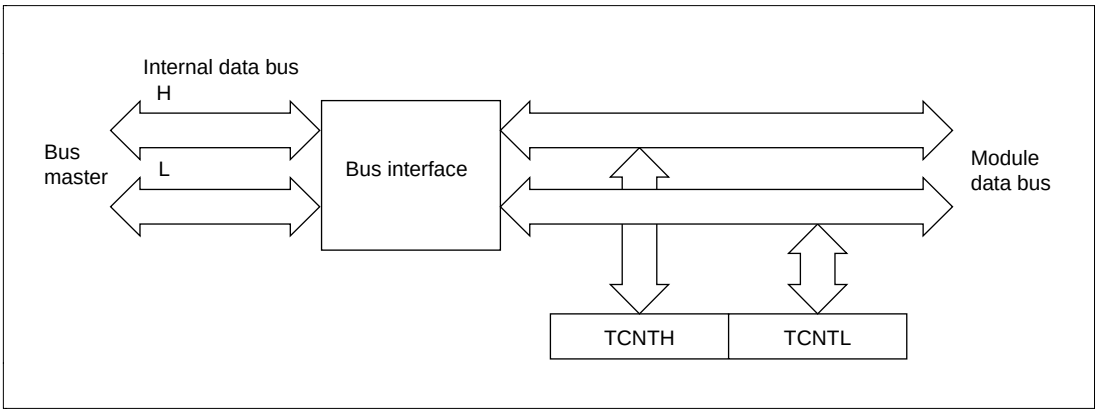


Figure 11-2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

11.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 11-3, 11-4, and 11-5.

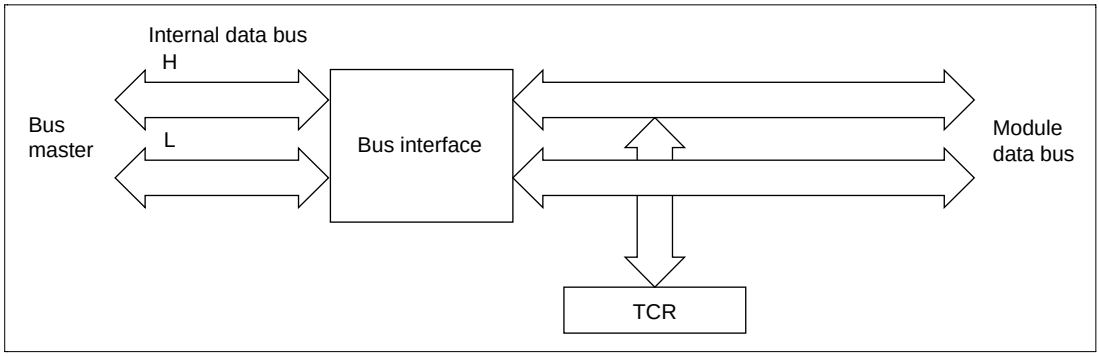


Figure 11-3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

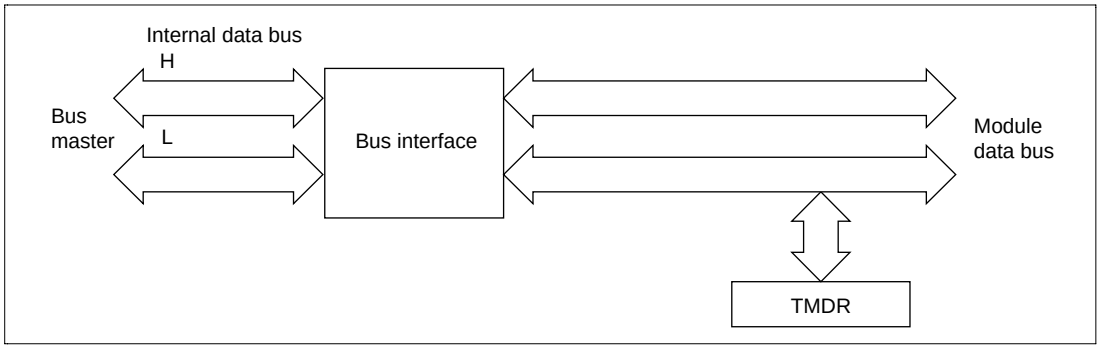


Figure 11-4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

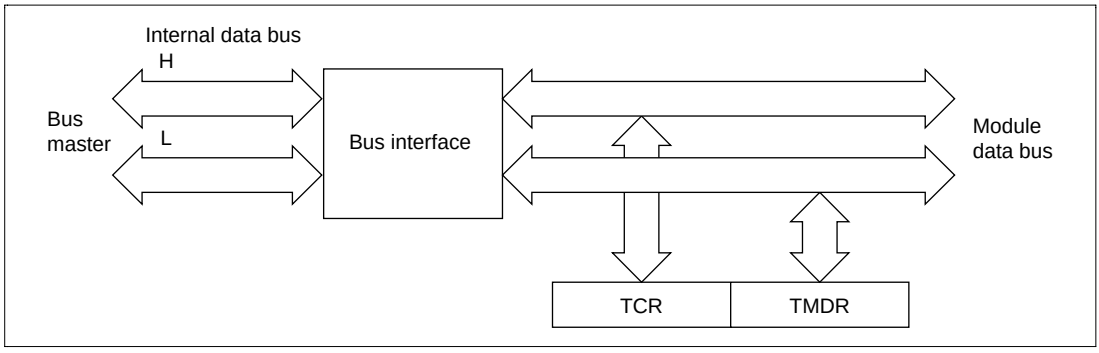


Figure 11-5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

11.4 Operation

11.4.1 Overview

Operation in each mode is outlined below.

Normal Operation: Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, TCNT for that channel performs synchronous presetting. That is, when TCNT for a channel designated for synchronous operation is rewritten, the TCNT counters for the other channels are also rewritten at the same time. Synchronous clearing of the TCNT counters is also possible by setting the timer synchronization bits in TSYR for channels designated for synchronous operation.

Buffer Operation

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR.

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value previously held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1), channel 2 counter (TCNT2), channel 4 counter (TCNT4), and channel 5 counter (TCNT5) can be connected together to operate as a 32-bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5. When phase counting mode is set, the corresponding TCLK pin functions as the clock pin, and TCNT performs up- or down-counting.

This can be used for two-phase encoder pulse input.

11.4.2 Basic Functions

Counter Operation: When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

- Example of count operation setting procedure

  Figure 11-6 shows an example of the count operation setting procedure.

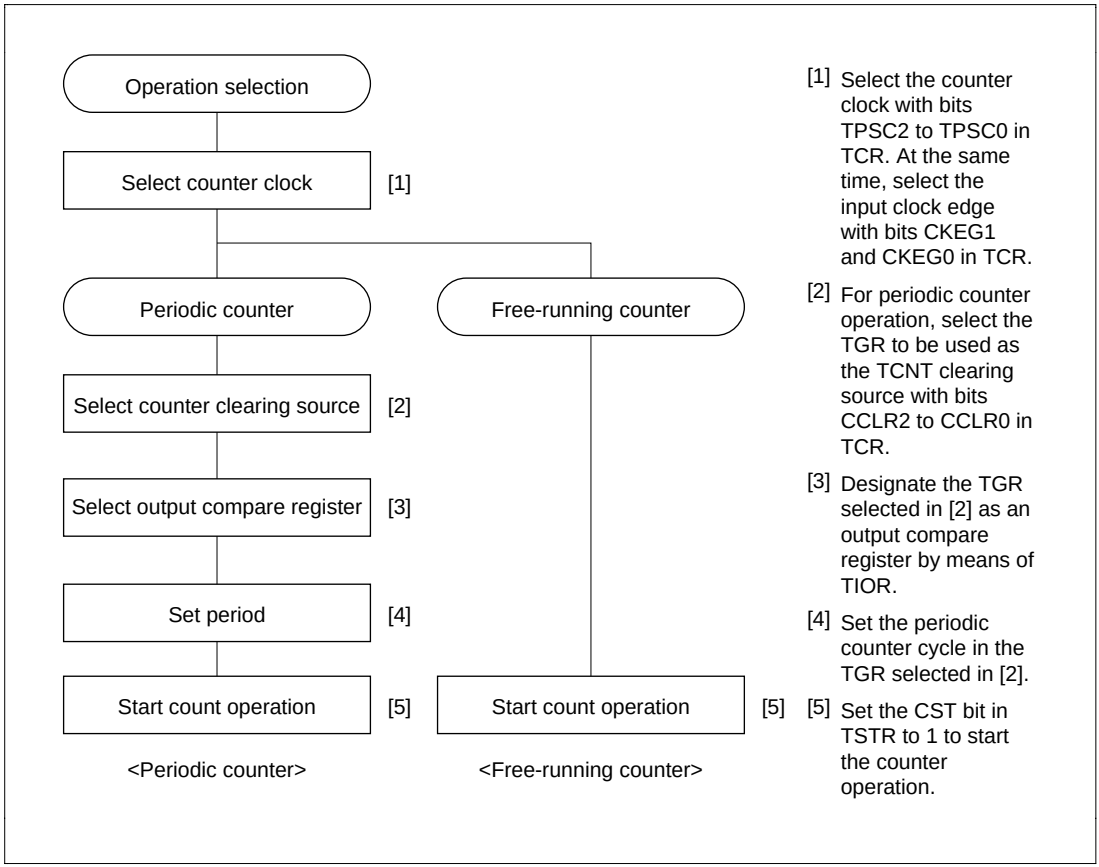


Figure 11-6 Example of Counter Operation Setting Procedure

- Free-running count operation and periodic count operation

  Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

  Figure 11-7 illustrates free-running counter operation.

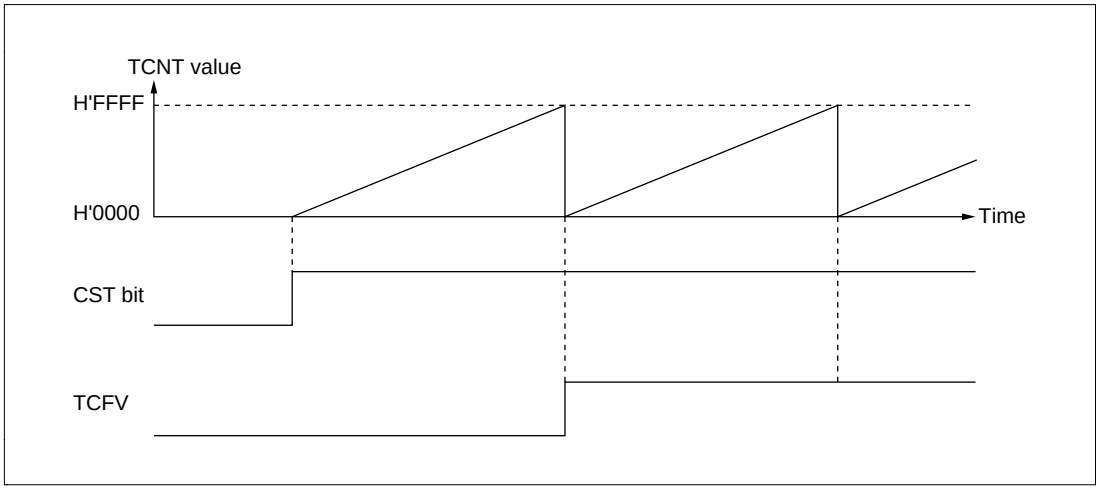


Figure 11-7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 11-8 illustrates periodic counter operation.

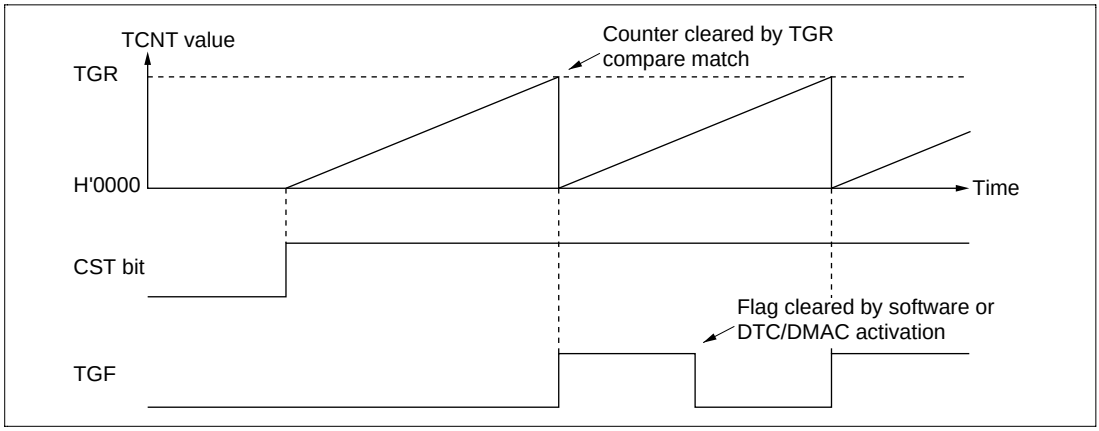


Figure 11-8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match
  Figure 11-9 shows an example of the setting procedure for waveform output by compare match

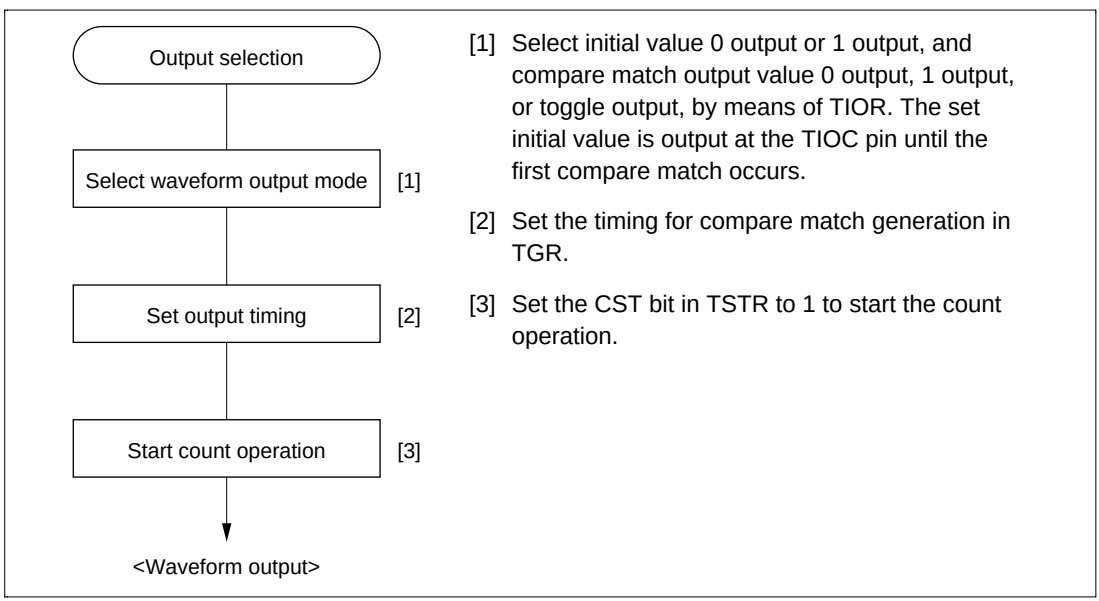


Figure 11-9 Example Of Setting Procedure For Waveform Output By Compare Match

- Examples of waveform output operation

  Figure 11-10 shows an example of 0 output/1 output.

  In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

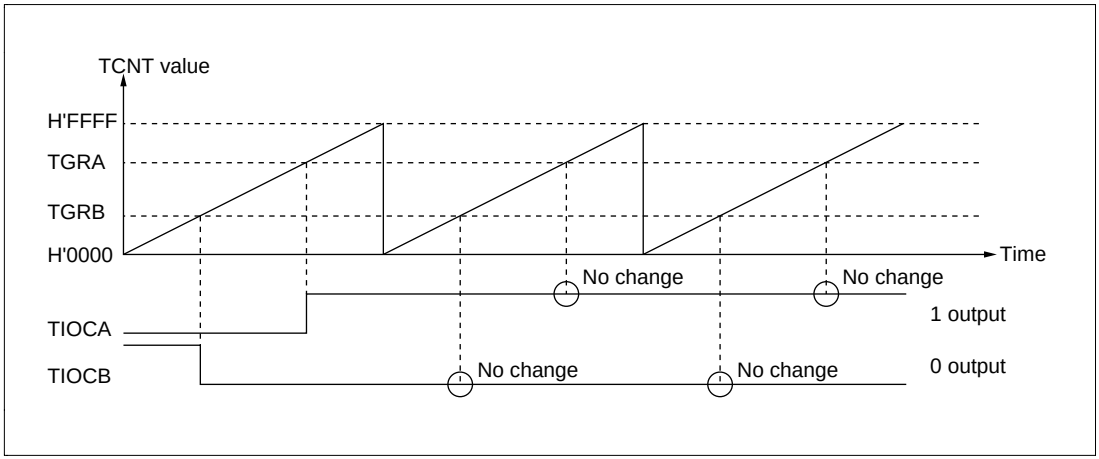


Figure 11-10 Example of 0 Output/1 Output Operation

Figure 11-11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

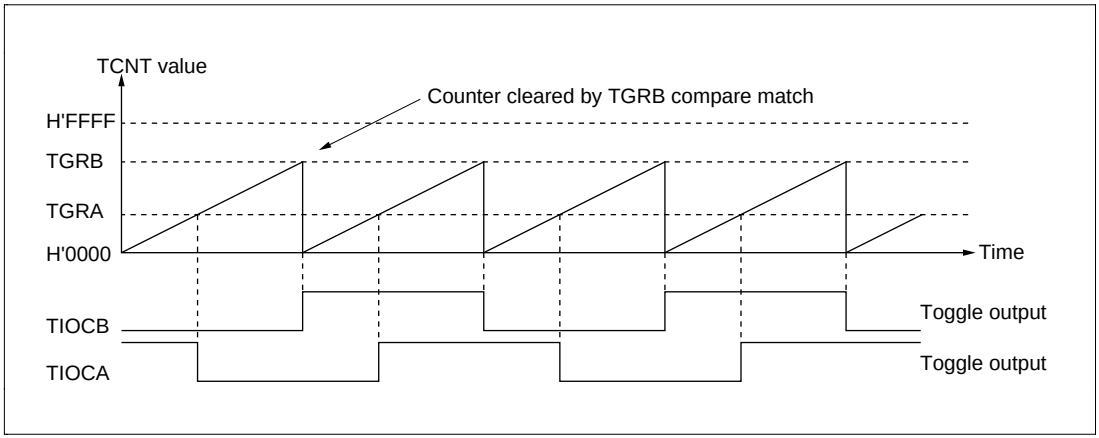


Figure 11-11 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, ø/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ø/1 is selected.

- Example of input capture operation setting procedure
  Figure 11-12 shows an example of the input capture operation setting procedure.

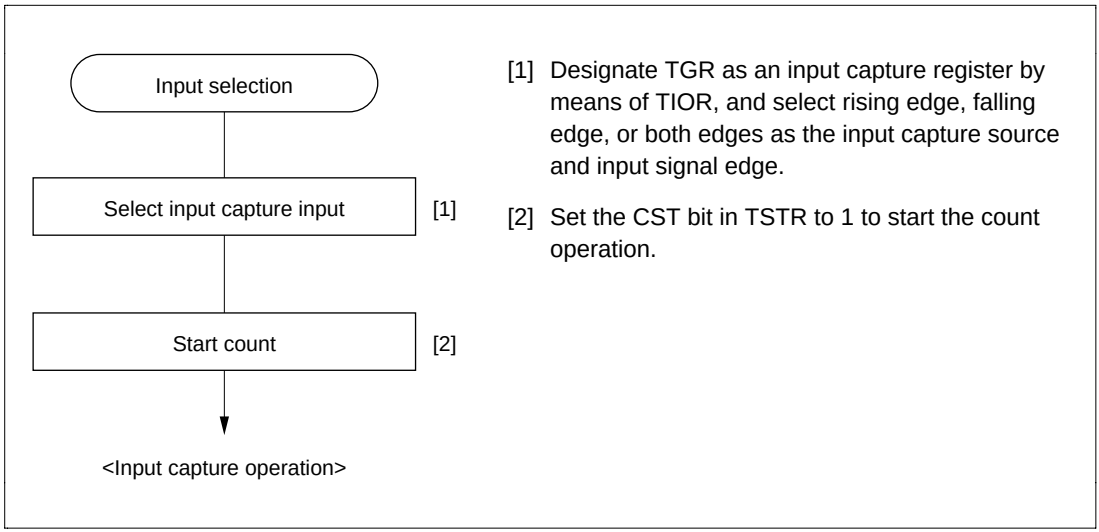


Figure 11-12 Example of Input Capture Operation Setting Procedure

- Example of input capture operation

  Figure 11-13 shows an example of input capture operation.

  In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

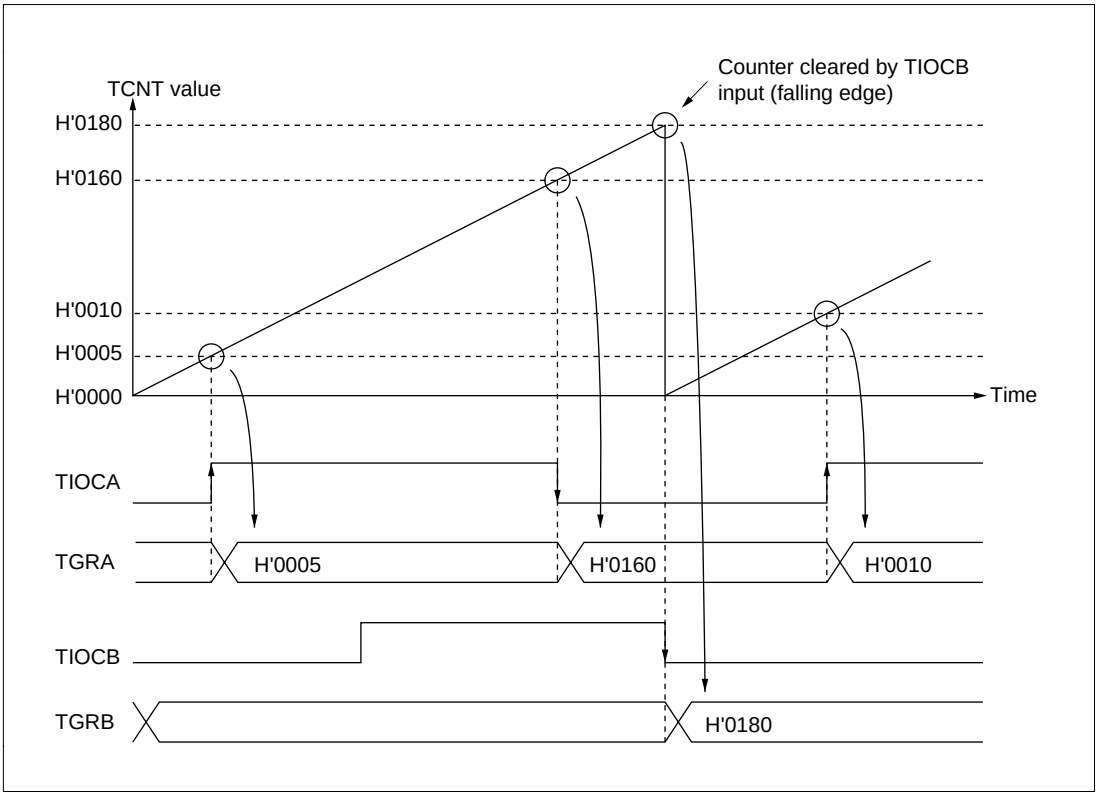


Figure 11-13 Example of Input Capture Operation

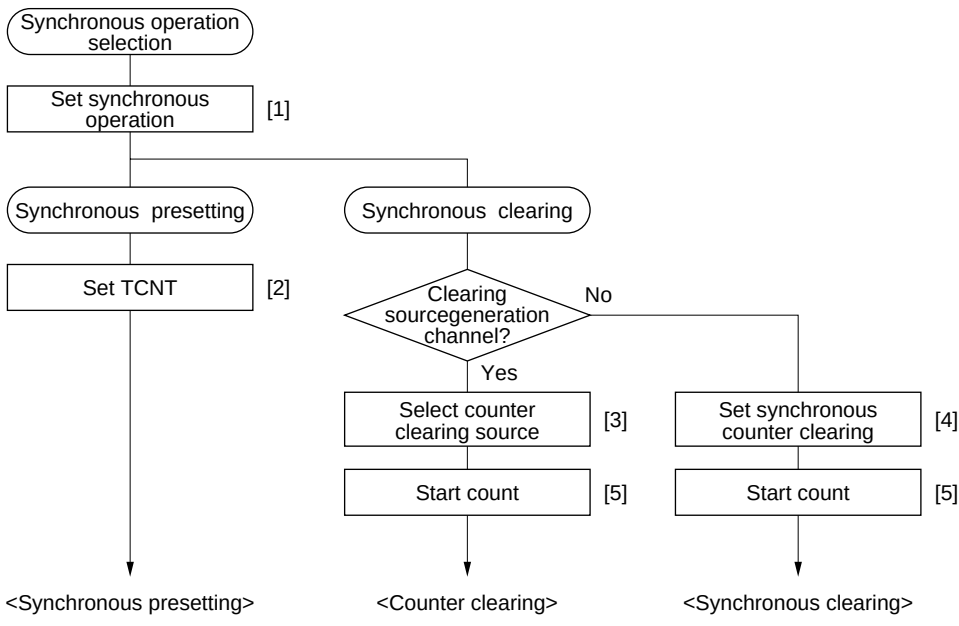
11.4.3 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 11-14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 11-14 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 11-15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGR0B compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycle.

For details of PWM modes, see section 11.4.6, PWM Modes.

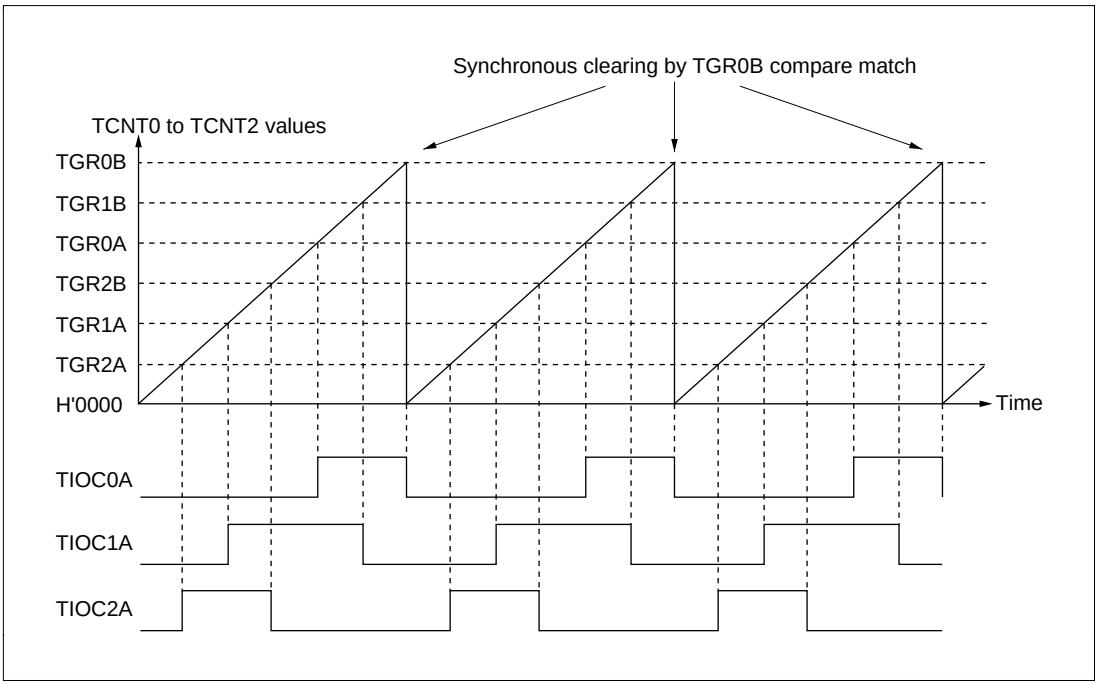


Figure 11-15 Example of Synchronous Operation

11.4.4 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 11-5 shows the register combinations used in buffer operation.

Table 11-5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register
  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 11-16.

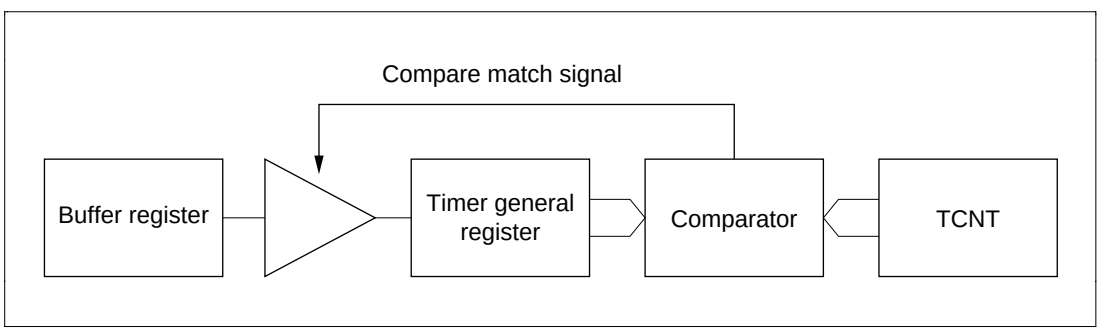


Figure 11-16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11-17.

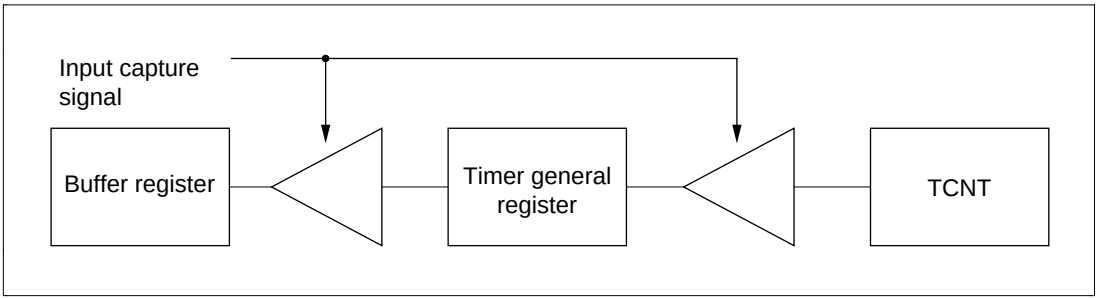


Figure 11-17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 11-18 shows an example of the buffer operation setting procedure.

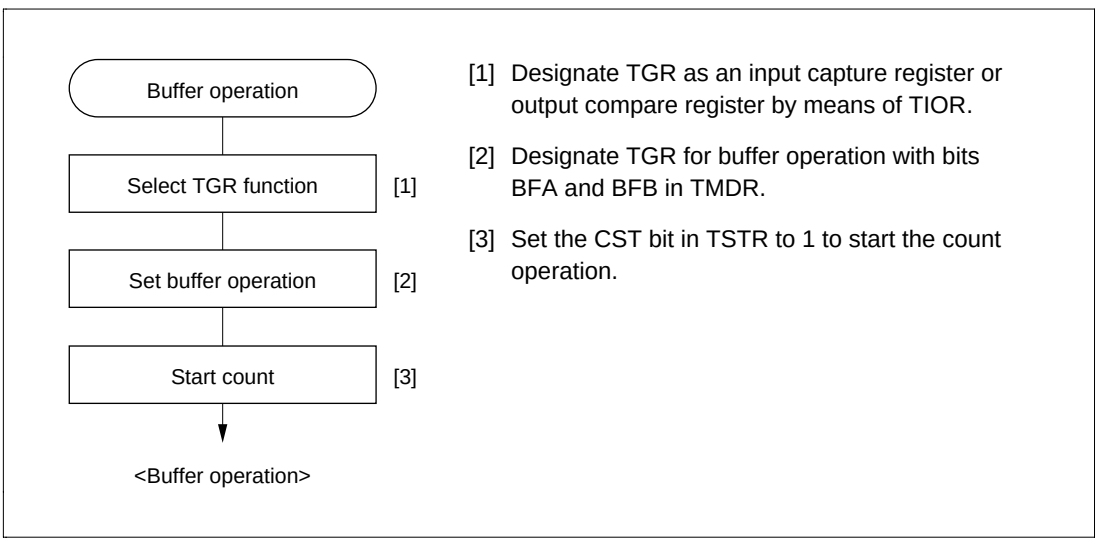


Figure 11-18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

- When TGR is an output compare register

  Figure 11-19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

  As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

  For details of PWM modes, see section 11.4.6, PWM Modes.

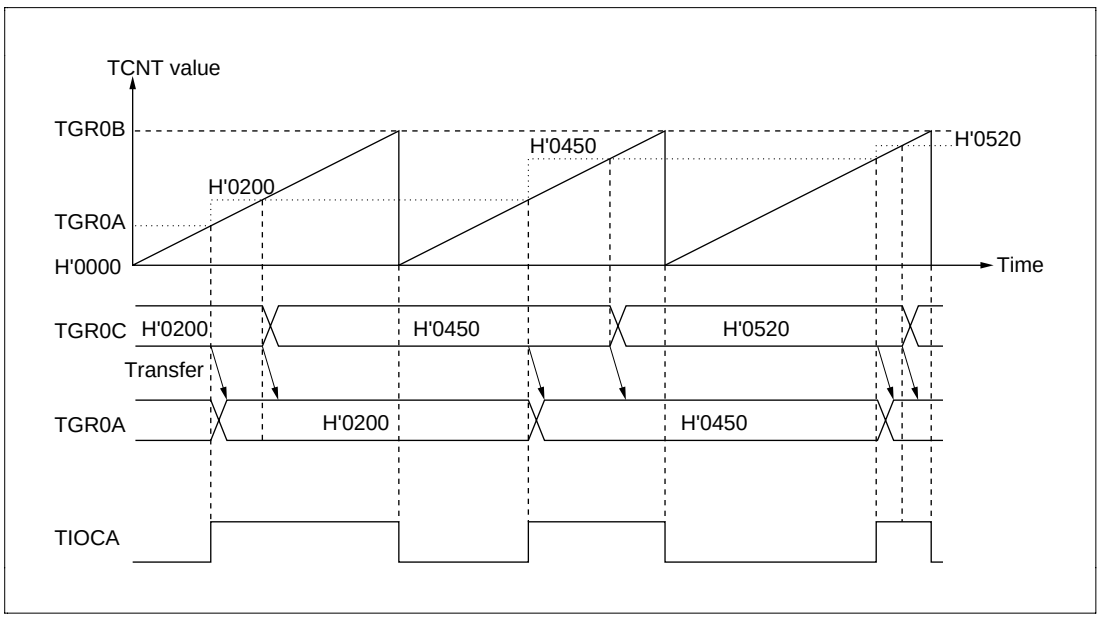


Figure 11-19 Example of Buffer Operation (1)

- When TGR is an input capture register

  Figure 11-20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

  Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

  As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

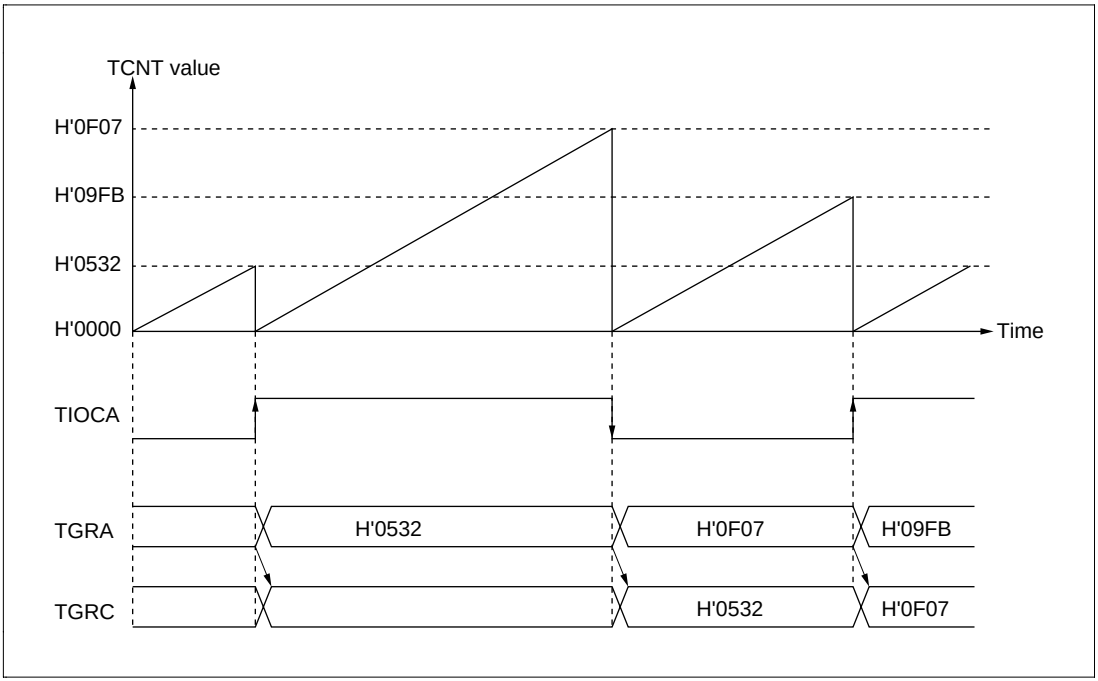


Figure 11-20 Example of Buffer Operation (2)

11.4.5 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT2 (TCNT5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 11-6 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 11-6 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure: Figure 11-21 shows an example of the setting procedure for cascaded operation.

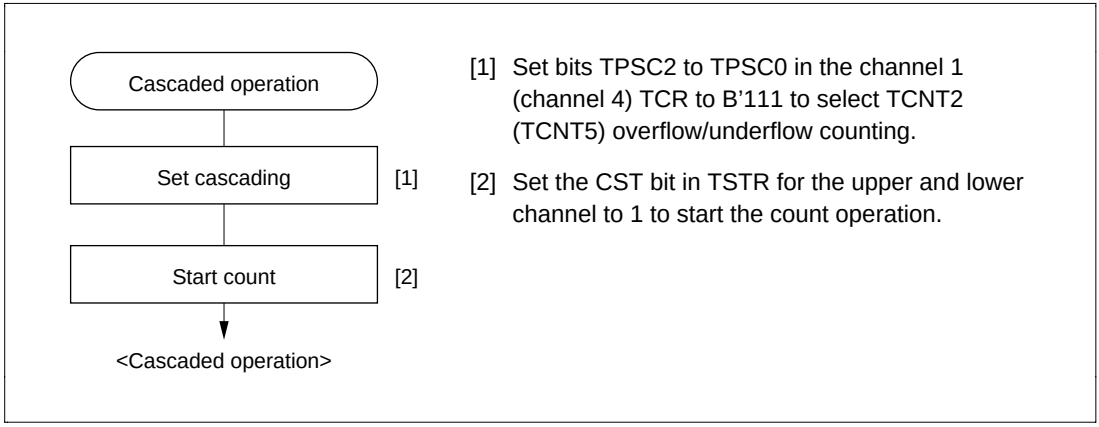


Figure 11-21 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 11-22 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, TGR1A and TGR2A have been designated as input capture registers, and TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGR1A, and the lower 16 bits to TGR2A.

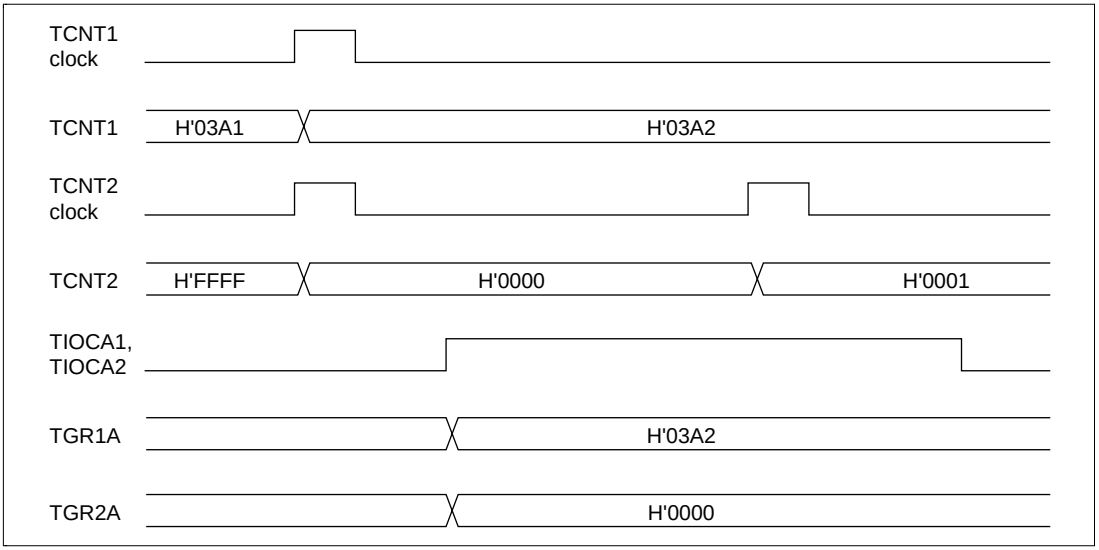


Figure 11-22 Example of Cascaded Operation (1)

Figure 11-23 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

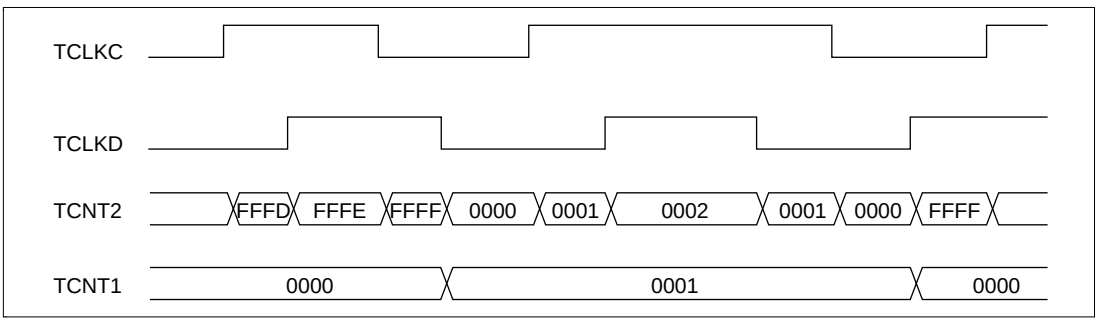


Figure 11-23 Example of Cascaded Operation (2)

11.4.6 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

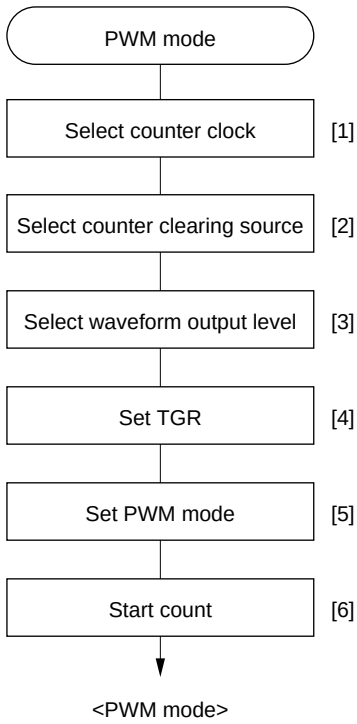
The correspondence between PWM output pins and registers is shown in table 11-7.

Table 11-7 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGR0A	TIOCA0	TIOCA0
	TGR0B		TIOCB0
	TGR0C	TIOCC0	TIOCC0
	TGR0D		TIOCD0
1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2
3	TGR3A	TIOCA3	TIOCA3
	TGR3B		TIOCB3
	TGR3C	TIOCC3	TIOCC3
	TGR3D		TIOCD3
4	TGR4A	TIOCA4	TIOCA4
	TGR4B		TIOCB4
5	TGR5A	TIOCA5	TIOCA5
	TGR5B		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 11-24 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the cycle in the TGR selected in [2], and set the duty in the other the TGR.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 11-24 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 11-25 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB registers as the duty.

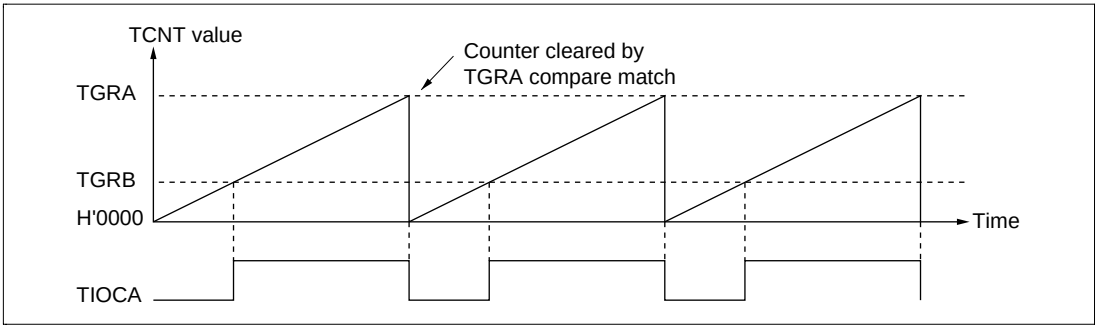


Figure 11-25 Example of PWM Mode Operation (1)

Figure 11-26 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGR0A to TGR0D, TGR1A), to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other TGRs as the duty.

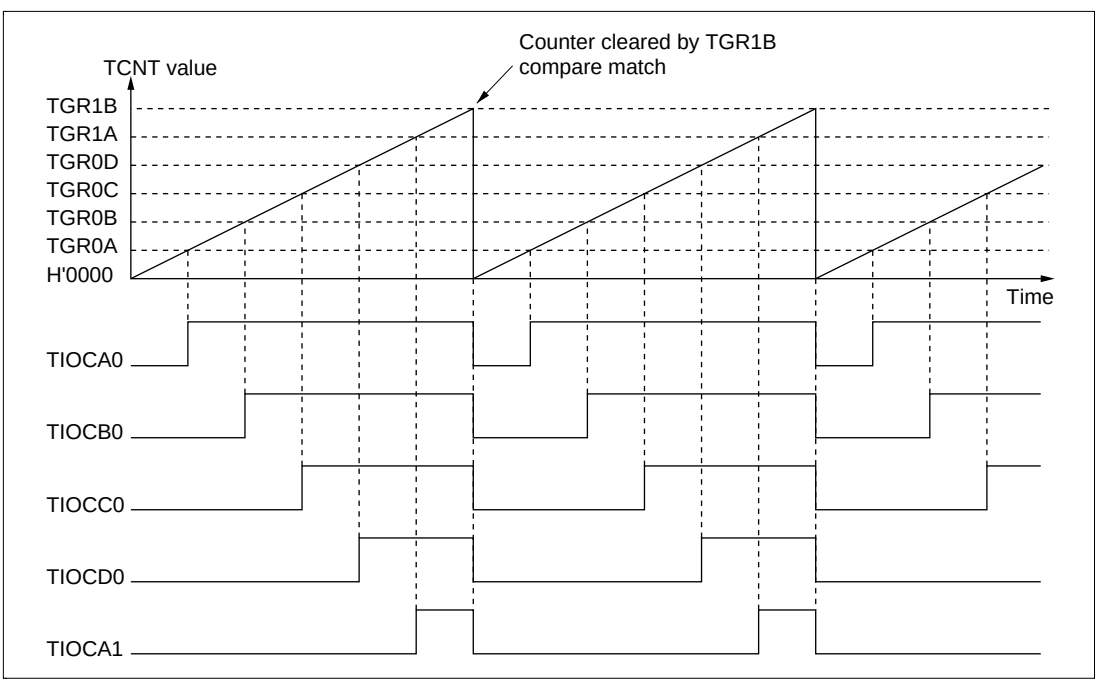


Figure 11-26 Example of PWM Mode Operation (2)

Figure 11-27 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

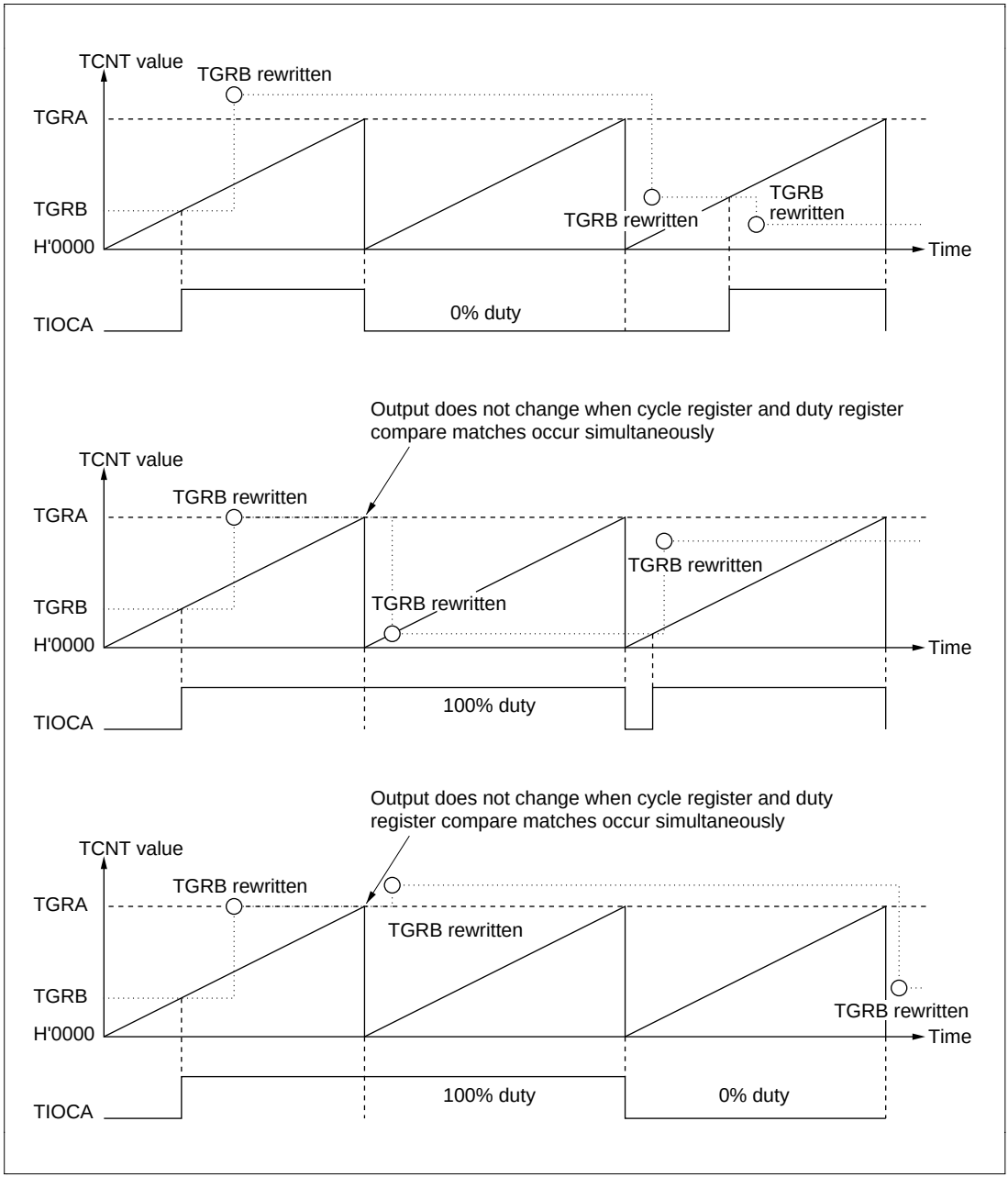


Figure 11-27 Example of PWM Mode Operation (3)

11.4.7 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 11-8 shows the correspondence between external clock pins and channels.

Table 11-8 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 11-28 shows an example of the phase counting mode setting procedure.

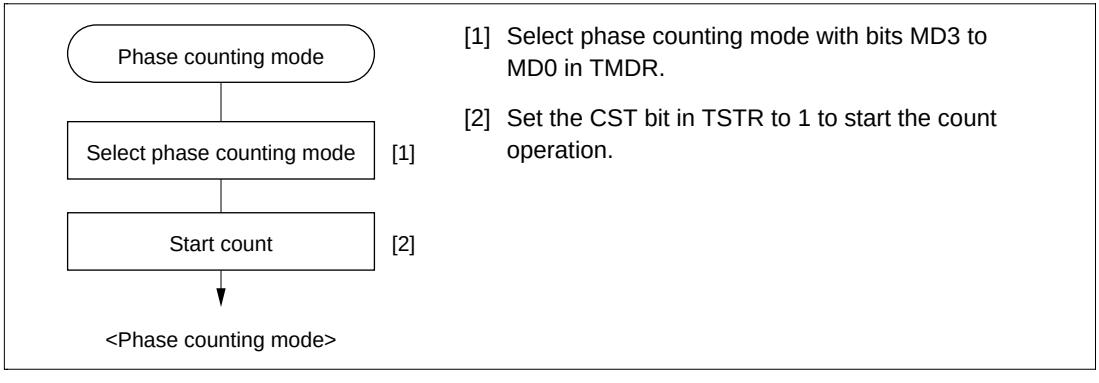


Figure 11-28 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

- Phase counting mode 1

  Figure 11-29 shows an example of phase counting mode 1 operation, and table 11-9 summarizes the TCNT up/down-count conditions.

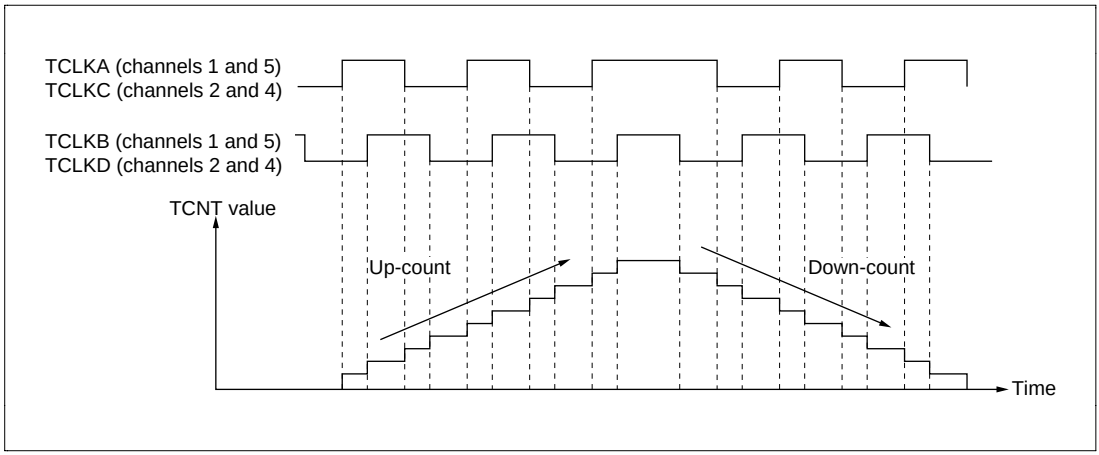


Figure 11-29 Example of Phase Counting Mode 1 Operation

Table 11-9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 2

  Figure 11-30 shows an example of phase counting mode 2 operation, and table 11-10 summarizes the TCNT up/down-count conditions.

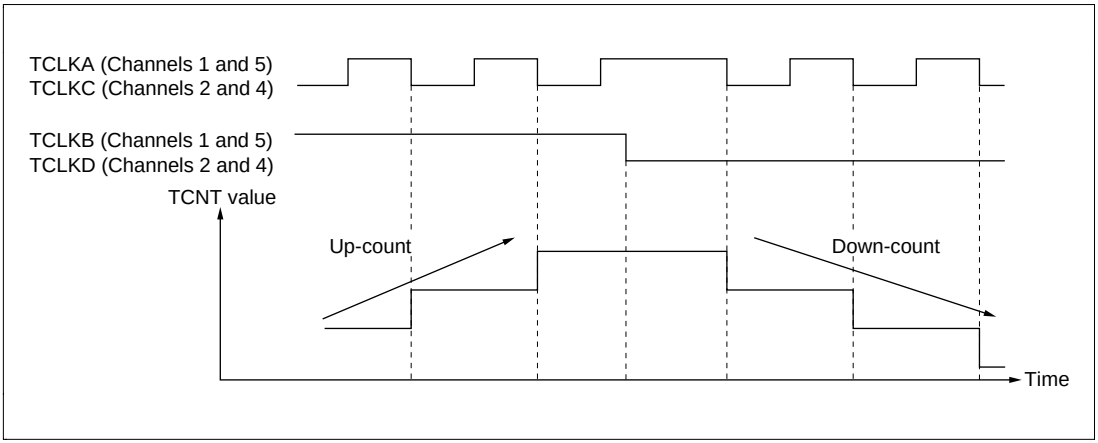


Figure 11-30 Example of Phase Counting Mode 2 Operation

Table 11-10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 3

  Figure 11-31 shows an example of phase counting mode 3 operation, and table 11-11 summarizes the TCNT up/down-count conditions.

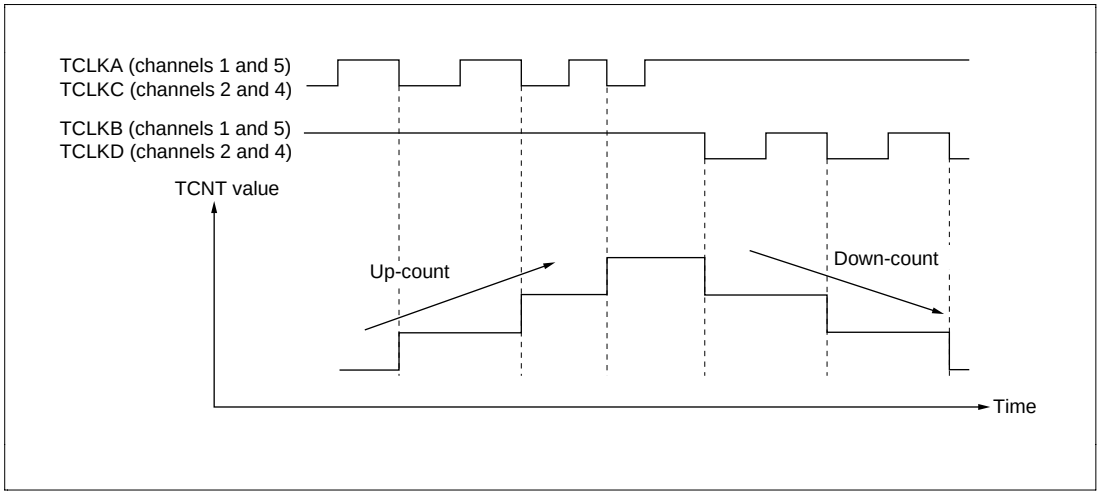


Figure 11-31 Example of Phase Counting Mode 3 Operation

Table 11-11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Down-count
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Don't care

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

- Phase counting mode 4

  Figure 11-32 shows an example of phase counting mode 4 operation, and table 11-12 summarizes the TCNT up/down-count conditions.

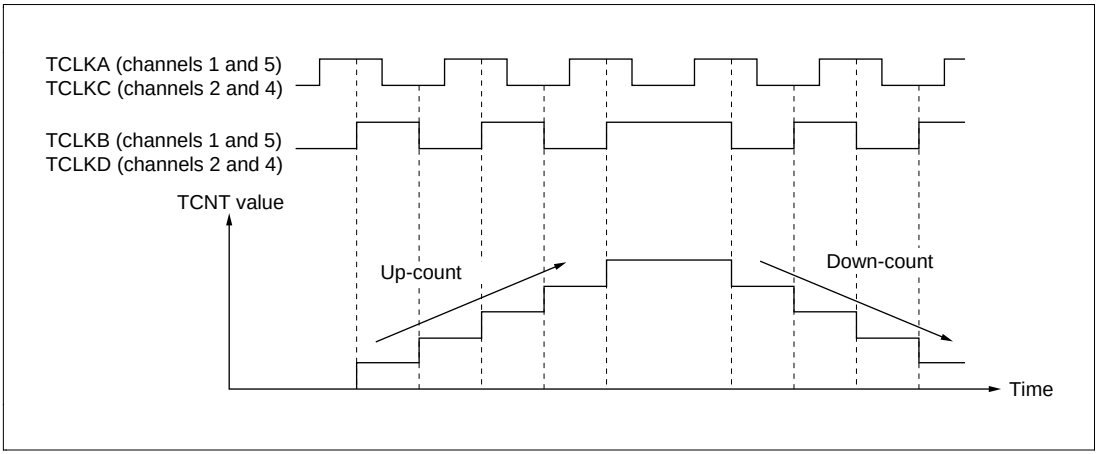


Figure 11-32 Example of Phase Counting Mode 4 Operation

Table 11-12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

Phase Counting Mode Application Example: Figure 11-33 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGR0C compare match; TGR0A and TGR0C are used for the compare match function, and are set with the speed control period and position control period. TGR0B is used for input capture, with TGR0B and TGR0D operating in buffer mode. The channel 1 counter input clock is designated as the TGR0B input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A and TGR0C compare matches are selected as the input capture source, and store the up/down-counter values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

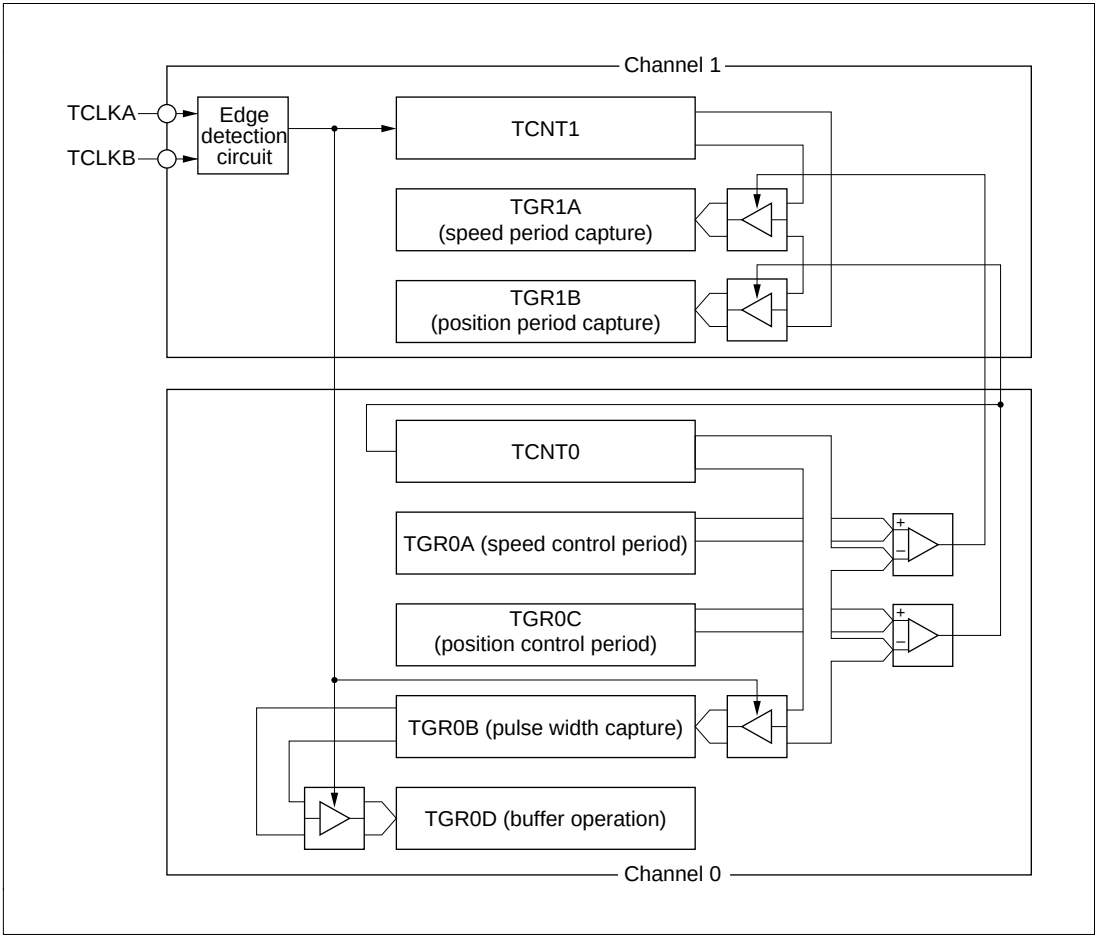


Figure 11-33 Phase Counting Mode Application Example

11.5 Interrupts

11.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 11-13 lists the TPU interrupt sources.

Table 11-13 TPU Interrupts

Channel	Interrupt Source	Description	DMAC Activation	DTC Activation	Priority
0	TGI0A	TGR0A input capture/compare match	Possible	Possible	High
	TGI0B	TGR0B input capture/compare match	Not possible	Possible	
	TGI0C	TGR0C input capture/compare match	Not possible	Possible	
	TGI0D	TGR0D input capture/compare match	Not possible	Possible	
	TCI0V	TCNT0 overflow	Not possible	Not possible	
1	TGI1A	TGR1A input capture/compare match	Possible	Possible	
	TGI1B	TGR1B input capture/compare match	Not possible	Possible	
	TCI1V	TCNT1 overflow	Not possible	Not possible	
	TCI1U	TCNT1 underflow	Not possible	Not possible	
2	TGI2A	TGR2A input capture/compare match	Possible	Possible	
	TGI2B	TGR2B input capture/compare match	Not possible	Possible	
	TCI2V	TCNT2 overflow	Not possible	Not possible	
	TCI2U	TCNT2 underflow	Not possible	Not possible	
3	TGI3A	TGR3A input capture/compare match	Possible	Possible	
	TGI3B	TGR3B input capture/compare match	Not possible	Possible	
	TGI3C	TGR3C input capture/compare match	Not possible	Possible	
	TGI3D	TGR3D input capture/compare match	Not possible	Possible	
	TCI3V	TCNT3 overflow	Not possible	Not possible	
4	TGI4A	TGR4A input capture/compare match	Possible	Possible	
	TGI4B	TGR4B input capture/compare match	Not possible	Possible	
	TCI4V	TCNT4 overflow	Not possible	Not possible	
	TCI4U	TCNT4 underflow	Not possible	Not possible	
5	TGI5A	TGR5A input capture/compare match	Possible	Possible	
	TGI5B	TGR5B input capture/compare match	Not possible	Possible	
	TCI5V	TCNT5 overflow	Not possible	Not possible	
	TCI5U	TCNT5 underflow	Not possible	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

11.5.2 DTC/DMAC Activation

DTC Activation: The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 9, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

DMAC Activation: It is possible to activate the DMAC by the TGRA input capture/compare match interrupt for each channel. See section 8, DMA Controller for details.

In TPU, it is possible to set the TGRA input capture/compare match interrupts for each channel, giving a total of 6, as DMAC activation factors.

11.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

11.6 Operation Timing

11.6.1 Input/Output Timing

TCNT Count Timing: Figure 11-34 shows TCNT count timing in internal clock operation, and figure 11-35 shows TCNT count timing in external clock operation.

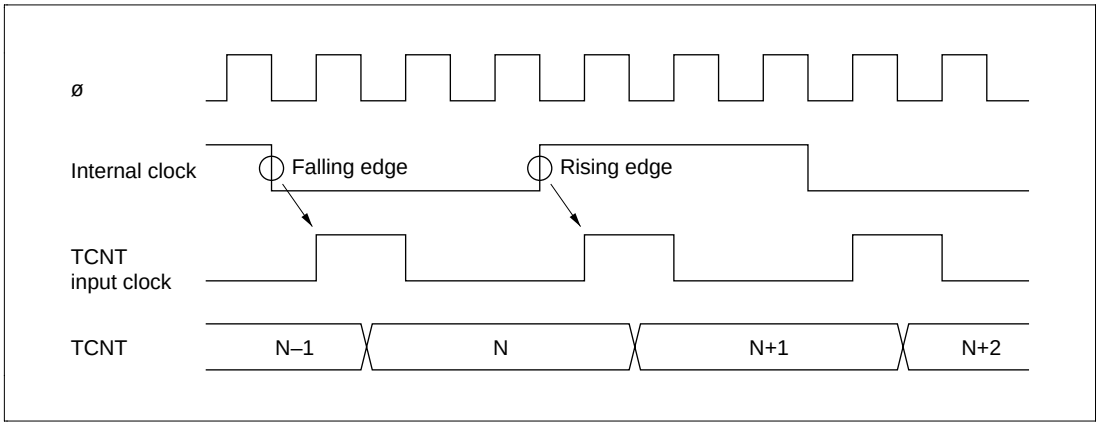


Figure 11-34 Count Timing in Internal Clock Operation

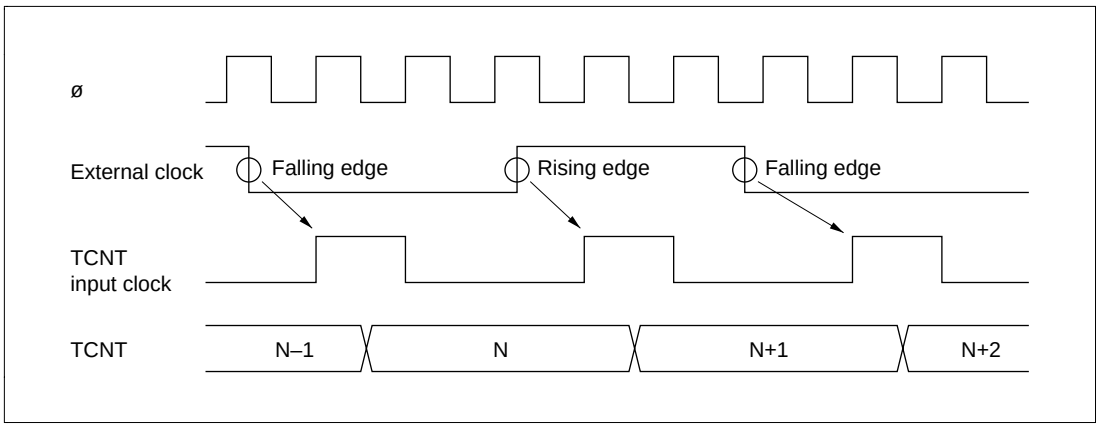


Figure 11-35 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 11-36 shows output compare output timing.

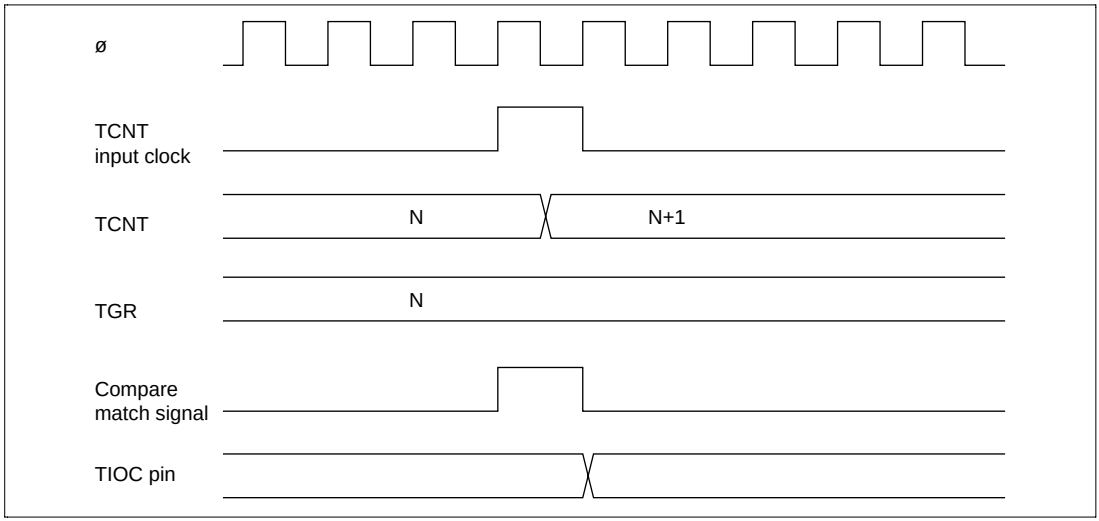


Figure 11-36 Output Compare Output Timing

Input Capture Signal Timing: Figure 11-37 shows input capture signal timing.

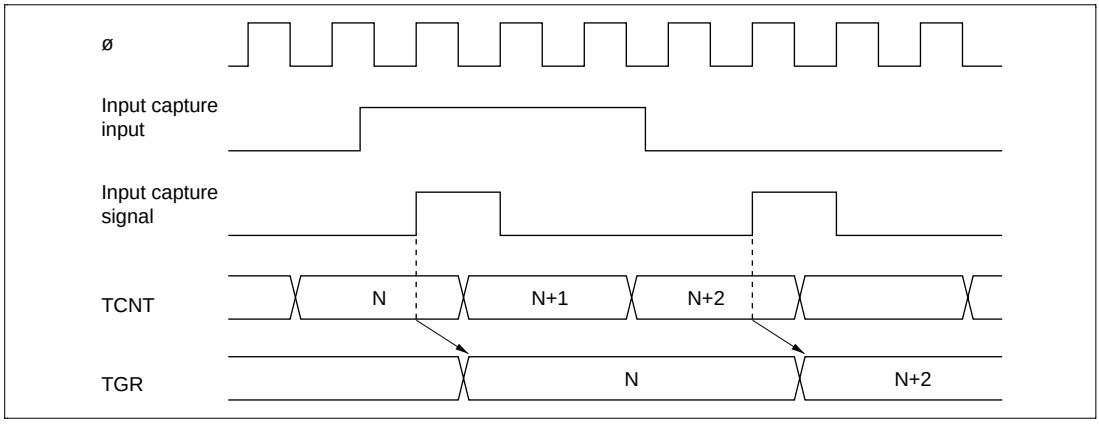


Figure 11-37 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 11-38 shows the timing when counter clearing by compare match occurrence is specified, and figure 11-39 shows the timing when counter clearing by input capture occurrence is specified.

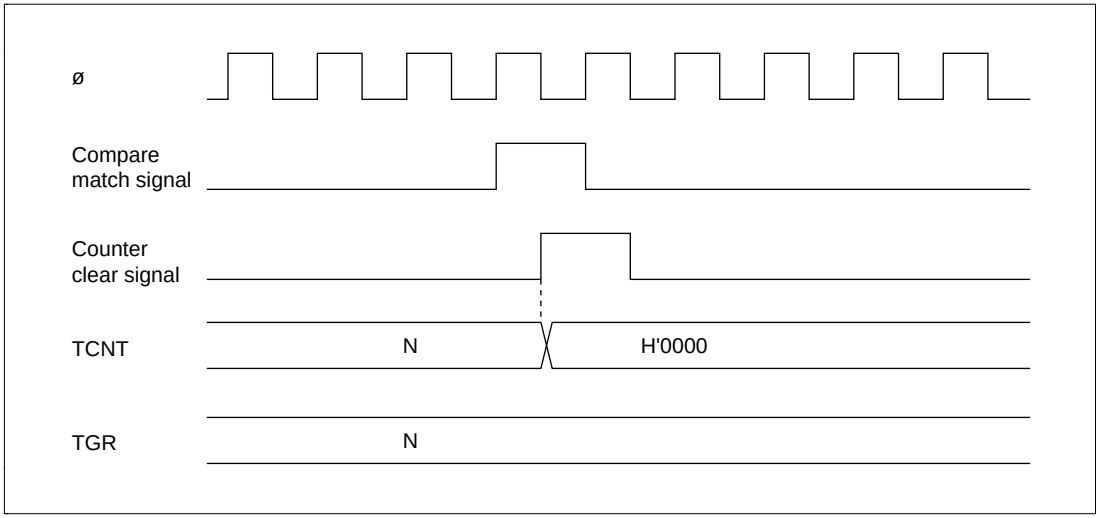


Figure 11-38 Counter Clear Timing (Compare Match)

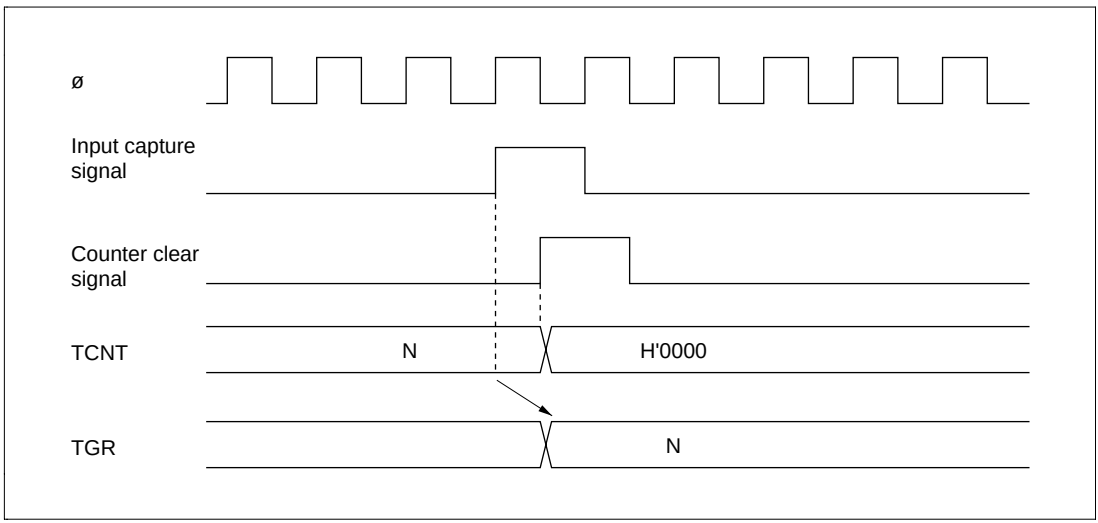


Figure 11-39 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 11-40 and 11-41 show the timing in buffer operation.

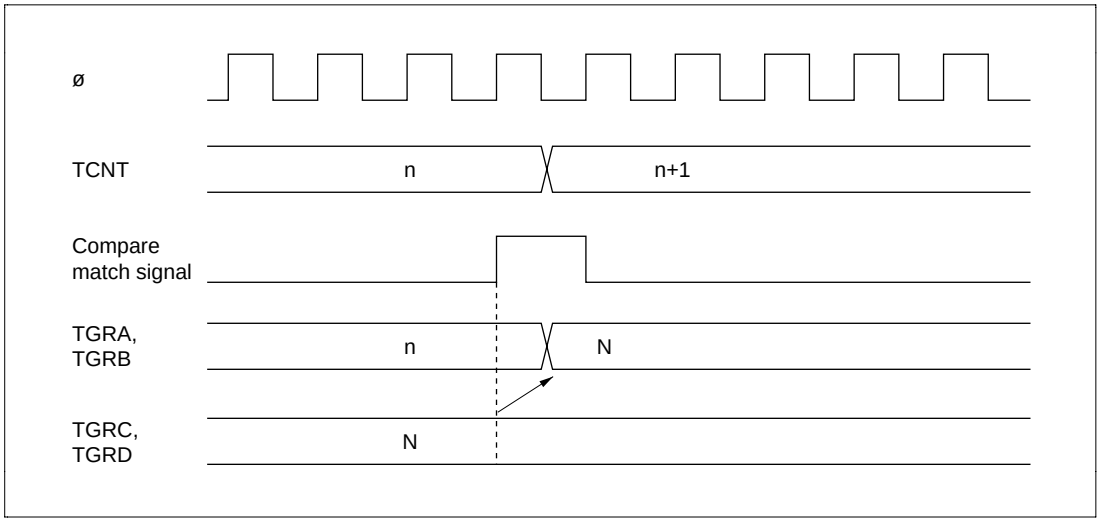


Figure 11-40 Buffer Operation Timing (Compare Match)

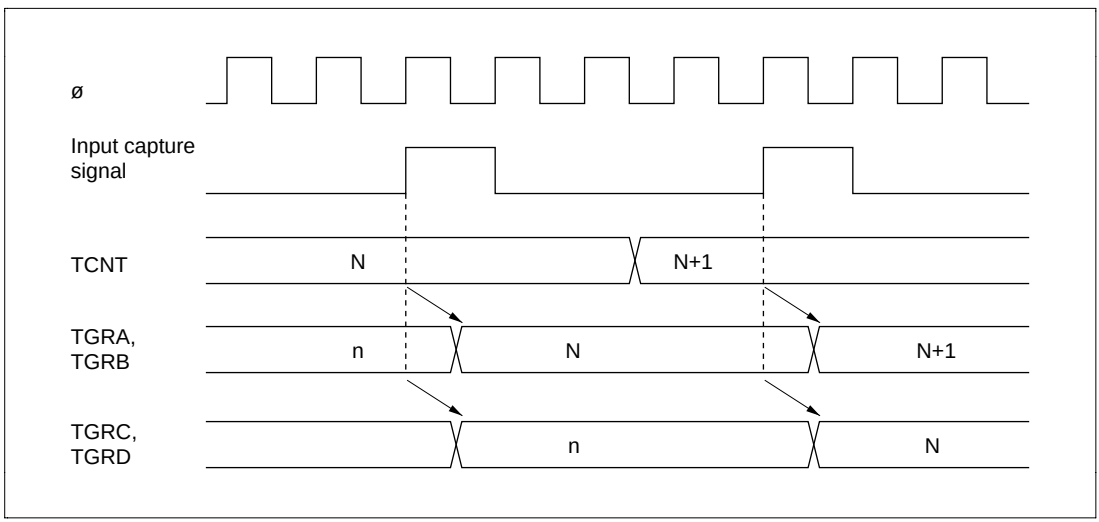


Figure 11-41 Buffer Operation Timing (Input Capture)

11.6.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 11-42 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

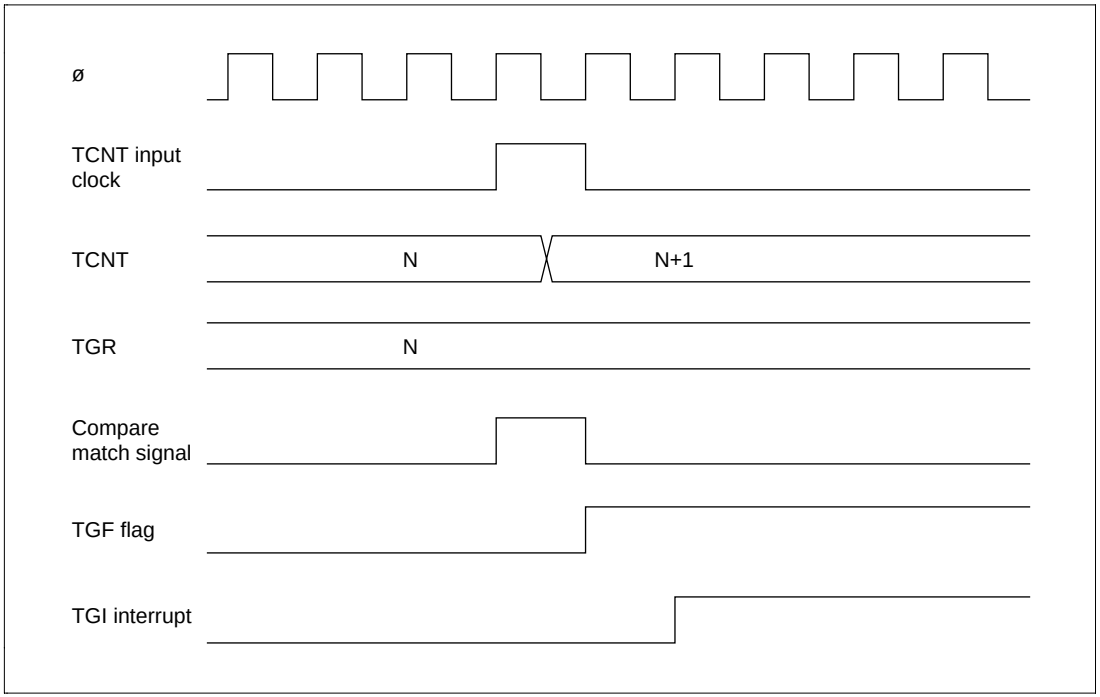


Figure 11-42 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 11-43 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

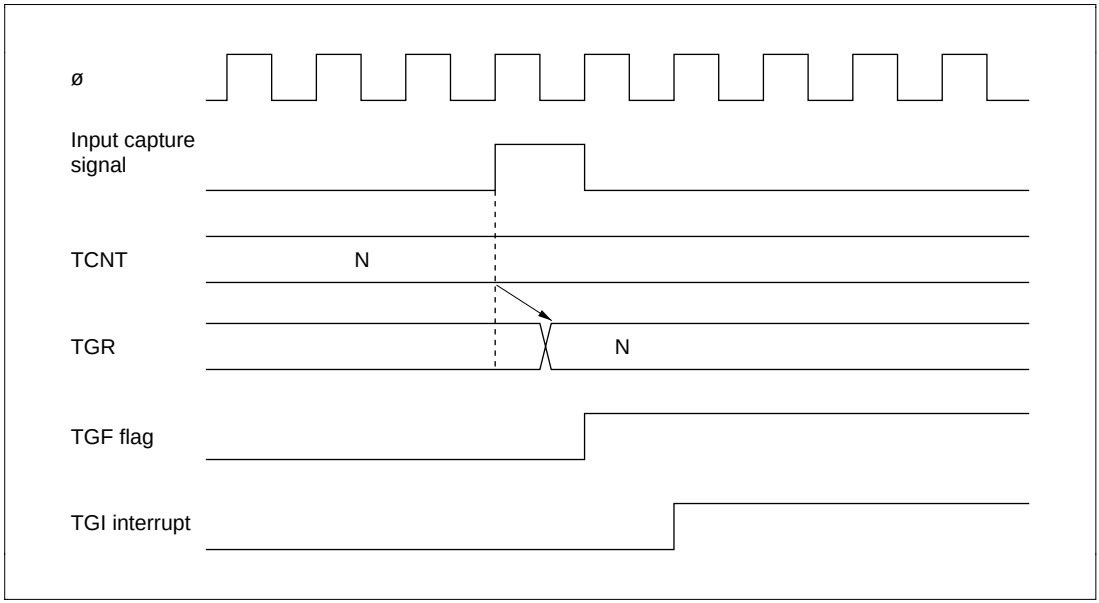


Figure 11-43 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 11-44 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing.

Figure 11-45 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

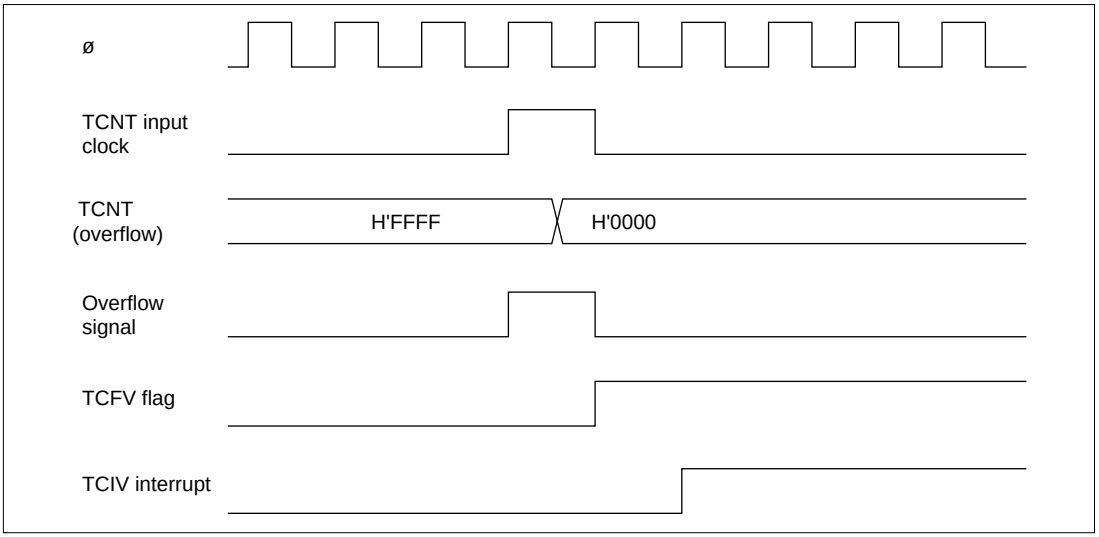


Figure 11-44 TCIV Interrupt Setting Timing

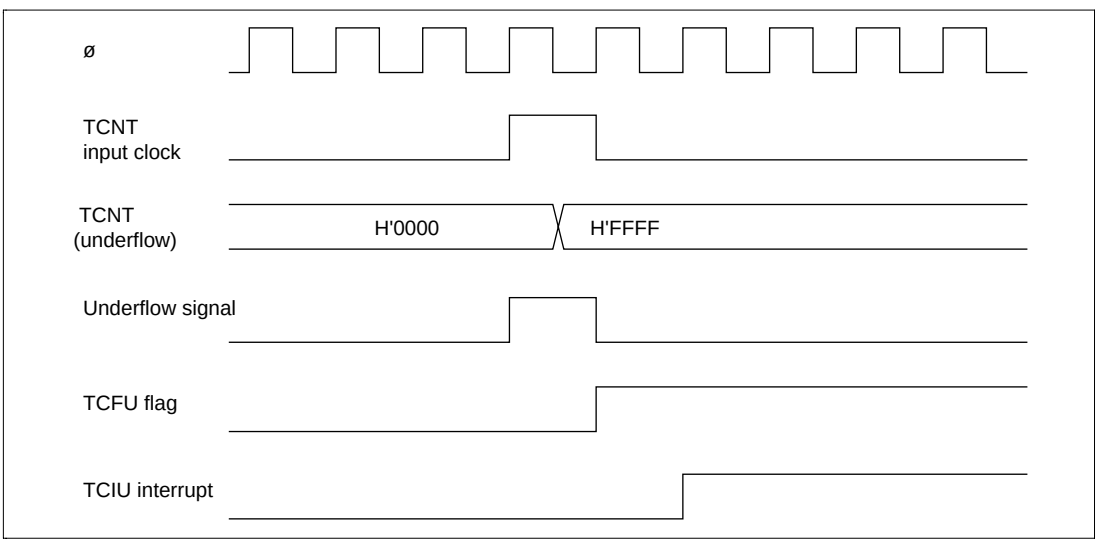


Figure 11-45 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC or DMAC is activated, the flag is cleared automatically. Figure 11-46 shows the timing for status flag clearing by the CPU, and figure 11-47 shows the timing for status flag clearing by the DTC or DMAC.

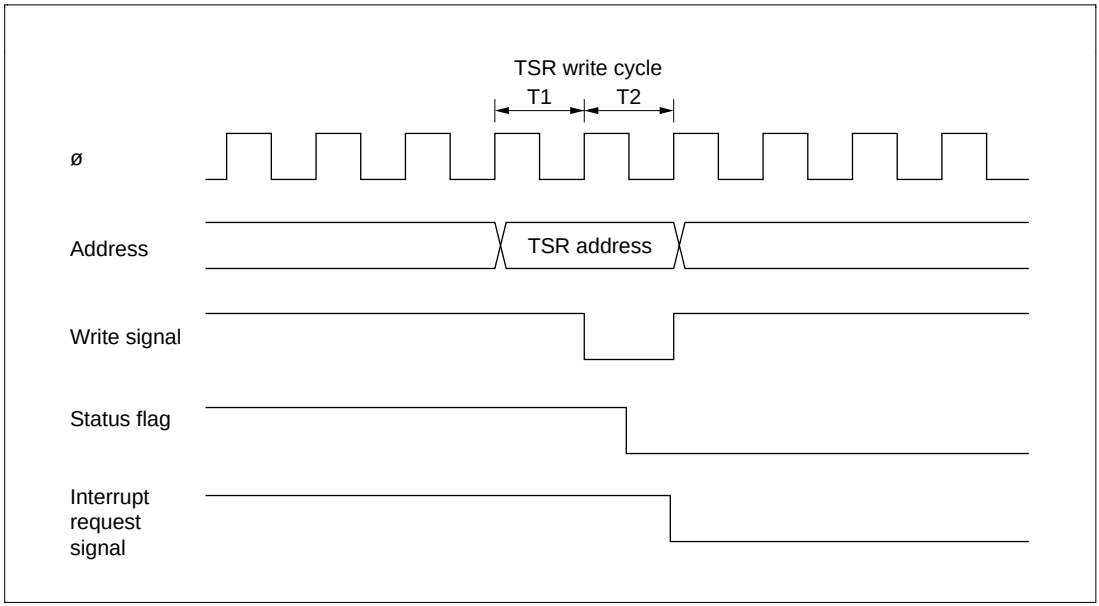


Figure 11-46 Timing for Status Flag Clearing by CPU

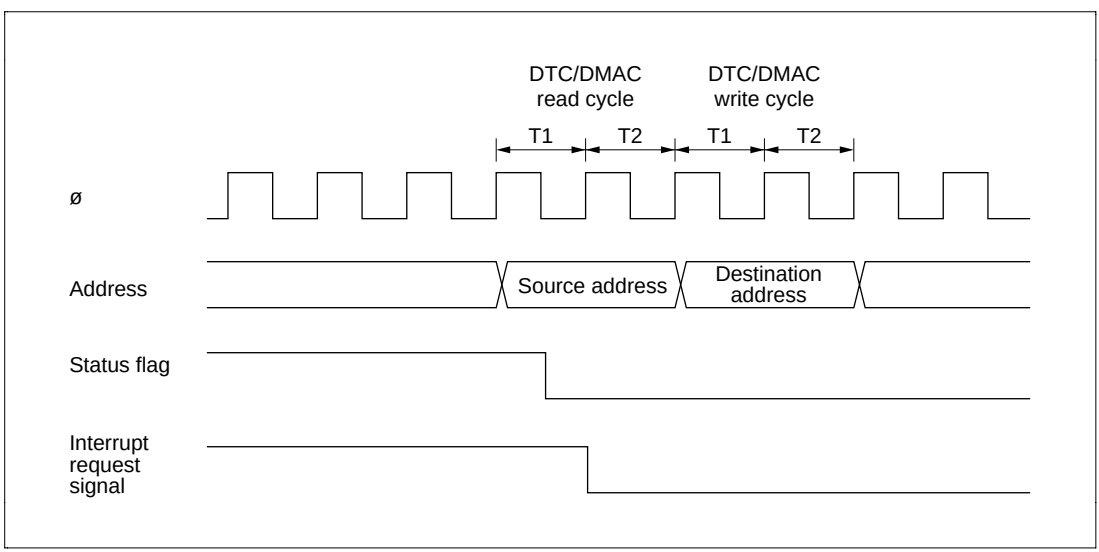


Figure 11-47 Timing for Status Flag Clearing by DTC or DMAC Activation

11.7 Usage Notes

Note that the kinds of operation and contention described below occur during TPU operation.

Input Clock Restrictions: The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 11-48 shows the input clock conditions in phase counting mode.

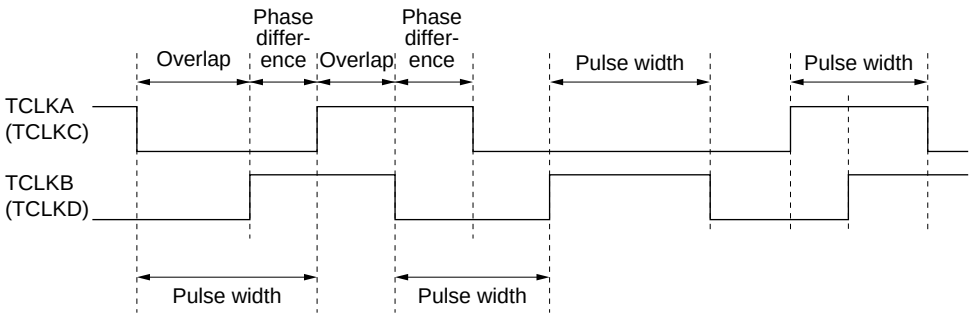


Figure 11-48 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting: When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{ø}{(N + 1)}$$

Where f : Counter frequency
ø : Operating frequency
N : TGR set value

Contention between TCNT Write and Clear Operations: If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 11-49 shows the timing in this case.

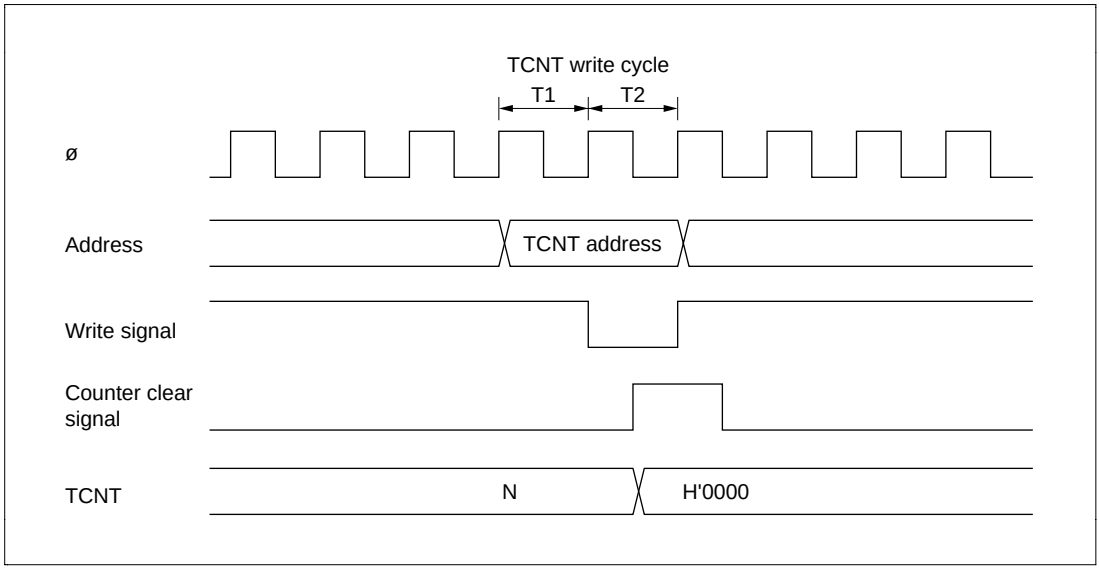


Figure 11-49 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations: If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 11-50 shows the timing in this case.

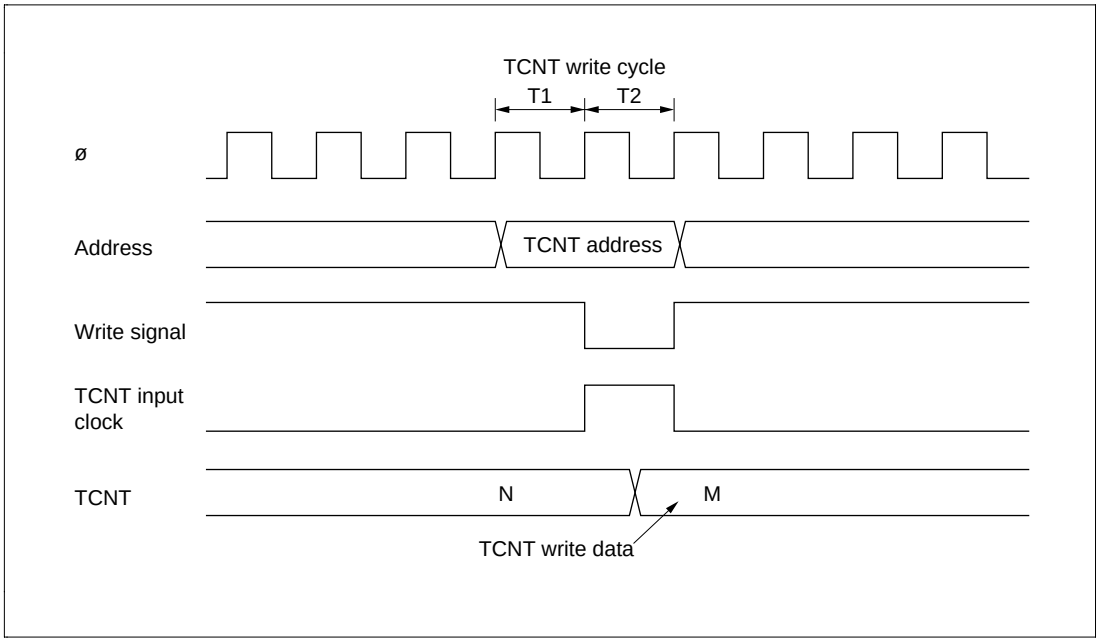


Figure 11-50 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match: If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written.

Figure 11-51 shows the timing in this case.

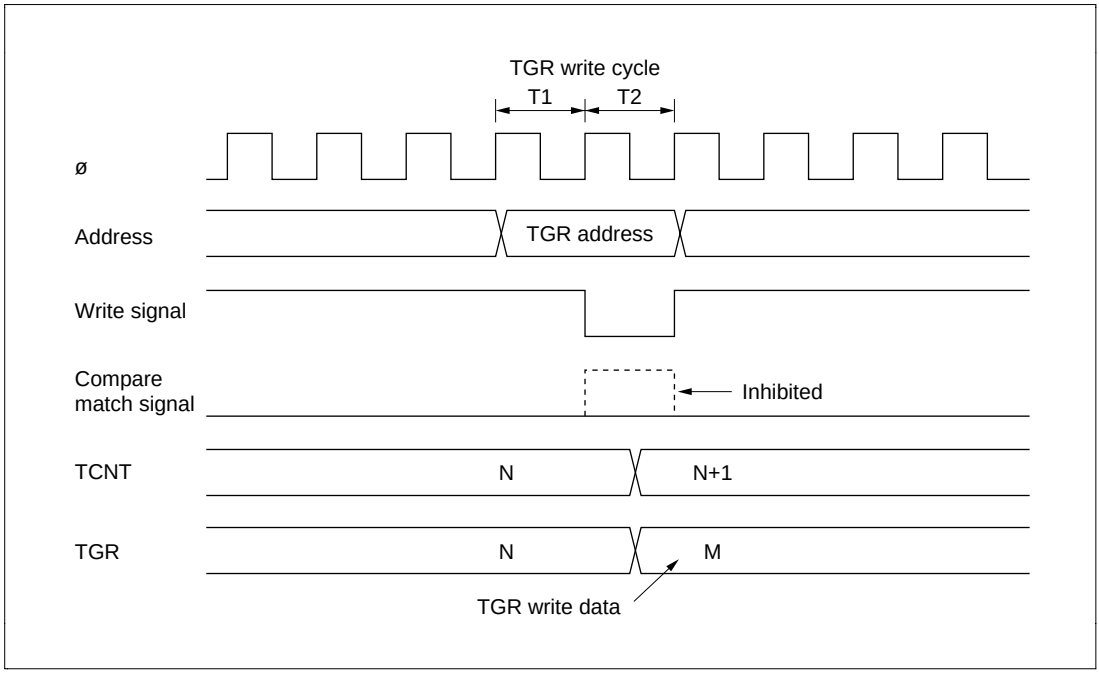


Figure 11-51 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match: If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

Figure 11-52 shows the timing in this case.

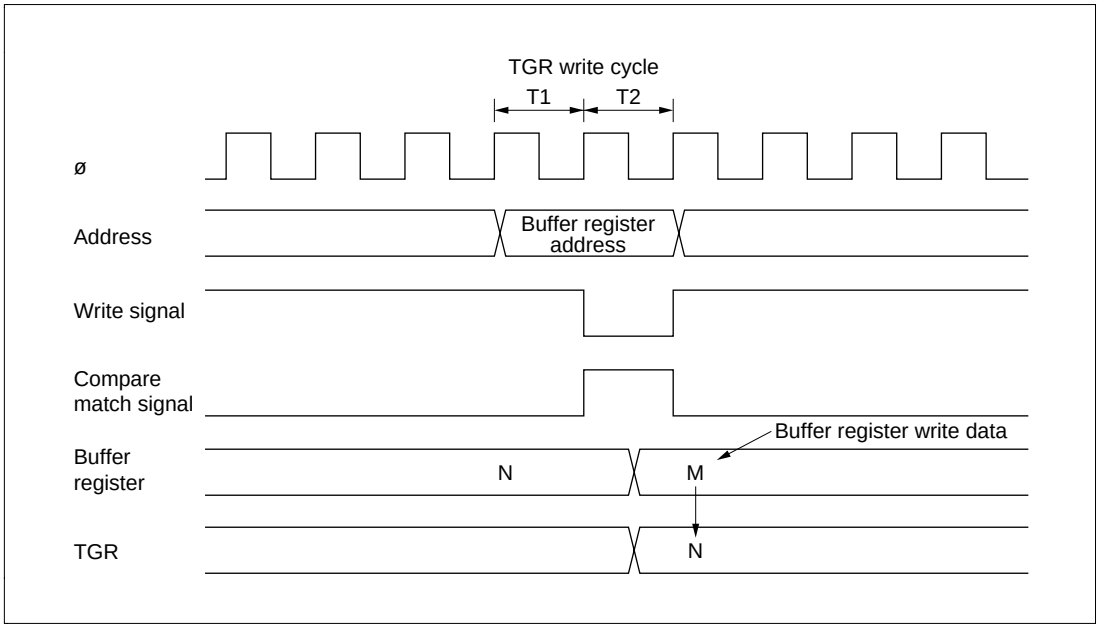


Figure 11-52 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture: If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 11-53 shows the timing in this case.

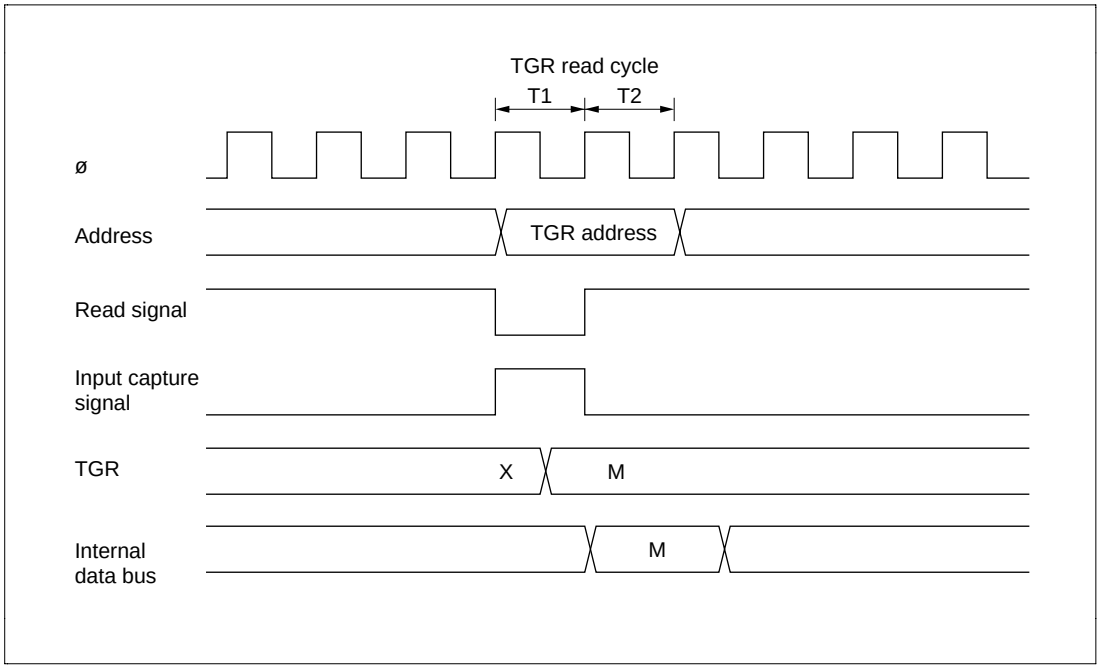


Figure 11-53 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture: If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 11-54 shows the timing in this case.

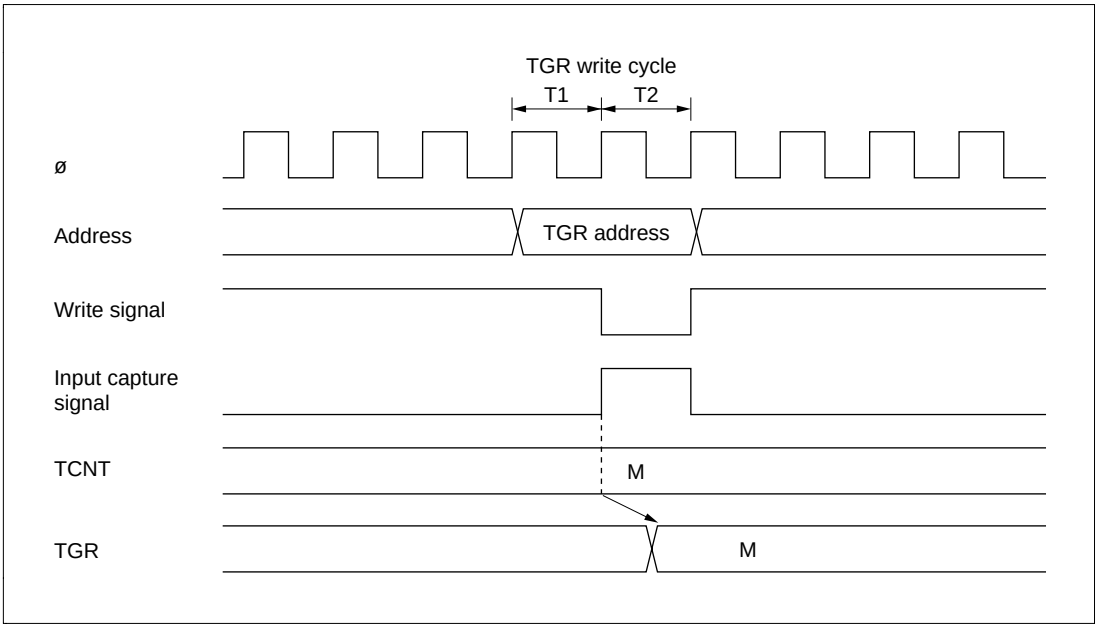


Figure 11-54 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture: If the input capture signal is generated in the T2 state of a buffer write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 11-55 shows the timing in this case.

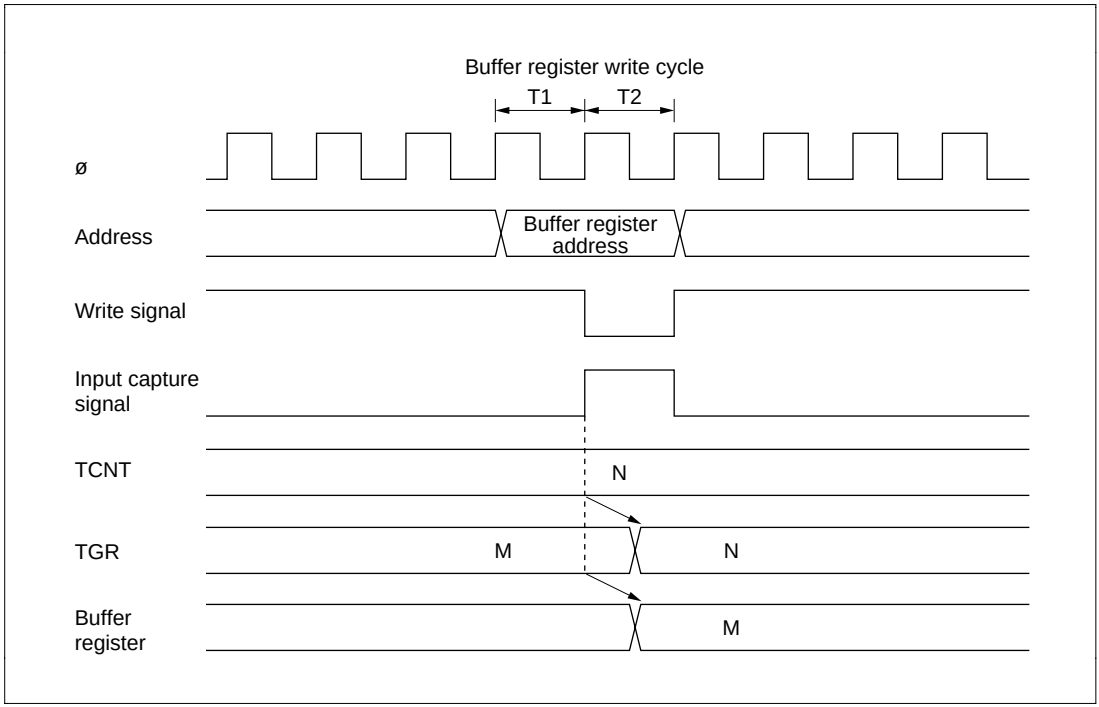


Figure 11-55 Contention between Buffer Register Write and Input Capture

Contention between Overflow/Underflow and Counter Clearing: If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 11-56 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

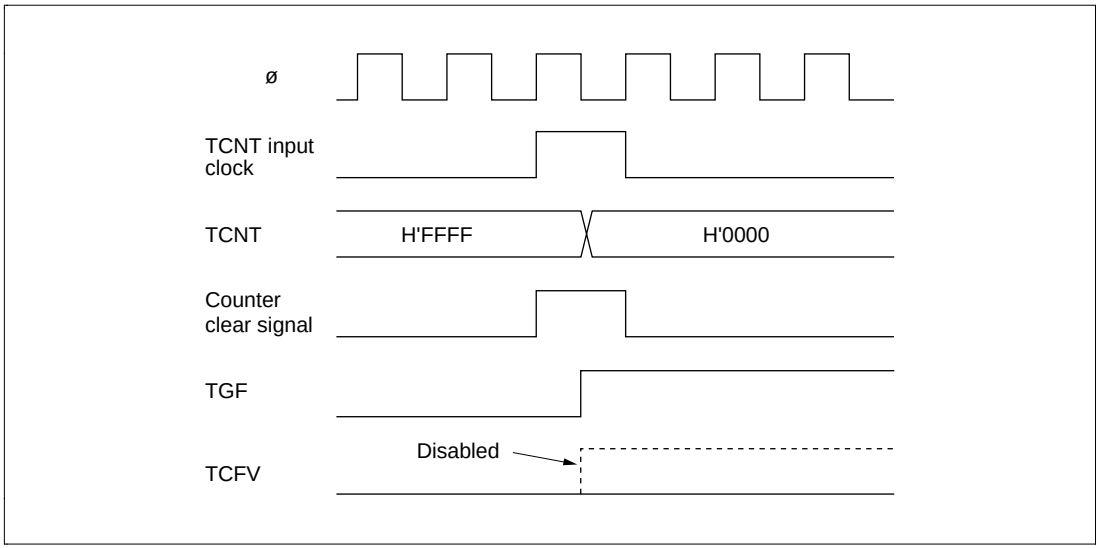


Figure 11-56 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow: If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 11-57 shows the operation timing when there is contention between TCNT write and overflow.

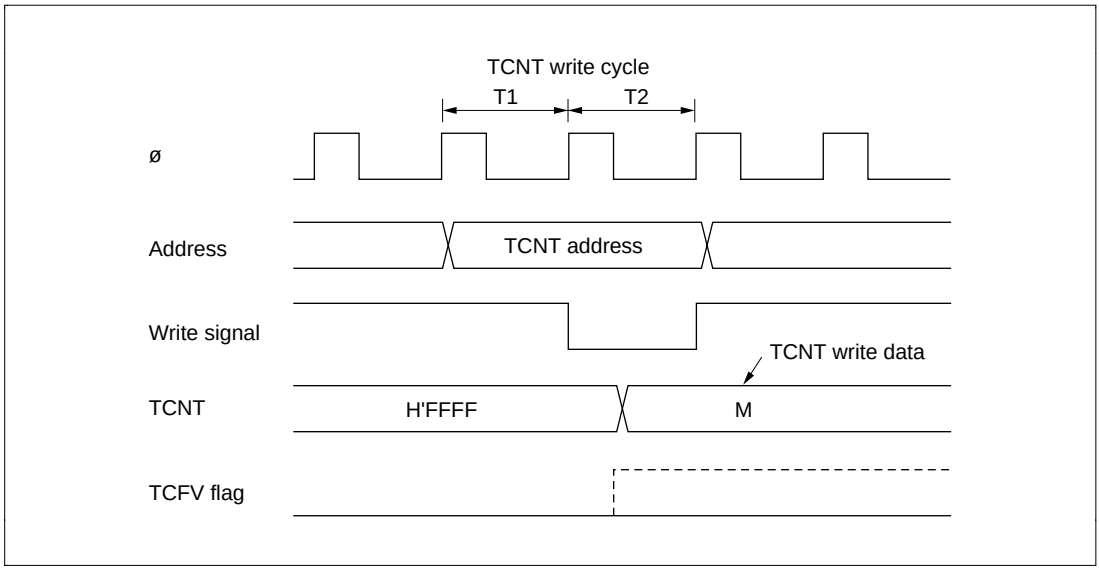


Figure 11-57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the H8S/2633 Series, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

Interrupts and Module Stop Mode: If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 12 Programmable Pulse Generator (PPG)

12.1 Overview

The H8S/2633 Series has a built-in programmable pulse generator (PPG) that provides pulse outputs by using the 16-bit timer-pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 3 and group 2) that can operate both simultaneously and independently.

12.1.1 Features

PPG features are listed below.

- 8-bit output data
  - — Maximum 8-bit data can be output, and output can be enabled on a bit-by-bit basis
- Two output groups
  - — Output trigger signals can be selected in 4-bit groups to provide up to two different 4-bit outputs
- Selectable output trigger signals
  - — Output trigger signals can be selected for each group from the compare match signals of four TPU channels
- Non-overlap mode
  - — A non-overlap margin can be provided between pulse outputs
- Can operate together with the data transfer controller (DTC) and DMA controller (DMAC)
  - — The compare match signals selected as output trigger signals can activate the DTC or DMAC for sequential output of data without CPU intervention
- Settable inverted output
  - — Inverted data can be output for each group
- Module stop mode can be set
  - — As the initial setting, PPG operation is halted. Register access is enabled by exiting module stop mode

12.1.2 Block Diagram

Figure 12-1 shows a block diagram of the PPG.

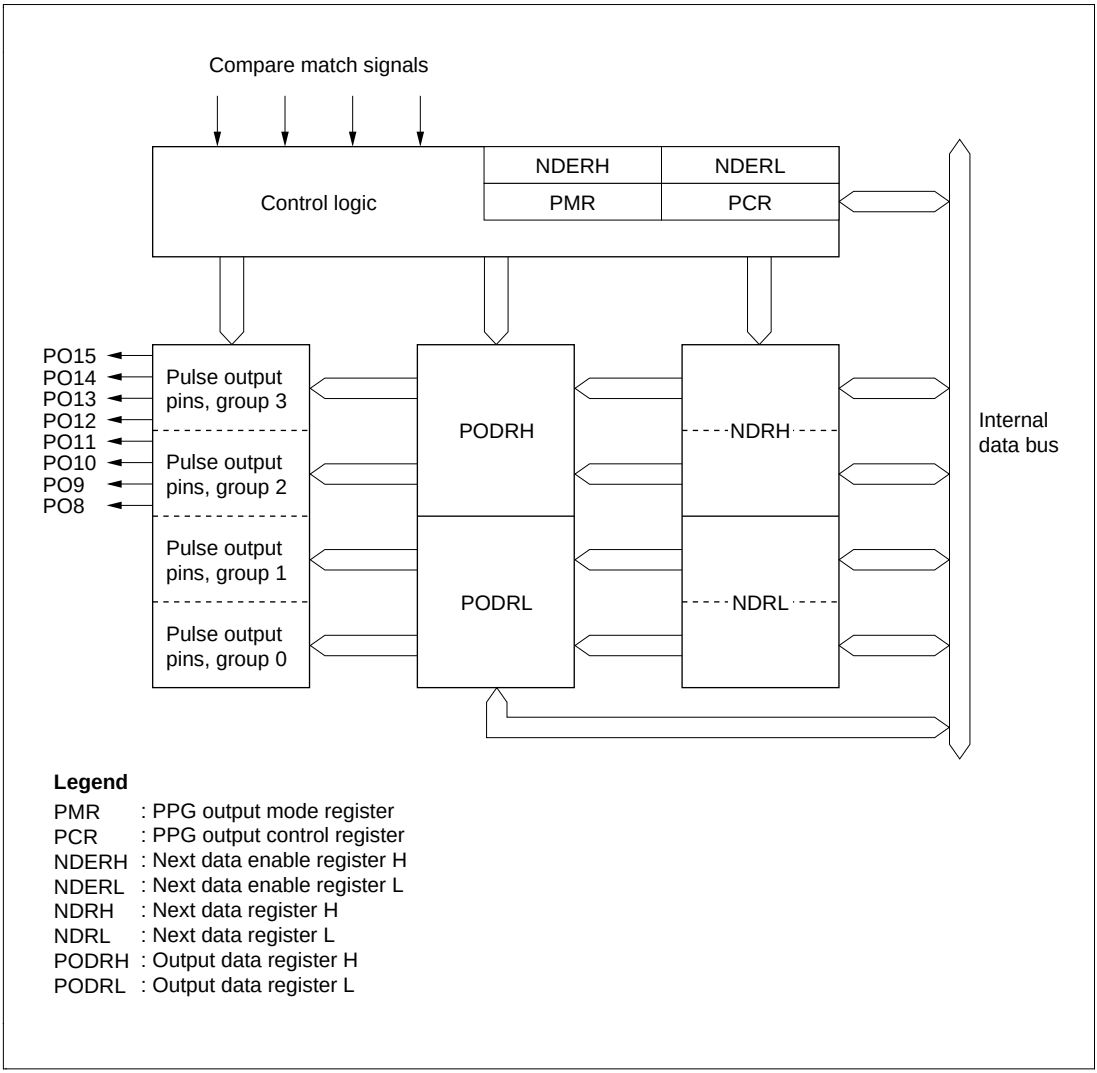


Figure 12-1 Block Diagram of PPG

12.1.3 Pin Configuration

Table 12-1 summarizes the PPG pins.

Table 12-1 PPG Pins

Name	Symbol	I/O	Function
Pulse output 8	PO8	Output	Group 2 pulse output
Pulse output 9	PO9	Output	
Pulse output 10	PO10	Output	
Pulse output 11	PO11	Output	
Pulse output 12	PO12	Output	Group 3 pulse output
Pulse output 13	PO13	Output	
Pulse output 14	PO14	Output	
Pulse output 15	PO15	Output	

12.1.4 Registers

Table 12-2 summarizes the PPG registers.

Table 12-2 PPG Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
PPG output control register	PCR	R/W	H'FF	H'FE26
PPG output mode register	PMR	R/W	H'F0	H'FE27
Next data enable register H	NDERH	R/W	H'00	H'FE28
Next data enable register L*[4]	NDERL	R/W	H'00	H'FE29
Output data register H	PODRH	R/(W)*[2]	H'00	H'FE2A
Output data register L	PODRL	R/(W)*[2]	H'00	H'FE2B
Next data register H	NDRH	R/W	H'00	H'FE2C*[3] H'FE2E
Next data register L*[4]	NDRL	R/W	H'00	H'FE2D*[3] H'FE2F
Port 1 data direction register	P1DDR	W	H'00	H'FE30
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 A bit that has been set for pulse output by NDER is read-only.

*3 When the same output trigger is selected for pulse output groups 2 and 3 by the PCR setting, the NDRH address is H'FE2C. When the output triggers are different, the NDRH address is H'FE2E for group 2 and H'FE2C for group 3.

Similarly, when the same output trigger is selected for pulse output groups 0 and 1 by the PCR setting, the NDRL address is H'FE2D. When the output triggers are different, the NDRL address is H'FE2F for group 0 and H'FE2D for group 1.

*4 The H8S/2633 Series has no pins corresponding to pulse output groups 0 and 1.

12.2 Register Descriptions

12.2.1 Next Data Enable Registers H and L (NDERH, NDERL)

NDERH

Bit	:	7	6	5	4	3	2	1	0
		NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERL

Bit	:	7	6	5	4	3	2	1	0
		NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse output on a bit-by-bit basis.

If a bit is enabled for pulse output by NDERH or NDERL, the NDR value is automatically transferred to the corresponding PODR bit when the TPU compare match event specified by PCR occurs, updating the output value. If pulse output is disabled, the bit value is not transferred from NDR to PODR and the output value does not change.

NDERH and NDERL are each initialized to H'00 by a reset and in hardware standby mode. They are not initialized in software standby mode.

NDERH Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable pulse output on a bit-by-bit basis.

Bits 7 to 0

NDER15 to NDER8	Description
0	Pulse outputs PO15 to PO8 are disabled (NDR15 to NDR8 are not transferred to POD15 to POD8) (Initial value)
1	Pulse outputs PO15 to PO8 are enabled (NDR15 to NDR8 are transferred to POD15 to POD8)

NDERL Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable pulse output on a bit-by-bit basis.

Bits 7 to 0

NDER7 to NDER0	Description
0	Pulse outputs PO7 to PO0 are disabled (NDR7 to NDR0 are not transferred to POD7 to POD0) (Initial value)
1	Pulse outputs PO7 to PO0 are enabled (NDR7 to NDR0 are transferred to POD7 to POD0)

12.2.2 Output Data Registers H and L (PODRH, PODRL)

PODRH

Bit	:	7	6	5	4	3	2	1	0
		POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

PODRL

Bit	:	7	6	5	4	3	2	1	0
		POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

PODRH and PODRL are 8-bit readable/writable registers that store output data for use in pulse output. However, the H8S/2633 Series has no pins corresponding to PODRL.

12.2.3 Next Data Registers H and L (NDRH, NDRL)

NDRH and NDRL are 8-bit readable/writable registers that store the next data for pulse output. During pulse output, the contents of NDRH and NDRL are transferred to the corresponding bits in PODRH and PODRL when the TPU compare match event specified by PCR occurs. The NDRH and NDRL addresses differ depending on whether pulse output groups have the same output trigger or different output triggers. For details see section 12.2.4, Notes on NDR Access.

NDRH and NDRL are each initialized to H'00 by a reset and in hardware standby mode. They are not initialized in software standby mode.

12.2.4 Notes on NDR Access

The NDRH and NDRL addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

Same Trigger for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by the same compare match event, the NDRH address is H'FE2C. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FE2E consists entirely of reserved bits that cannot be modified and are always read as 1.

Address H'FE2C

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2E

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—	—

If pulse output groups 0 and 1 are triggered by the same compare match event, the NDRL address is H'FE2D. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FE2F consists entirely of reserved bits that cannot be modified and are always read as 1. However, the H8S/2633 Series has no output pins corresponding to pulse output groups 0 and 1.

Address H'FE2D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits in NDRH (group 3) is H'FE2C and the address of the lower 4 bits (group 2) is H'FE2E. Bits 3 to 0 of address H'FE2C and bits 7 to 4 of address H'FE2E are reserved bits that cannot be modified and are always read as 1.

Address H'FE2C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the address of the upper 4 bits in NDRL (group 1) is H'FE2D and the address of the lower 4 bits (group 0) is H'FE2F. Bits 3 to 0 of address H'FE2D and bits 7 to 4 of address H'FE2F are reserved bits that cannot be modified and are always read as 1. However, the H8S/2633 Series has no output pins corresponding to pulse output groups 0 and 1.

Address H'FE2D

Bit	:	7	6	5	4	3	2	1	0
		NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2F

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

12.2.5 PPG Output Control Register (PCR)

Bit	:	7	6	5	4	3	2	1	0
		G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	:	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

PCR is an 8-bit readable/writable register that selects output trigger signals for PPG outputs on a group-by-group basis.

PCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare match that triggers pulse output group 3 (pins PO15 to PO12).

Bit 7	Bit 6	Description
G3CMS1	**G3CMS0**	**Output Trigger for Pulse Output Group 3**
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3 (Initial value)

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare match that triggers pulse output group 2 (pins PO11 to PO8).

Bit 5	Bit 4	Description	
G2CMS1	**G2CMS0**	**Output Trigger for Pulse Output Group 2**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match that triggers pulse output group 1 (pins PO7 to PO4). However, the H8S/2633 Series has no output pins corresponding to pulse output group 1.

Bit 3	Bit 2	Description	
G1CMS1	**G1CMS0**	**Output Trigger for Pulse Output Group 1**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match that triggers pulse output group 0 (pins PO3 to PO0). However, the H8S/2633 Series has no output pins corresponding to pulse output group 0.

Bit 1	Bit 0	Description	
G0CMS1	**G0CMS0**	**Output Trigger for Pulse Output Group 0**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

12.2.6 PPG Output Mode Register (PMR)

Bit	:	7	6	5	4	3	2	1	0
		G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR is an 8-bit readable/writable register that selects pulse output inversion and non-overlapping operation for each group.

The output trigger period of a non-overlapping operation PPG output waveform is set in TGRB and the non-overlap margin is set in TGRA. The output values change at compare match A and B.

For details, see section 12.3.4, Non-Overlapping Pulse Output.

PMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Group 3 Inversion (G3INV): Selects direct output or inverted output for pulse output group 3 (pins PO15 to PO12).

Bit 7

G3INV	Description
0	Inverted output for pulse output group 3 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 6—Group 2 Inversion (G2INV): Selects direct output or inverted output for pulse output group 2 (pins PO11 to PO8).

Bit 6

G2INV	Description
0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 5—Group 1 Inversion (G1INV): Selects direct output or inverted output for pulse output group 1 (pins PO7 to PO4). However, the H8S/2633 Series has no pins corresponding to pulse output group 1.

Bit 5

G1INV	Description
0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 4—Group 0 Inversion (G0INV): Selects direct output or inverted output for pulse output group 0 (pins PO3 to PO0). However, the H8S/2633 Series has no pins corresponding to pulse output group 0.

Bit 4

G0INV	Description
0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping operation for pulse output group 3 (pins PO15 to PO12).

Bit 3

G3NOV	Description
0	Normal operation in pulse output group 3 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 3 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping operation for pulse output group 2 (pins PO11 to PO8).

Bit 2

G2NOV	Description
0	Normal operation in pulse output group 2 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping operation for pulse output group 1 (pins PO7 to PO4). However, the H8S/2633 Series has no pins corresponding to pulse output group 1.

Bit 1

G1NOV	Description
0	Normal operation in pulse output group 1 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping operation for pulse output group 0 (pins PO3 to PO0). However, the H8S/2633 Series has no pins corresponding to pulse output group 0.

Bit 0

G0NOV	Description
0	Normal operation in pulse output group 0 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

12.2.7 Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1.

Port 1 is multiplexed with pins PO15 to PO8. Bits corresponding to pins used for PPG output must be set to 1. For further information about P1DDR, see section 10.2, Port 1.

12.2.8 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA3 bit in MSTPCRA is set to 1, PPG operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 3—Module Stop (MSTPA3): Specifies the PPG module stop mode.

Bit 3		
MSTPA3	**Description**	
0	PPG module stop mode cleared	
1	PPG module stop mode set	(Initial value)

12.3 Operation

12.3.1 Overview

PPG pulse output is enabled when the corresponding bits in P1DDR and NDER are set to 1. In this state the corresponding PODR contents are output.

When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values.

Figure 12-2 illustrates the PPG output operation and table 12-3 summarizes the PPG operating conditions.

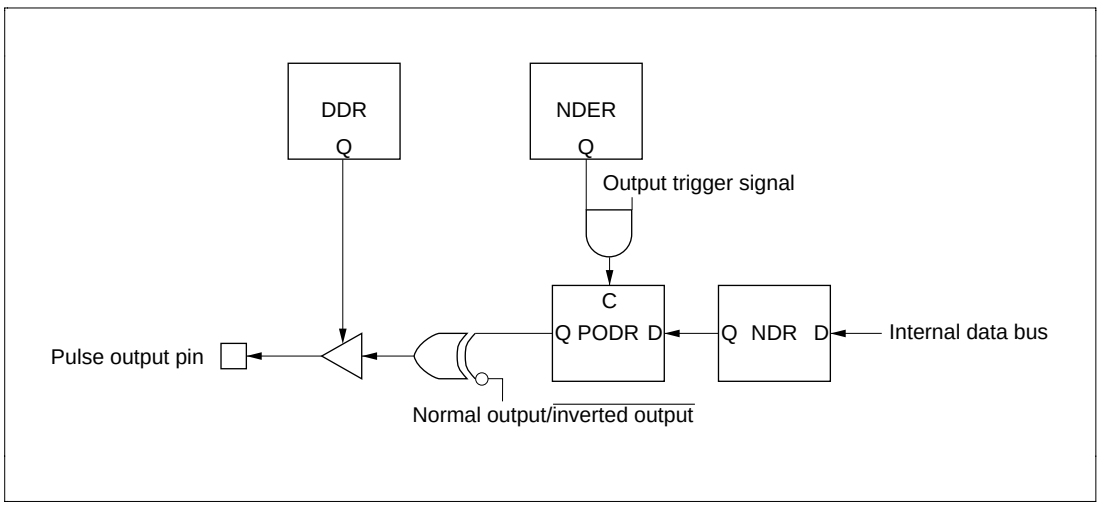


Figure 12-2 PPG Output Operation

Table 12-3 PPG Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the PODR bit is a read-only bit, and when compare match occurs, the NDR bit value is transferred to the PODR bit)
	1	PPG pulse output

Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match. For details of non-overlapping operation, see section 12.3.4, Non-Overlapping Pulse Output.

12.3.2 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 12-3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

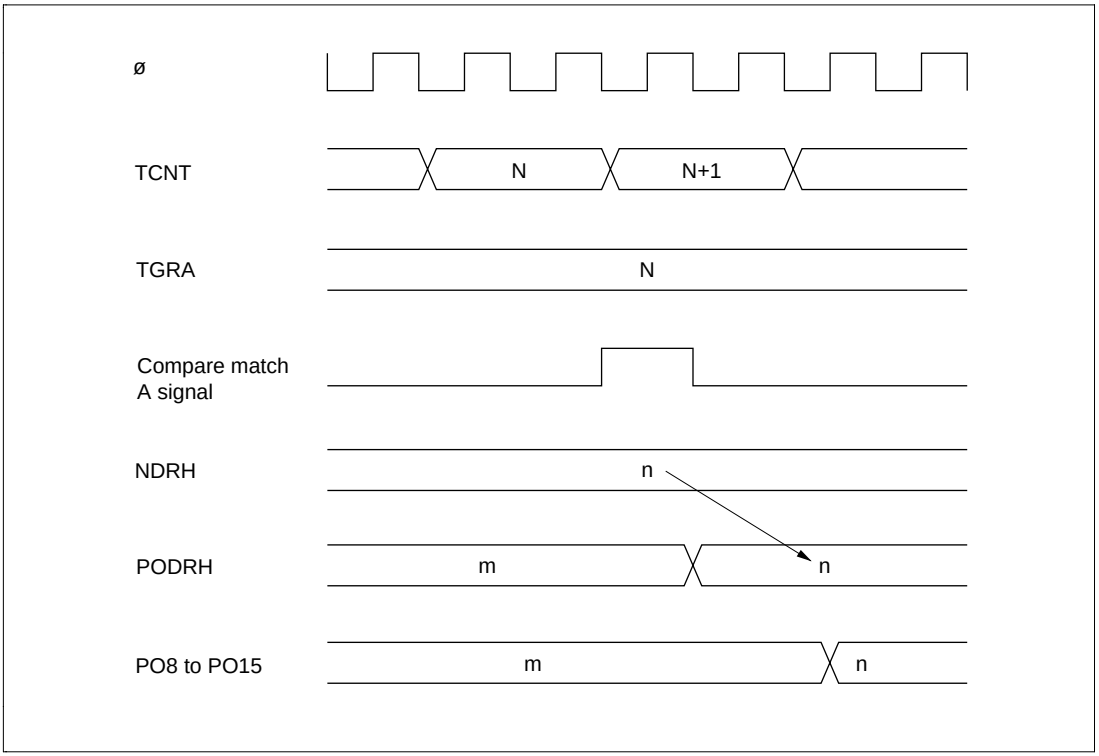
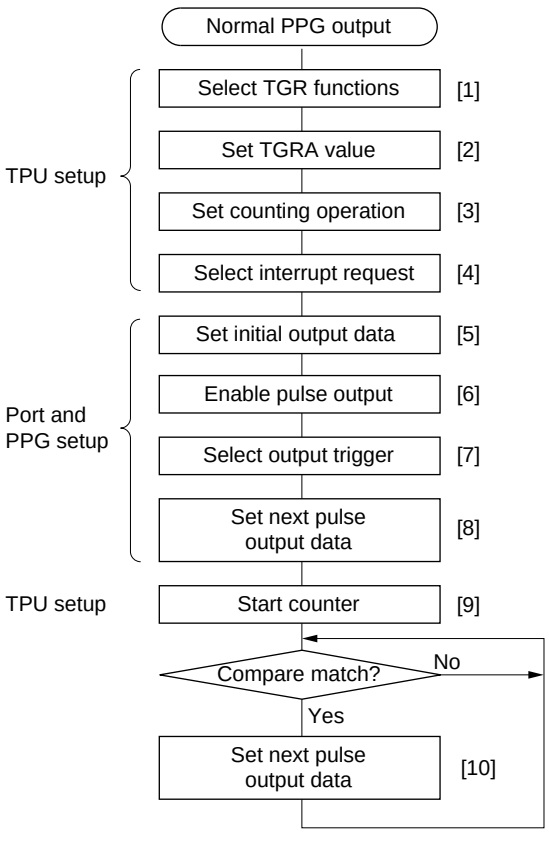


Figure 12-3 Timing of Transfer and Output of NDR Contents (Example)

12.3.3 Normal Pulse Output

Sample Setup Procedure for Normal Pulse Output: Figure 12-4 shows a sample procedure for setting up normal pulse output.



[1] Set TIOR to make TGRA an output compare register (with output disabled)

[2] Set the PPG output trigger period

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the output trigger in PCR.

[8] Set the next pulse output values in NDR.

[9] Set the CST bit in TSTR to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 12-4 Setup Procedure for Normal Pulse Output (Example)

Example of Normal Pulse Output (Example of Five-Phase Pulse Output): Figure 12-5 shows an example in which pulse output is used for cyclic five-phase pulse output.

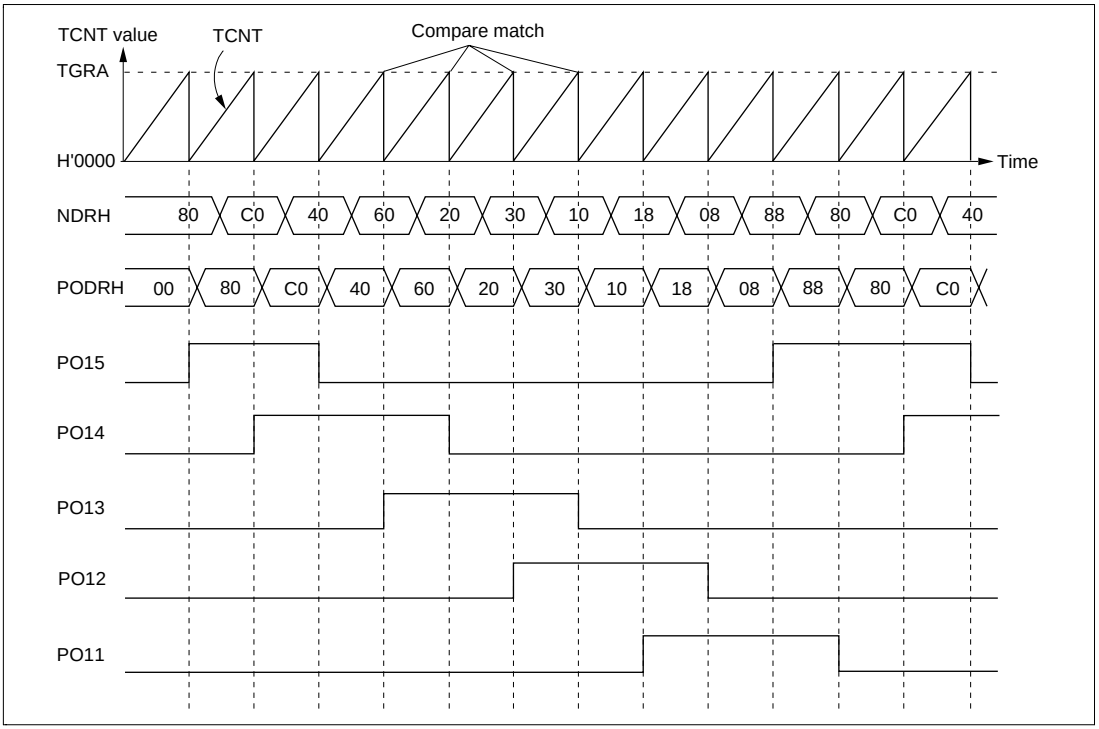


Figure 12-5 Normal Pulse Output Example (Five-Phase Pulse Output)

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA is an output compare register and the counter will be cleared by compare match A. Set the trigger period in TGRA and set the TGIEA bit in TIER to 1 to enable the compare match A (TGIA) interrupt.

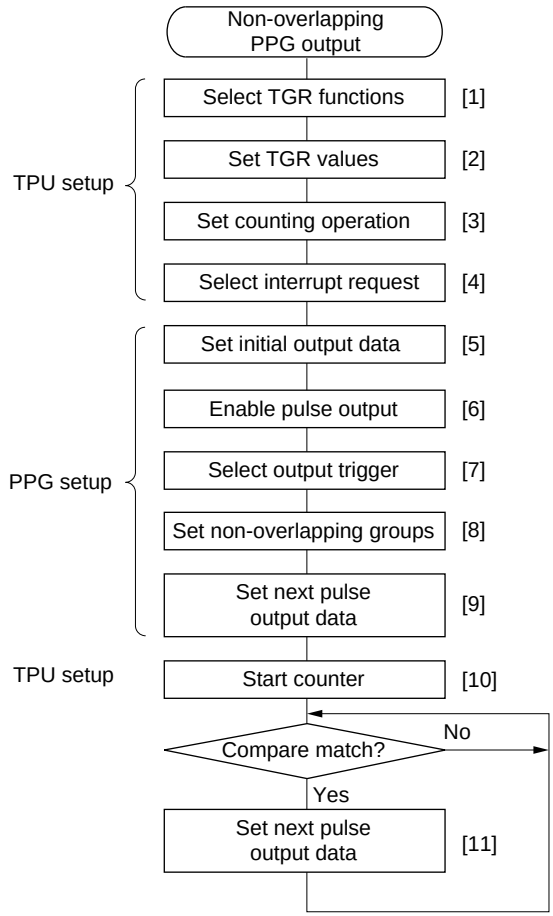
[2] Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.

[3] The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.

[4] Five-phase overlapping pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88... at successive TGIA interrupts. If the DTC or DMAC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

12.3.4 Non-Overlapping Pulse Output

Sample Setup Procedure for Non-Overlapping Pulse Output: Figure 12-6 shows a sample procedure for setting up non-overlapping pulse output.



[1] Set TIOR to make TGRA and TGRB an output compare registers (with output disabled)

[2] Set the pulse output trigger period in TGRB and the non-overlap margin in TGRA.

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlap mode.

[9] Set the next pulse output values in NDR.

[10] Set the CST bit in TSTR to 1 to start the TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 12-6 Setup Procedure for Non-Overlapping Pulse Output (Example)

Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 12-7 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

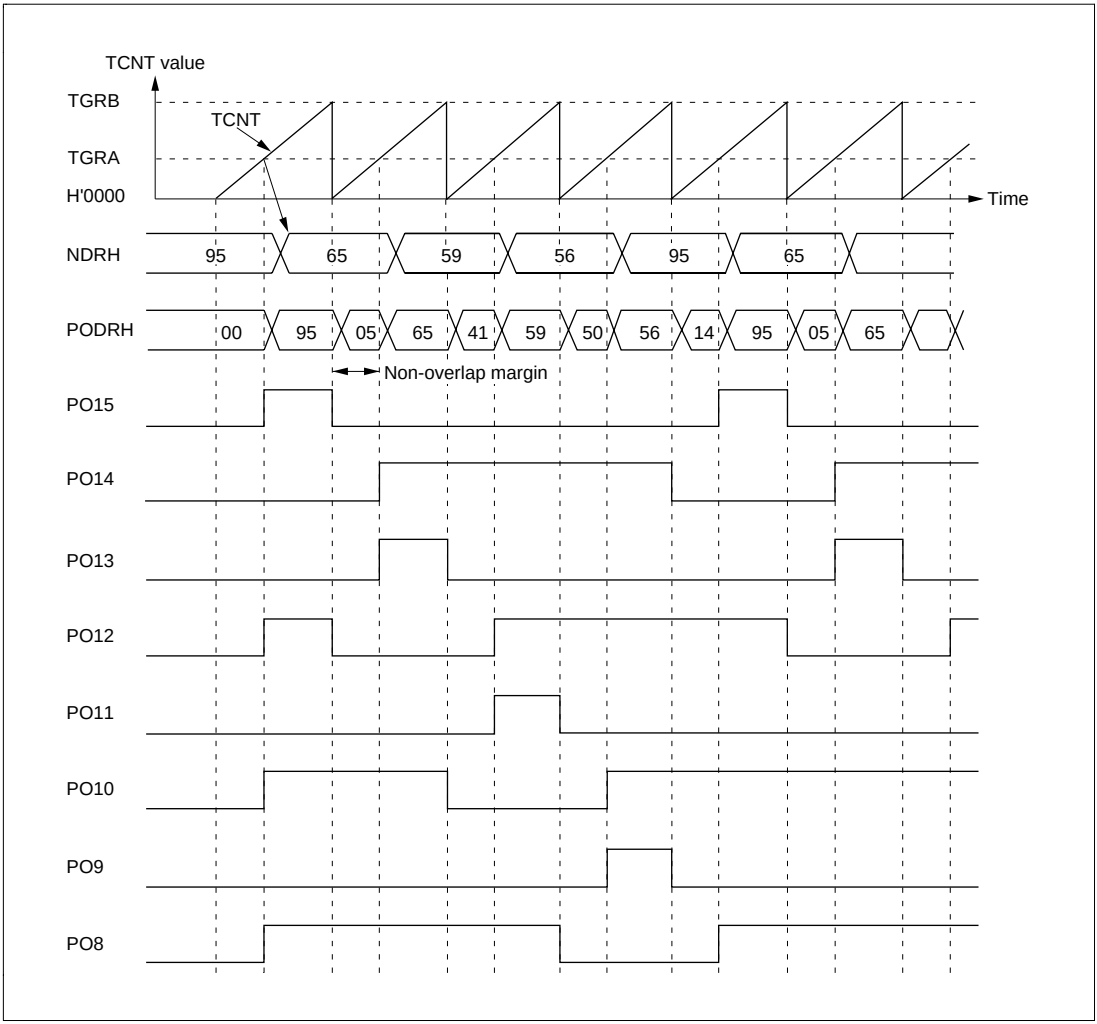


Figure 12-7 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

[2] Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

[3] The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

[4] Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts. If the DTC or DMAC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

12.3.5 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 12-8 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 12-7.

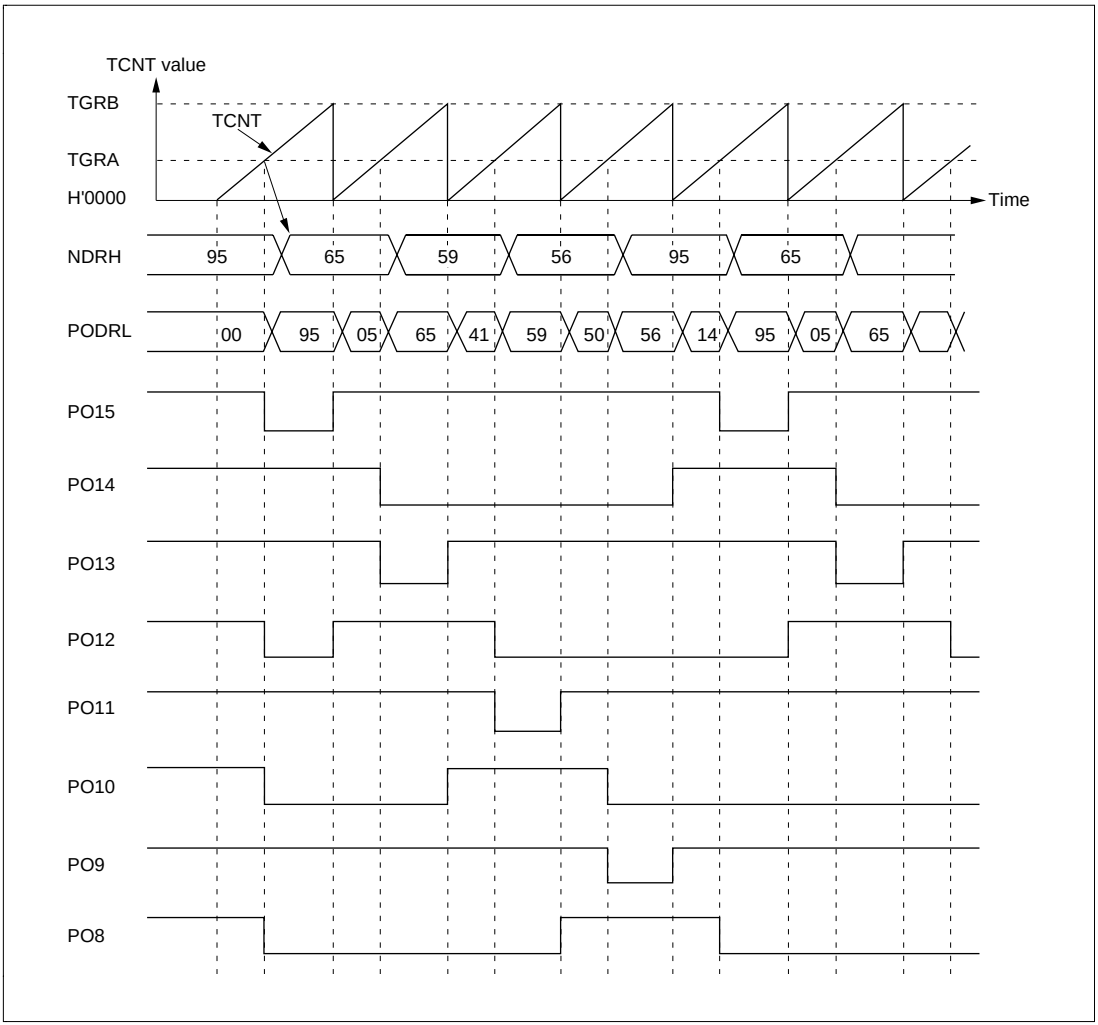


Figure 12-8 Inverted Pulse Output (Example)

12.3.6 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 12-9 shows the timing of this output.

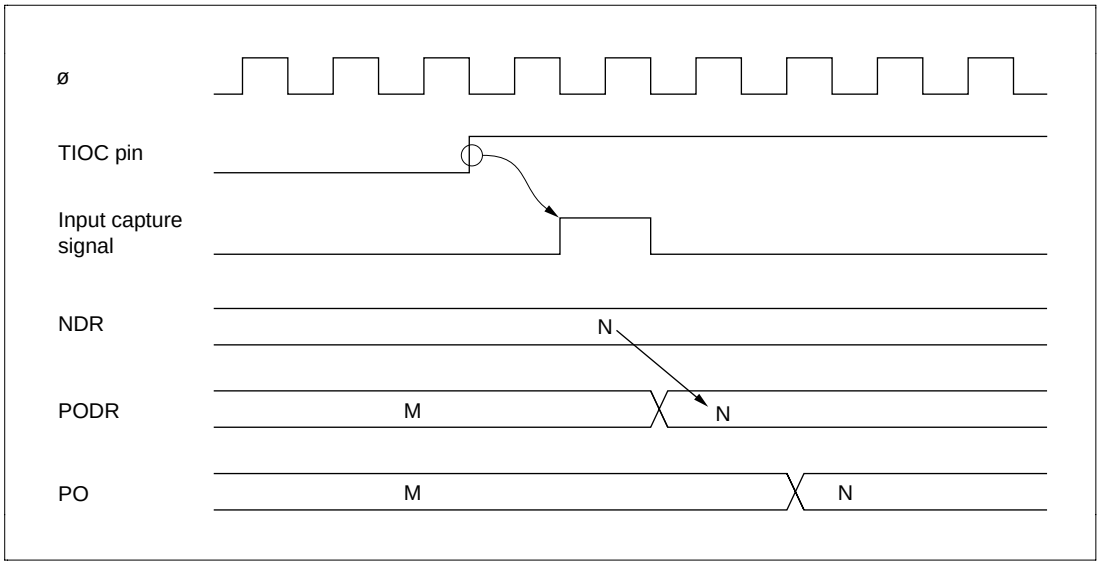


Figure 12-9 Pulse Output Triggered by Input Capture (Example)

12.4 Usage Notes

Operation of Pulse Output Pins: Pins PO8 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Note on Non-Overlapping Output: During non-overlapping operation, the transfer of NDR bit values to PODR bits takes place as follows.

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 12-10 illustrates the non-overlapping pulse output operation.

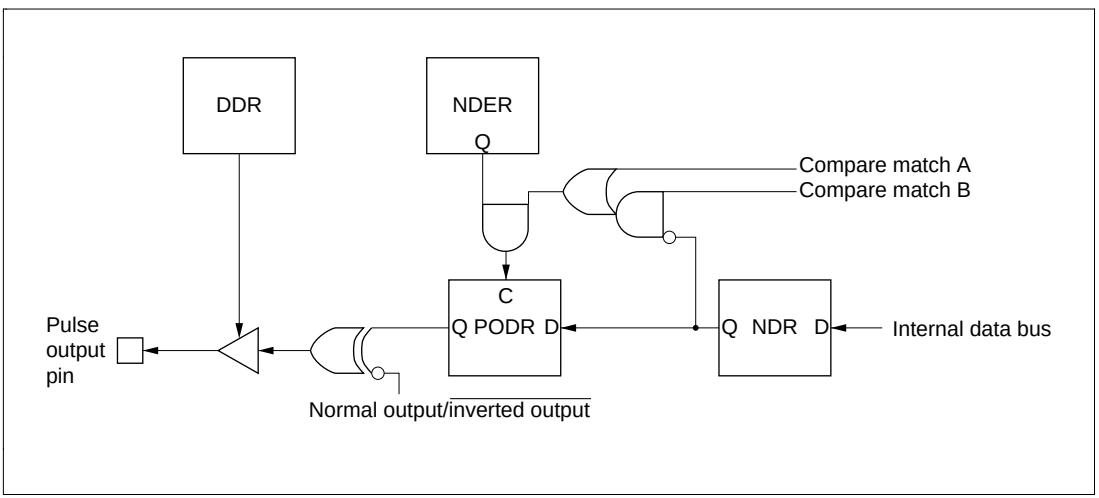


Figure 12-10 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 12-11 shows the timing of this operation.

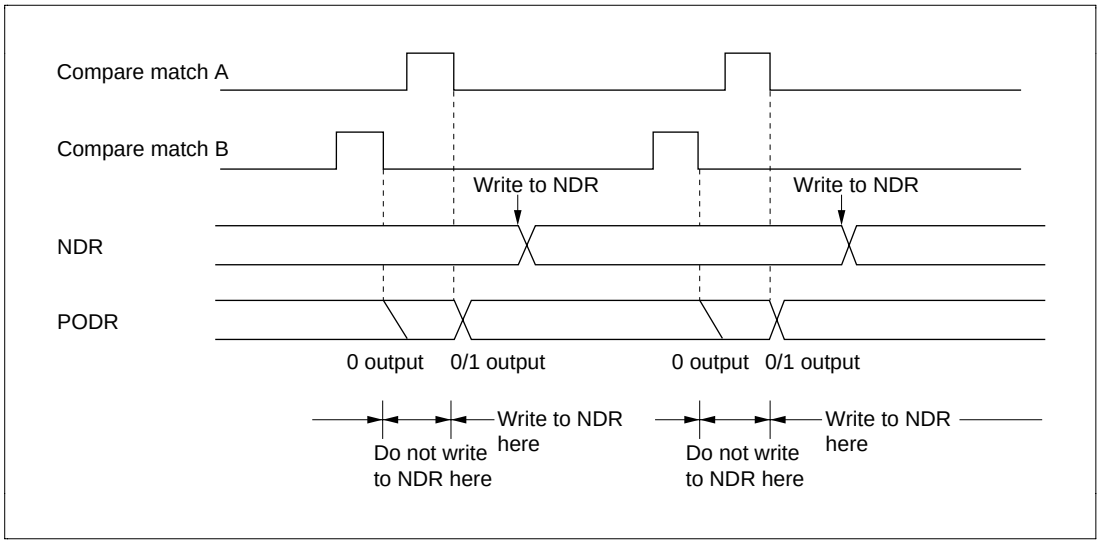


Figure 12-11 Non-Overlapping Operation and NDR Write Timing

Section 13 8-Bit Timers (TMR)

13.1 Overview

The H8S/2633 Series includes an 8-bit timer module with four channels (TMR0, TMR1, TMR2, and TMR3). Each channel has an 8-bit counter (TCNT) and two time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare match events. The 8-bit timer module can thus be used for a variety of functions, including pulse output with an arbitrary duty cycle.

13.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources
  - The counters can be driven by one of three internal clock signals (ø/8, ø/64, or ø/8192) or an external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters
  - The counters can be cleared on compare match A or B, or by an external reset signal.
- Timer output control by a combination of two compare match signals
  - The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output.
- Provision for cascading of two channels
  - Operation as a 16-bit timer is possible, using channel 0 (channel 2) for the upper 8 bits and channel 1 (channel 3) for the lower 8 bits (16-bit count mode).
  - Channel 1 (channel 3) can be used to count channel 0 (channel 2) compare matches (compare match count mode).
- Three independent interrupts
  - Compare match A and B and overflow interrupts can be requested independently.
- A/D converter conversion start trigger can be generated
  - Channel 0 compare match A signal can be used as an A/D converter conversion start trigger.
- Module stop mode can be set
  - As the initial setting, 8-bit timer operation is halted. Register access is enabled by exiting module stop mode.

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the 8-bit timer module (TMR0, TMR1).

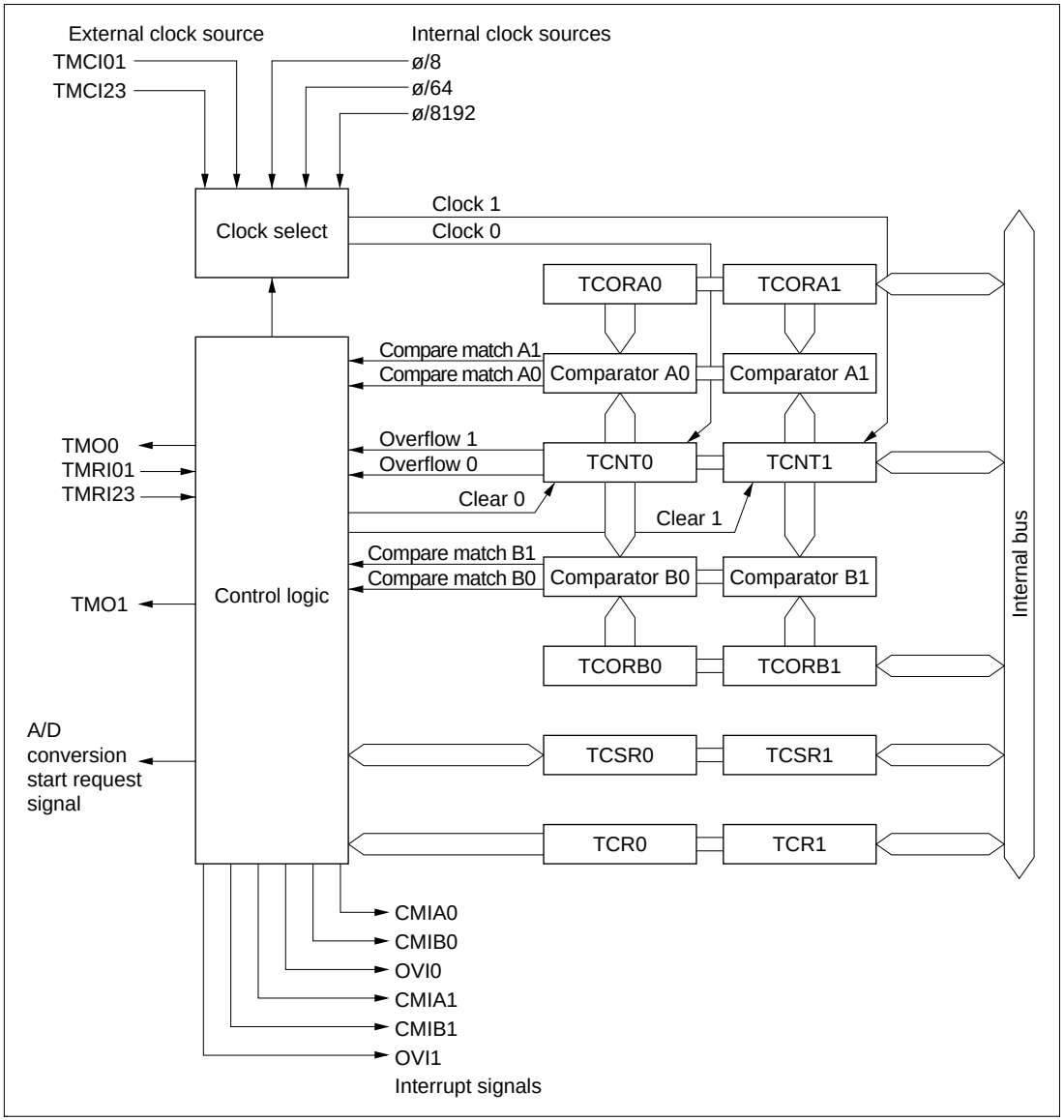


Figure 13-1 Block Diagram of 8-Bit Timer

13.1.3 Pin Configuration

Table 13-1 summarizes the input and output pins of the 8-bit timer.

Table 13-1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin 0	TMO0	Output	Outputs at compare match
	Timer clock input pin 01	TMCI01	Input	Inputs external clock for counter
	Timer reset input pin 01	TMRI01	Input	Inputs external reset to counter
1	Timer output pin 1	TMO1	Output	Outputs at compare match
	Timer clock input pin 23	TMCI23	Input	Inputs external clock for counter
	Timer reset input pin 23	TMRI23	Input	Inputs external reset to counter
2	Timer output pin 2	TMO2	Output	Outputs at compare match
	Timer clock input pin 23	TMCI23	Input	Inputs external clock for counter
	Timer reset input pin 23	TMRI23	Input	Inputs external reset to counter
3	Timer output pin 3	TMO3	Output	Outputs at compare match
	Timer clock input pin 01	TMCI01	Input	Inputs external clock for counter
	Timer reset input pin 01	TMRI01	Input	Inputs external reset to counter

13.1.4 Register Configuration

Table 13-2 summarizes the registers of the 8-bit timer module.

Table 13-2 8-Bit Timer Registers

Channel	Name	Abbreviation	R/W	Initial value	Address*[1]
0	Timer control register 0	TCR0	R/W	H'00	H'FF68
	Timer control/status register 0	TCSR0	R/(W)*[2]	H'00	H'FF6A
	Time constant register A0	TCORA0	R/W	H'FF	H'FF6C
	Time constant register B0	TCORB0	R/W	H'FF	H'FF6E
	Timer counter 0	TCNT0	R/W	H'00	H'FF70
1	Timer control register 1	TCR1	R/W	H'00	H'FF69
	Timer control/status register 1	TCSR1	R/(W)*[2]	H'10	H'FF6B
	Time constant register A1	TCORA1	R/W	H'FF	H'FF6D
	Time constant register B1	TCORB1	R/W	H'FF	H'FF6F
	Timer counter 1	TCNT1	R/W	H'00	H'FF71
2	Timer control register 2	TCR2	R/W	H'00	H'FDC0
	Timer control/status register 2	TCSR2	R/(W)*[2]	H'00	H'FDC2
	Time constant register A2	TCORA2	R/W	H'FF	H'FDC4
	Time constant register B2	TCORB2	R/W	H'FF	H'FDC6
	Timer counter 2	TCNT2	R/W	H'00	H'FDC8
3	Timer control register 3	TCR3	R/W	H'00	H'FDC1
	Timer control/status register 3	TCSR3	R/(W)*[2]	H'10	H'FDC3
	Time constant register A3	TCORA3	R/W	H'FF	H'FDC5
	Time constant register B3	TCORB3	R/W	H'FF	H'FDC7
	Timer counter 3	TCNT3	R/W	H'00	H'FDC9
All	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address
*2 Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 (channel 2) and channel 1 (channel 3) is a 16-bit register with the upper 8 bits for channel 0 (channel 2) and the lower 8 bits for channel 1 (channel 3), so they can be accessed together by word transfer instruction.

13.2 Register Descriptions

13.2.1 Timer Counters 0 to 3 (TCNT0 to TCNT3)

	TCNT0 (TCNT2)								TCNT1 (TCNT3)							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	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

TCNT0 to TCNT3 are 8-bit readable/writable up-counters that increment on pulses generated from an internal or external clock source. This clock source is selected by clock select bits CKS2 to CKS0 of TCR. The CPU can read or write to TCNT0 to TCNT3 at all times.

TCNT0 and TCNT1 (TCNT2 and TCNT3) comprise a single 16-bit register, so they can be accessed together by word transfer instruction.

TCNT0 and TCNT1 (TCNT2 and TCNT3) can be cleared by an external reset input or by a compare match signal. Which signal is to be used for clearing is selected by clock clear bits CCLR1 and CCLR0 of TCR.

When a timer counter overflows from H'FF to H'00, OVF in TCSR is set to 1.

TCNT0 and TCNT1 are each initialized to H'00 by a reset and in hardware standby mode.

13.2.2 Time Constant Registers A0 to A3 (TCORA0 to TCORA3)

	TCORA0 (TCORA2)								TCORA1 (TCORA3)							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	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

TCORA0 to TCORA3 are 8-bit readable/writable registers. TCORA0 and TCORA1 (TCORA2 and TCORA3) comprise a single 16-bit register so they can be accessed together by word transfer instruction.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag of TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the settings of bits OS1 and OS0 of TCSR.

TCORA0 and TCORA1 are each initialized to H'FF by a reset and in hardware standby mode.

13.2.3 Time Constant Registers B0 to B3 (TCORB0 to TCORB3)

	TCORB0 (TCORB2)								TCORB1 (TCORB3)							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	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

TCORB0 to TCORB3 are 8-bit readable/writable registers. TCORB0 and TCORB1 (TCORB2 and TCORB3) comprise a single 16-bit register so they can be accessed together by word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag of TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the settings of output select bits OS3 and OS2 of TCSR.

TCORB0 and TCORB1 are each initialized to H'FF by a reset and in hardware standby mode.

13.2.4 Timer Control Registers 0 to 3 (TCR0 to TCR3)

Bit :	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR0 to TCR3 are 8-bit readable/writable registers that select the input clock source and the time at which TCNT is cleared, and enable interrupts.

TCR0 and TCR1 are each initialized to H'00 by a reset and in hardware standby mode.

For details of this timing, see section 13.3, Operation.

Bit 7—Compare Match Interrupt Enable B (CMIEB): Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag of TCSR is set to 1.

Bit 7		
CMIEB	**Description**	
0	CMFB interrupt requests (CMIB) are disabled	(Initial value)
1	CMFB interrupt requests (CMIB) are enabled	

Bit 6—Compare Match Interrupt Enable A (CMIEA): Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag of TCSR is set to 1.

Bit 6		
CMIEA	**Description**	
0	CMFA interrupt requests (CMIA) are disabled	(Initial value)
1	CMFA interrupt requests (CMIA) are enabled	

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag of TCSR is set to 1.

Bit 5		
OVIE	**Description**	
0	OVF interrupt requests (OVI) are disabled	(Initial value)
1	OVF interrupt requests (OVI) are enabled	

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select the method by which TCNT is cleared: by compare match A or B, or by an external reset input.

Bit 4	Bit 3		
CCLR1	**CCLR0**	**Description**	
0	0	Clear is disabled	(Initial value)
	1	Clear by compare match A	
1	0	Clear by compare match B	
	1	Clear by rising edge of external reset input	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select whether the clock input to TCNT is an internal or external clock.

Three internal clocks can be selected, all divided from the system clock (ø): ø/8, ø/64, and ø/8192. The falling edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the rising edge, the falling edge, and both rising and falling edges.

Some functions differ between channel 0 and channel 1.

Bit 2	Bit 1	Bit 0		
CKS2	**CKS1**	**CKS0**	**Description**	
0	0	0	Clock input disabled	(Initial value)
		1	Internal clock, counted at falling edge of ø/8	
	1	0	Internal clock, counted at falling edge of ø/64	
		1	Internal clock, counted at falling edge of ø/8192	
1	0	0	For channel 0: count at TCNT1 overflow signal*	
			For channel 1: count at TCNT0 compare match A*	
			For channel 2: count at TCNT3 overflow signal*	
			For channel 3: count at TCNT2 compare match A*	
		1	External clock, counted at rising edge	
	1	0	External clock, counted at falling edge	
		1	External clock, counted at both rising and falling edges	

Note: * If the count input of channel 0 (channel 2) is the TCNT1 (TCNT3) overflow signal and that of channel 1 (channel 3) is the TCNT0 (TCNT2) compare match signal, no incrementing clock is generated. Do not use this setting.

13.2.5 Timer Control/Status Registers 0 to 3 (TCSR0 to TCSR3)

TCSR0

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

TCSR1, TCSR3

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	1	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

TCSR2

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

TCSR0 to TCSR3 are 8-bit registers that display compare match and overflow statuses, and control compare match output.

TCSR0 and TCSR2 are initialized to H'00, and TCSR1 and TCSR3 to H'10, by a reset and in hardware standby mode.

Bit 7—Compare Match Flag B (CMFB): Status flag indicating whether the values of TCNT and TCORB match.

Bit 7

CMFB	Description
0	[Clearing conditions] (Initial value) • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
1	[Setting condition] Set when TCNT matches TCORB

Bit 6—Compare Match Flag A (CMFA): Status flag indicating whether the values of TCNT and TCORA match.

Bit 6

CMFA	Description
0	[Clearing conditions] (Initial value) • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
1	[Setting condition] Set when TCNT matches TCORA

Bit 5—Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflowed (changed from H'FF to H'00).

Bit 5

OVF	Description
0	[Clearing condition] (Initial value) Cleared by reading OVF when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows from H'FF to H'00

Bit 4—A/D Trigger Enable (ADTE) (TCSR0 Only): Selects enabling or disabling of A/D converter start requests by compare-match A.

TCSR1 to TCSR3 are reserved bits. When TCSR1 and TCSR3 are read, always 1 is read off. Write is disenabled. TCSR2 is readable/writable.

Bit 4		
ADTE	**Description**	
0	A/D converter start requests by compare match A are disabled	(Initial value)
1	A/D converter start requests by compare match A are enabled	

Bits 3 to 0—Output Select 3 to 0 (OS3 to OS0): These bits specify how the timer output level is to be changed by a compare match of TCOR and TCNT.

Bits OS3 and OS2 select the effect of compare match B on the output level, bits OS1 and OS0 select the effect of compare match A on the output level, and both of them can be controlled independently.

Note, however, that priorities are set such that: toggle output > 1 output > 0 output. If compare matches occur simultaneously, the output changes according to the compare match with the higher priority.

Timer output is disabled when bits OS3 to OS0 are all 0.

After a reset, the timer output is 0 until the first compare match event occurs.

Bit 3	Bit 2		
OS3	**OS2**	**Description**	
0	0	No change when compare match B occurs	(Initial value)
	1	0 is output when compare match B occurs	
1	0	1 is output when compare match B occurs	
	1	Output is inverted when compare match B occurs (toggle output)	

Bit 1	Bit 0		
OS1	**OS0**	**Description**	
0	0	No change when compare match A occurs	(Initial value)
	1	0 is output when compare match A occurs	
1	0	1 is output when compare match A occurs	
	1	Output is inverted when compare match A occurs (toggle output)	

13.2.6 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA4 and MSTPA0 bits in MSTPCR is set to 1, the 8-bit timer operation stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 4—Module Stop (MSTPA4): Specifies the TMR0 and TMR1 module stop mode.

Bit 4

MSTPA4	Description	
0	TMR0, TMR1 module stop mode cleared	
1	TMR0, TMR1 module stop mode set	(Initial value)

Bit 0—Module Stop (MSTPA0): Specifies the TMR2 and TMR3 module stop mode.

Bit 0

MSTPA0	Description	
0	TMR2, TMR3 module stop mode cleared	
1	TMR2, TMR3 module stop mode set	(Initial value)

13.3 Operation

13.3.1 TCNT Incrementation Timing

TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: Three different internal clock signals (ø/8, ø/64, or ø/8192) divided from the system clock (ø) can be selected, by setting bits CKS2 to CKS0 in TCR. Figure 13-2 shows the count timing.

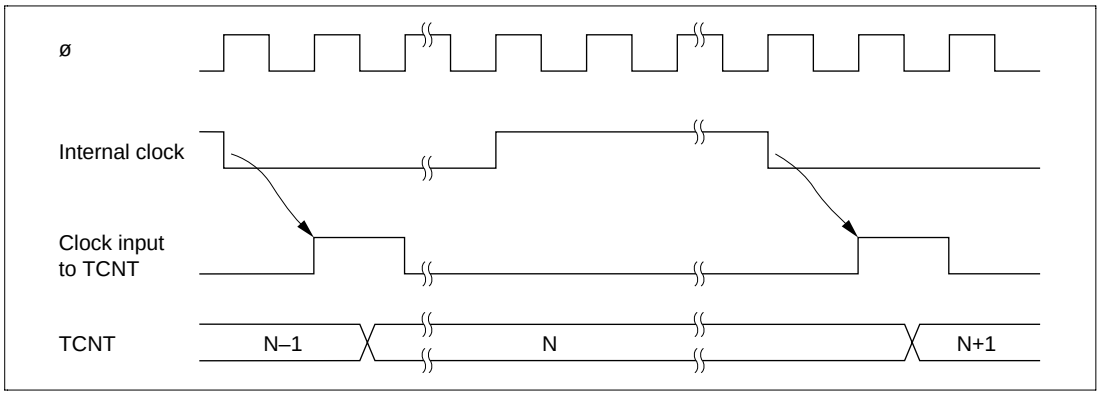


Figure 13-2 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in TCR: at the rising edge, the falling edge, and both rising and falling edges.

Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

Figure 13-3 shows the timing of incrementation at both edges of an external clock signal.

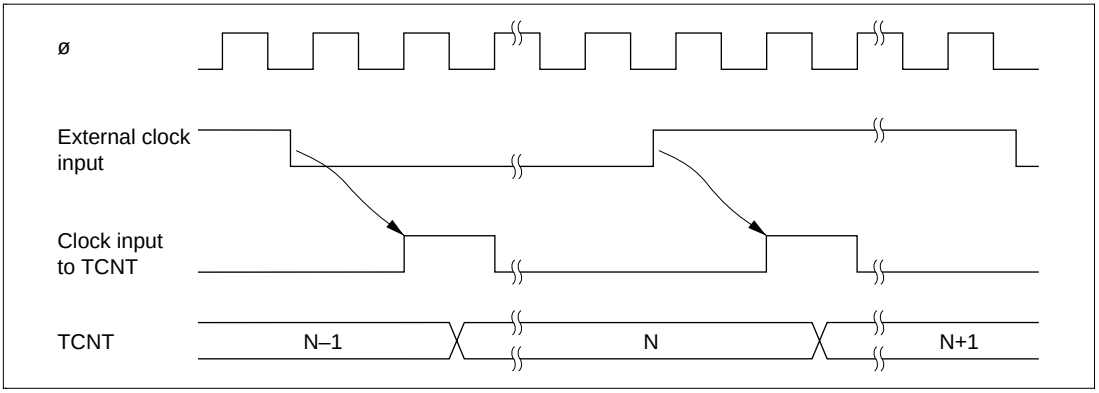


Figure 13-3 Count Timing for External Clock Input

13.3.2 Compare Match Timing

Setting of Compare Match Flags A and B (CMFA, CMFB): The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated.

Therefore, when TCOR and TCNT match, the compare match signal is not generated until the next incrementation clock input. Figure 13-4 shows this timing.

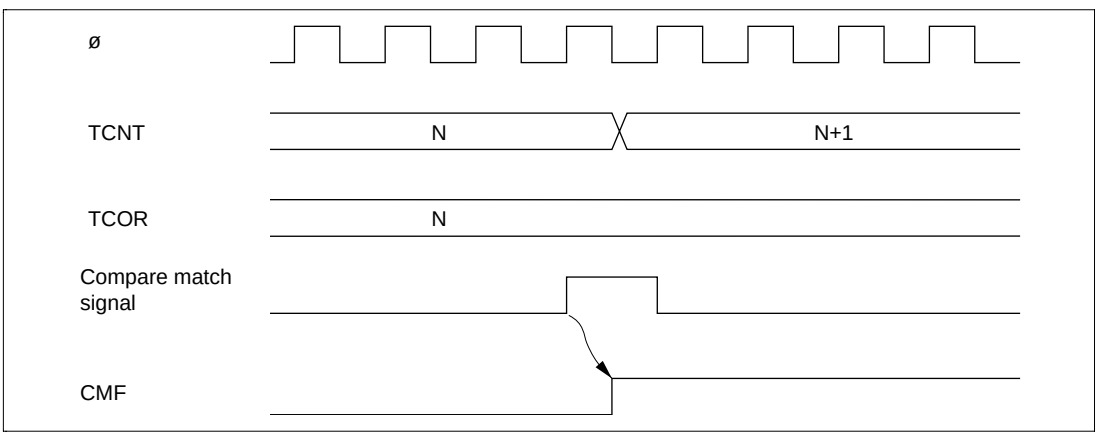


Figure 13-4 Timing of CMF Setting

Timer Output Timing: When compare match A or B occurs, the timer output changes a specified by bits OS3 to OS0 in TCSR. Depending on these bits, the output can remain the same, change to 0, change to 1, or toggle.

Figure 13-5 shows the timing when the output is set to toggle at compare match A.

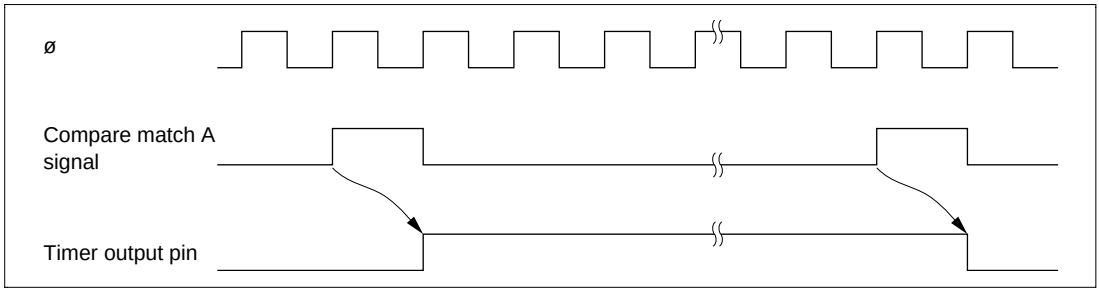


Figure 13-5 Timing of Timer Output

Timing of Compare Match Clear: The timer counter is cleared when compare match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 13-6 shows the timing of this operation.

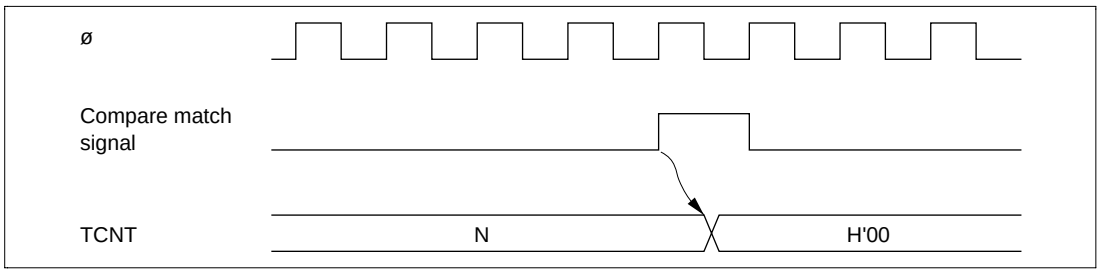


Figure 13-6 Timing of Compare Match Clear

13.3.3 Timing of External RESET on TCNT

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The clear pulse width must be at least 1.5 states. Figure 13-7 shows the timing of this operation.

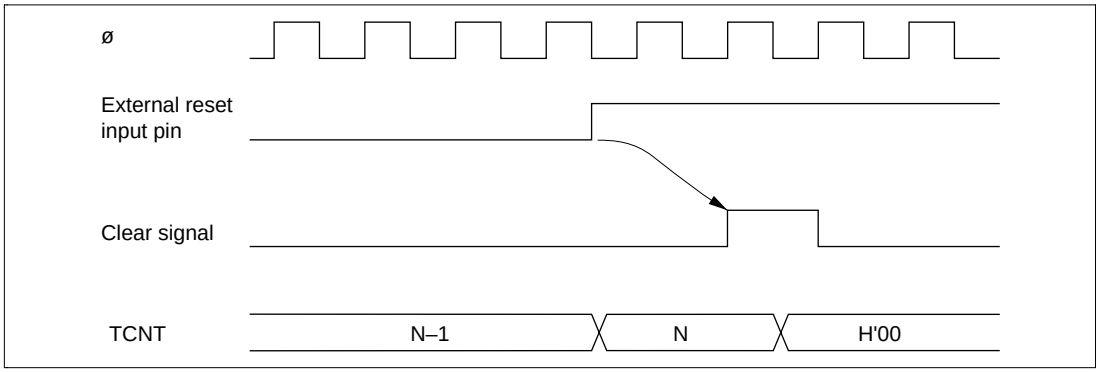


Figure 13-7 Timing of External Reset

13.3.4 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 13-8 shows the timing of this operation.

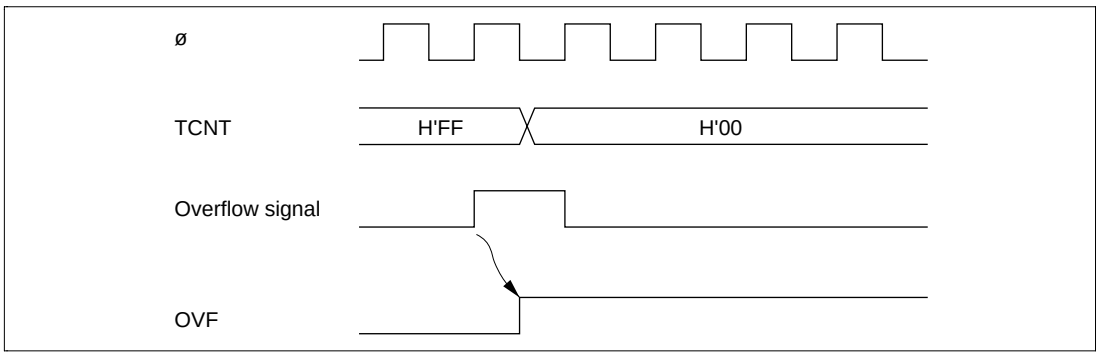


Figure 13-8 Timing of OVF Setting

13.3.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR0 or TCR1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit timer mode) or compare matches of the 8-bit timer channel 0 could be counted by the timer of channel 1 (compare match counter mode). In this case, the timer operates as below.

16-Bit Counter Mode: When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

- Setting of compare match flags
  - — The CMF flag in TCSR0 and TCSR2 is set to 1 when a 16-bit compare match event occurs.
  - — The CMF flag in TCSR1 and TCSR3 is set to 1 when a lower 8-bit compare match event occurs.

- Counter clear specification
  - — If the CCLR1 and CCLR0 bits in TCR0 (TCR2) have been set for counter clear at compare match, the 16-bit counter (TCNT0 and TCNT1 (TCNT2 and TCNT3) together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TCNT0 and TCNT1 (TCNT2 and TCNT3) together) is cleared even if counter clear by the TMRI01 (TMRI23) pin has also been set.
  - — The settings of the CCLR1 and CCLR0 bits in TCR1 and TCR3 are ignored. The lower 8 bits cannot be cleared independently.

- Pin output
  - — Control of output from the TMO0 (TMO2) pin by bits OS3 to OS0 in TCSR0 (TCSR2) is in accordance with the 16-bit compare match conditions.
  - — Control of output from the TMO1 (TMO3) pin by bits OS3 to OS0 in TCSR1 (TCSR3) is in accordance with the lower 8-bit compare match conditions.

Compare Match Counter Mode: When bits CKS2 to CKS0 in TCR1 (TCR3) are B'100, TCNT1 (TCNT3) counts compare match A's for channel 0 (channel 2).

Channels 0 to 3 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

Note on Usage: If the 16-bit counter mode and compare match counter mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 (TCNT2 and TCNT3) are not generated and thus the counters will stop operating. Software should therefore avoid using both these modes.

13.4 Interrupts

13.4.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in Table 13-3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

Table 13-3 8-Bit Timer Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation	Priority
0	CMIA0	Interrupt by CMFA	Possible	High
	CMIB0	Interrupt by CMFB	Possible	
	OVI0	Interrupt by OVF	Not possible	
1	CMIA1	Interrupt by CMFA	Possible	
	CMIB1	Interrupt by CMFB	Possible	
	OVI1	Interrupt by OVF	Not possible	
2	CMIA2	Interrupt by CMFA	Possible	
	CMIB2	Interrupt by CMFB	Possible	
	OVI2	Interrupt by OVF	Not possible	
3	CMIA3	Interrupt by CMFA	Possible	
	CMIB3	Interrupt by CMFB	Possible	
	OVI3	Interrupt by OVF	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

13.4.2 A/D Converter Activation

The A/D converter can be activated only by channel 0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of channel 0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

13.5 Sample Application

In the example below, the 8-bit timer is used to generate a pulse output with a selected duty cycle, as shown in figure 13-9. The control bits are set as follows:

[1] In TCR, bit CCLR1 is cleared to 0 and bit CCLR0 is set to 1 so that the timer counter is cleared when its value matches the constant in TCORA.

[2] In TCSR, bits OS3 to OS0 are set to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

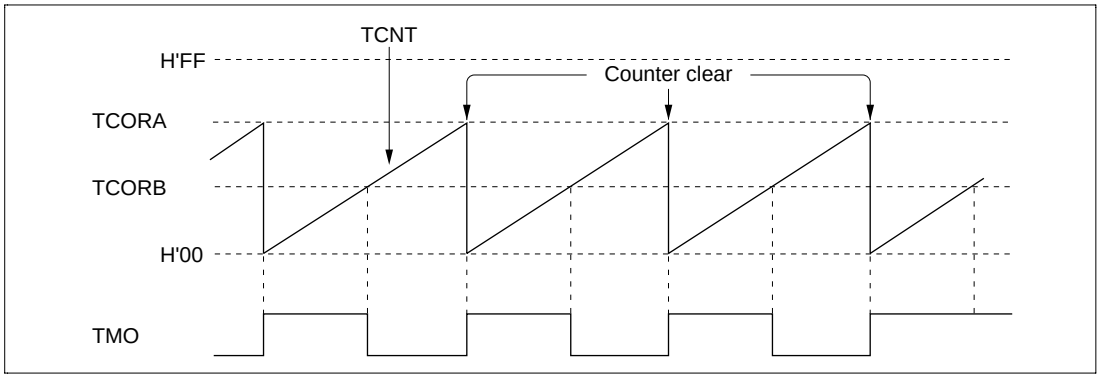


Figure 13-9 Example of Pulse Output

13.6 Usage Notes

Application programmers should note that the following kinds of contention can occur in the 8-bit timer.

13.6.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed.

Figure 13-10 shows this operation.

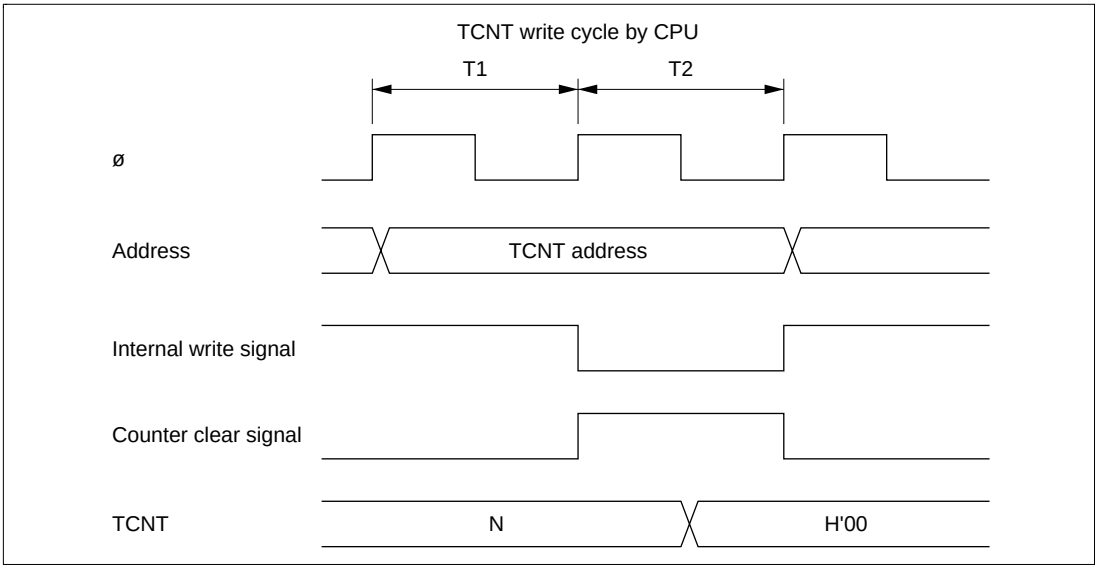


Figure 13-10 Contention between TCNT Write and Clear

13.6.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the write takes priority and the counter is not incremented.

Figure 13-11 shows this operation.

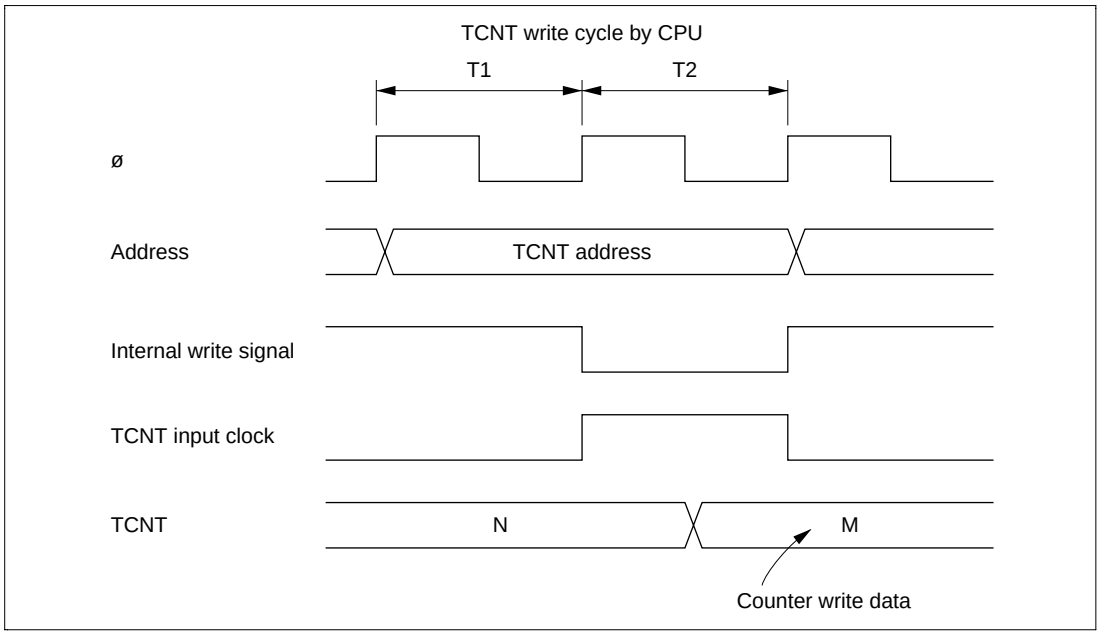


Figure 13-11 Contention between TCNT Write and Increment

13.6.3 Contention between TCOR Write and Compare Match

During the T2 state of a TCOR write cycle, the TCOR write has priority and the compare match signal is disabled even if a compare match event occurs.

Figure 13-12 shows this operation.

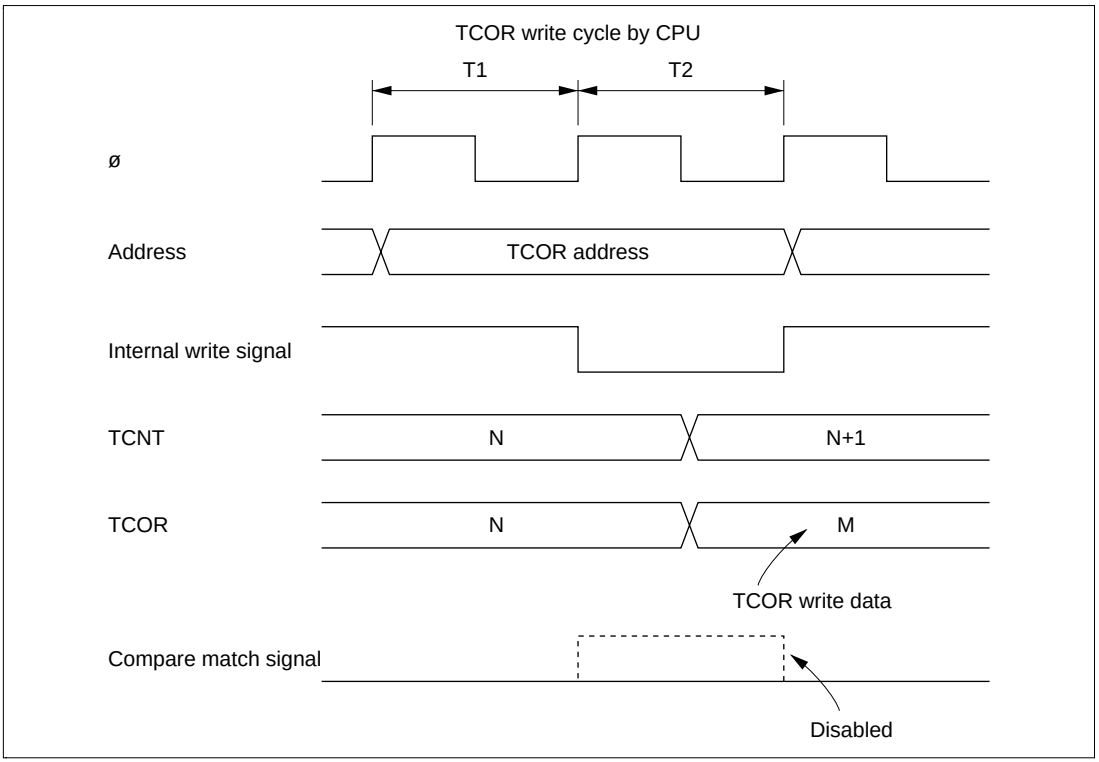


Figure 13-12 Contention between TCOR Write and Compare Match

13.6.4 Contention between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 13-4.

Table 13-4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

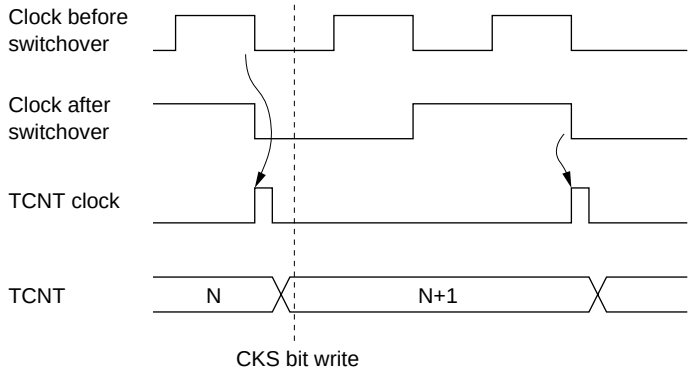
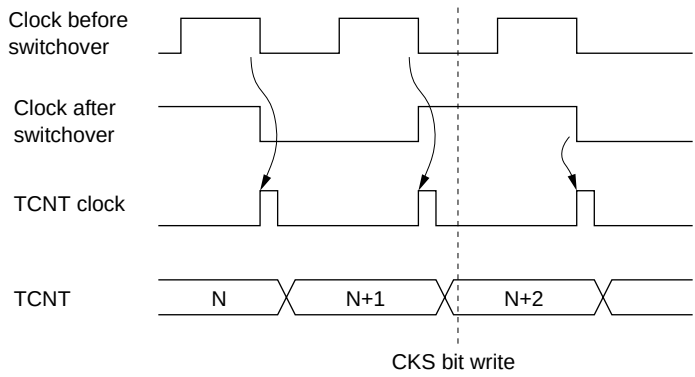
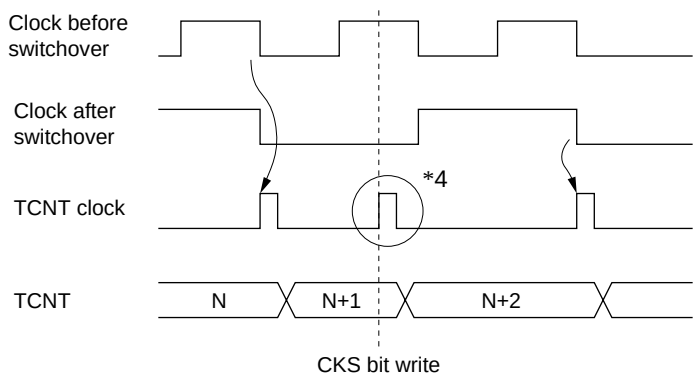
13.6.5 Switching of Internal Clocks and TCNT Operation

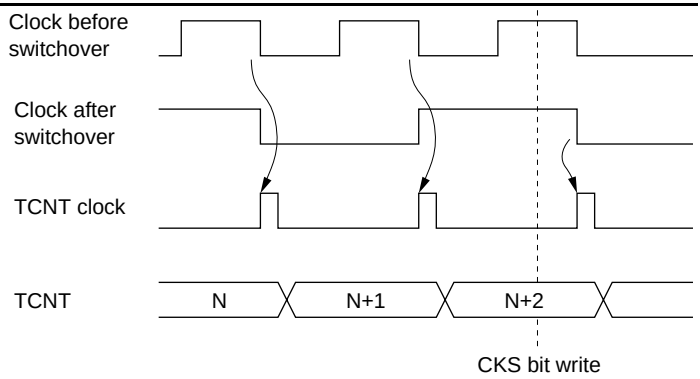
TCNT may increment erroneously when the internal clock is switched over. Table 13-5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in case 3 in table 13-5, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and external clocks.

Table 13-5 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*[1]	
2	Switching from low to high*[2]	
3	Switching from high to low*[3]	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
4	Switching from high to high	

Notes: *1 Includes switching from low to stop, and from stop to low.
*2 Includes switching from stop to high.
*3 Includes switching from high to stop.
*4 Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

13.6.6 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or DMAC and DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 14 14-Bit PWM D/A

14.1 Overview

The H8S/2633 Series has an on-chip 14-bit pulse-width modulator (PWM) with four output channels.

Each channel can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

Both channels share the same counter (DACNT) and control register (DACR).

14.1.1 Features

The features of the 14-bit PWM D/A are listed below.

- The pulse is subdivided into multiple base cycles to reduce ripple.
- Two resolution settings and two base cycle settings are available

  The resolution can be set equal to one or two system clock cycles. The base cycle can be set equal to T × 64 or T × 256, where T is the resolution.
- Four operating rates

  The two resolution settings and two base cycle settings combine to give a selection of four operating rates.

14.1.2 Block Diagram

Figure 14-1 shows a block diagram of the PWM D/A module.

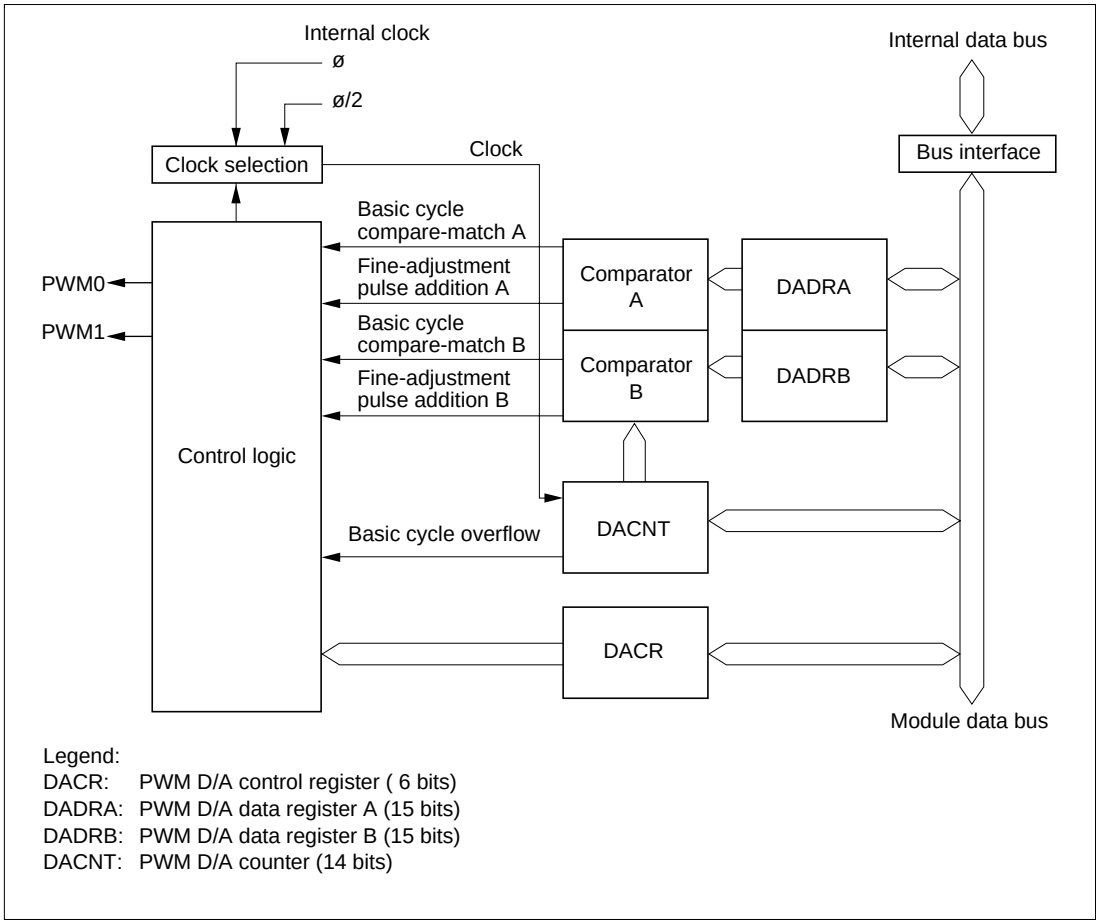


Figure 14-1 PWM D/A Block Diagram

14.1.3 Pin Configuration

Table 14-1 lists the pins used by the PWM D/A module.

Table 14-1 Input and Output Pins

Name	Abbr.	I/O	Function
PWM output pin 0	PWM0	Output	PWM output, channel 0A
PWM output pin 1	PWM1	Output	PWM output, channel 0B
PWM output pin 2	PWM2	Output	PWM output, channel 1A
PWM output pin 3	PWM3	Output	PWM output, channel 1B

14.1.4 Register Configuration

Table 14-2 lists the registers of the PWM D/A module.

Table 14-2 Register Configuration

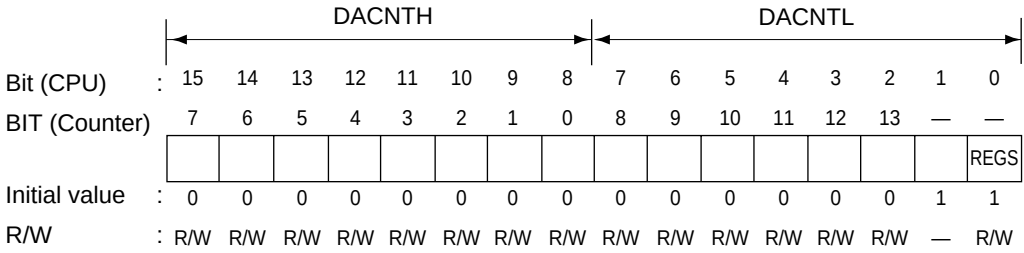
Channel	Name	Abbreviation	R/W	Initial value	Address*[1]
0	PWM D/A control register 0	DACR0	R/W	H'30	H'FDB8*[2]
	PWM D/A data register AH0	DADRAH0	R/W	H'FF	H'FDB8*[2]
	PWM D/A data register AL0	DADRAL0	R/W	H'FF	H'FDB9*[2]
	PWM D/A data register BH0	DADRBH0	R/W	H'FF	H'FDBA*[2]
	PWM D/A data register BL0	DADRBL0	R/W	H'FF	H'FDBB*[2]
	PWM D/A counter H0	DACNTH0	R/W	H'00	H'FDBA*[2]
	PWM D/A counter L0	DACNTL0	R/W	H'03	H'FDBB*[2]
1	PWM D/A control register 1	DACR1	R/W	H'30	H'FDBC*[2]
	PWM D/A data register AH1	DADRAH1	R/W	H'FF	H'FDBC*[2]
	PWM D/A data register AL1	DADRAL1	R/W	H'FF	H'FDBD*[2]
	PWM D/A data register BH1	DADRBH1	R/W	H'FF	H'FDBE*[2]
	PWM D/A data register BL1	DADRBL1	R/W	H'FF	H'FDBF*[2]
	PWM D/A counter H1	DACNTH1	R/W	H'00	H'FDBE*[2]
	PWM D/A counter L1	DACNTL1	R/W	H'03	H'FDBF*[2]
All	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.

*2 The same addresses are shared by DADRA and DACR, and by DADRB and DACNT. Switching is performed by the REGS bit in DACNT or DADRB.

14.2 Register Descriptions

14.2.1 PWM D/A Counter (DACNT)



DACNT is a 14-bit readable/writable up-counter that increments on an input clock pulse. The input clock is selected by the clock select bit (CKS) in DACR. The CPU can read and write the DACNT value, but since DACNT is a 16-bit register, data transfers between it and the CPU are performed using a temporary register (TEMP). See section 14.3, Bus Master Interface, for details.

DACNT functions as the time base for both PWM D/A channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 (counter) bits and ignores the upper two (counter) bits.

DACNT is initialized to H'0003 by a reset, in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode, and by the PWME bit.

Bit 1 of DACNTL (CPU) is not used, and is always read as 1.

DACNTL Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0

REGS	Description	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

14.2.2 PWM D/A Data Registers A and B (DADRA and DADRB)

	DADRH								DADRL							
Bit (CPU)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit (Data)	13	12	11	10	9	8	7	6	5	4	3	2	1	0	—	—
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
Initial value :	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	—
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
Initial value :	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

There are two 16-bit readable/writable PWM D/A data registers: DADRA and DADRB. DADRA corresponds to PWM D/A channel A, and DADRB to PWM D/A channel B. The CPU can read and write the PWM D/A data register values, but since DADRA and DADRB are 16-bit registers, data transfers between them and the CPU are performed using a temporary register (TEMP). See section 14.3, Bus Master Interface, for details.

The least significant (CPU) bit of DADRA is not used and is always read as 1.

DADR is initialized to H'FFFF by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

Bits 15 to 3—PWM D/A Data 13 to 0 (DA13 to DA0): The digital value to be converted to an analog value is set in the upper 14 bits of the PWM D/A data register.

In each base cycle, the DACNT value is continually compared with these upper 14 bits to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, the data register must be set within a range that depends on the carrier frequency select bit (CFS). If the DADR value is outside this range, the PWM output is held constant.

A channel can be operated with 12-bit precision by keeping the two lowest data bits (DA0 and DA1) cleared to 0 and writing the data to be converted in the upper 12 bits. The two lowest data bits correspond to the two highest counter (DACNT) bits.

Bit 1—Carrier Frequency Select (CFS)

Bit 1		
CFS	**Description**	
0	Base cycle = resolution (T) × 64 DADR range = H'0401 to H'FFFD	
1	Base cycle = resolution (T) × 256 DADR range = H'0103 to H'FFFF	(Initial value)

Bit 0—Reserved: This bit cannot be modified and is always read as 1.

DADRB Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0		
REGS	**Description**	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

14.2.3 PWM D/A Control Register (DACR)

Bit	:	7	6	5	4	3	2	1	0
		TEST	PWME	—	—	OEB	OEA	OS	CKS
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	—	—	R/W	R/W	R/W	R/W

DACR is an 8-bit readable/writable register that selects test mode, enables the PWM outputs, and selects the output phase and operating speed.

DACR is initialized to H'30 by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—Test Mode (TEST): Selects test mode, which is used in testing the chip. Normally this bit should be cleared to 0.

Bit 7		
TEST	**Description**	
0	PWM (D/A) in user state: normal operation	(Initial value)
1	PWM (D/A) in test state: correct conversion results unobtainable	

Bit 6—PWM Enable (PWME): Starts or stops the PWM D/A counter (DACNT).

Bit 6		
PWME	**Description**	
0	DACNT operates as a 14-bit up-counter	(Initial value)
1	DACNT halts at H'0003	

Bits 5 and 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Output Enable B (OEB): Enables or disables output on PWM D/A channel B.

Bit 3		
OEB	**Description**	
0	PWM (D/A) channel B output (at the PWM1/PWM3 pin) is disabled	(Initial value)
1	PWM (D/A) channel B output (at the PWM1/PWM3 pin) is enabled	

Bit 2—Output Enable A (OEA): Enables or disables output on PWM D/A channel A.

Bit 2		
OEA	**Description**	
0	PWM (D/A) channel A output (at the PWM0/PWM2 pin) is disabled	(Initial value)
1	PWM (D/A) channel A output (at the PWM0/PWM2 pin) is enabled	

Bit 1—Output Select (OS): Selects the phase of the PWM D/A output.

Bit 1		
OS	**Description**	
0	Direct PWM output	(Initial value)
1	Inverted PWM output	

Bit 0—Clock Select (CKS): Selects the PWM D/A resolution. If the system clock (ø) frequency is 10 MHz, resolutions of 100 ns and 200 ns can be selected.

Bit 0

CKS	Description	
0	Operates at resolution (T) = system clock cycle time (t_{cyc})	(Initial value)
1	Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2	

14.2.4 Module Stop Control Register B (MSTPCRB)

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	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

MSTPCRB is an 8-bit readable/writable register, and is used to perform module stop mode control.

When the MSTPB2 is set to 1, at the end of the bus cycle 14-bit PWM timer 0 operation is halted and a transition made to module stop mode. When the MSTPB1 is set to 1, at the end of the bus cycle PWM timer 1 operation is halted and a transition made to module stop mode. See 24.5 Module Stop Mode for details.

MSTPCRB is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized in manual reset or software standby mode.

Bit 2—Module Stop (MSTPB2): Specifies PWM0 module stop mode.

Bit 2

MSTPB2	Description	
0	PWM0 module stop mode is cleared	
1	PWM0 module stop mode is set	(Initial value)

Bit 1—Module Stop (MSTPB1): Specifies PWM1 module stop mode.

Bit 1

MSTPB1	Description	
0	PWM1 module stop mode is cleared	
1	PWM1 module stop mode is set	(Initial value)

14.3 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip supporting modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written and read as follows (taking the example of the CPU interface).

- Write

  When the upper byte is written, the upper-byte write data is stored in TEMP. Next, when the lower byte is written, the lower-byte write data and TEMP value are combined, and the combined 16-bit value is written in the register.

- Read

  When the upper byte is read, the upper-byte value is transferred to the CPU and the lower-byte value is transferred to TEMP. Next, when the lower byte is read, the lower-byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time (by word access or two consecutive byte accesses), and the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed.

Figure 14-2 shows the data flow for access to DACNT. The other registers are accessed similarly.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT        ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0        ; Copy contents of DADRA to R0
```

Table 14-3 Read and Write Access Methods for 16-Bit Registers

	Read		Write	
Register Name	**Word**	**Byte**	**Word**	**Byte**
DADRA and DADRB	Yes	Yes	Yes	×
DACNT	Yes	×	Yes	×

Notes: Yes: Permitted type of access. Word access includes successive byte accesses to the upper byte (first) and lower byte (second).
×: This type of access may give incorrect results.

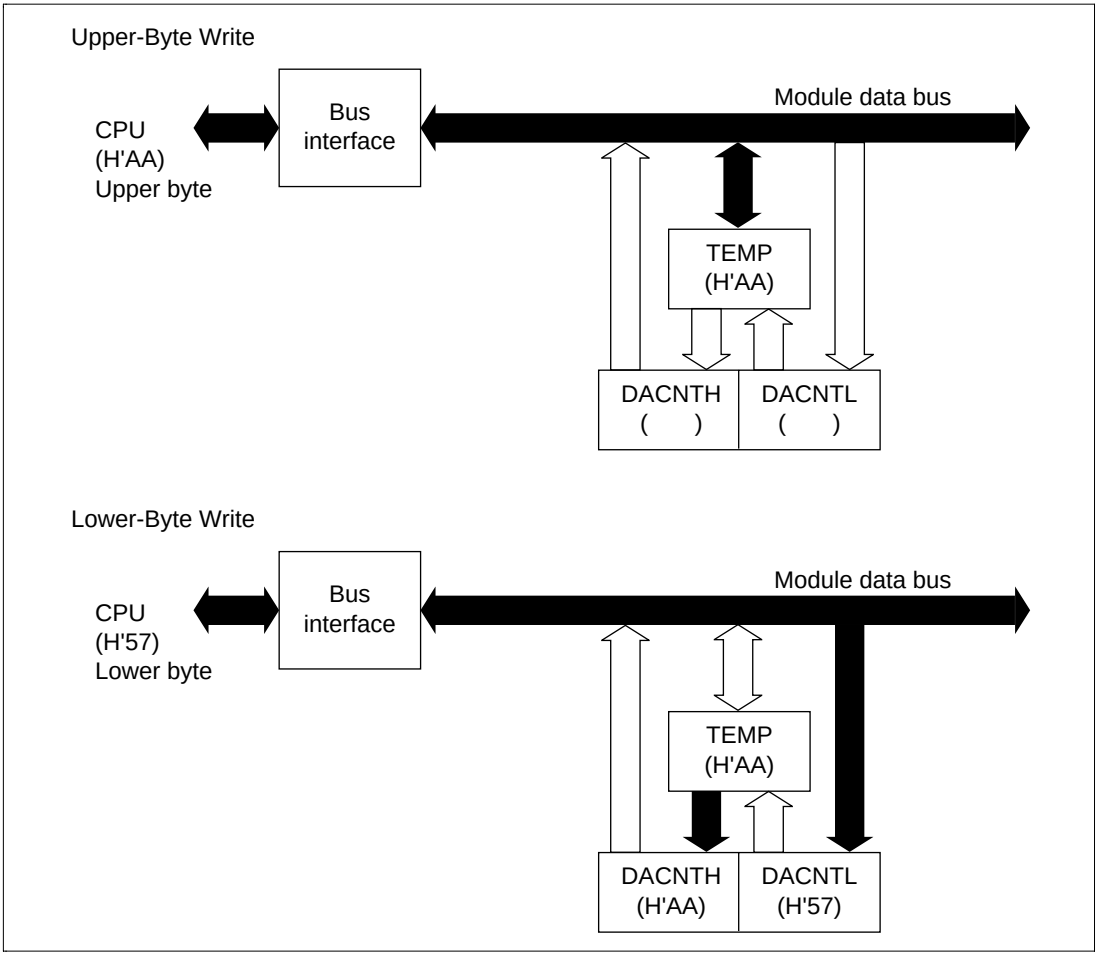


Figure 14-2 (a) Access to DACNT (CPU Writes H'AA57 to DACNT)

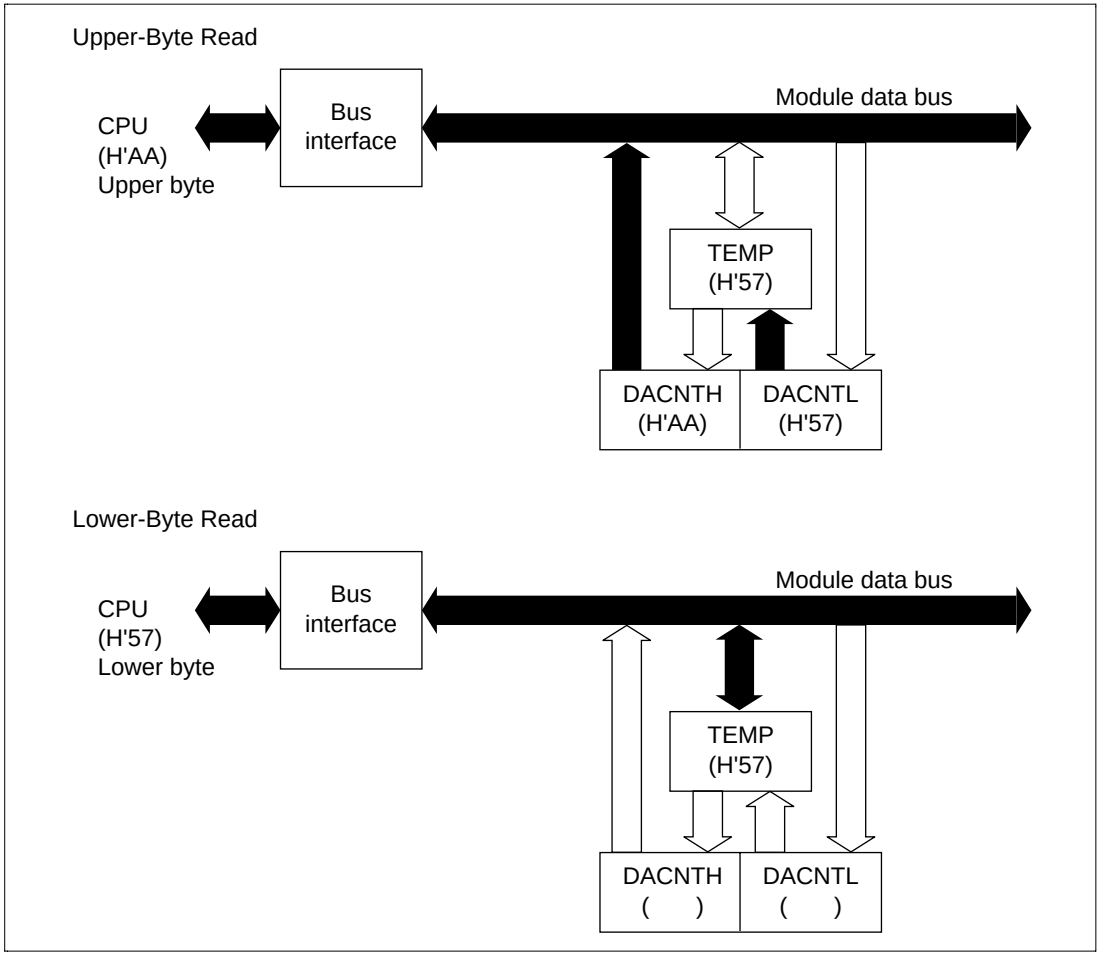


Figure 14-2 (b) Access to DACNT (CPU Reads H'AA57 from DACNT)

14.4 Operation

A PWM waveform like the one shown in figure 14-3 is output from the PWMX pin. When OS = 0, the value in DADR corresponds to the total width (T_L) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 1, the output waveform is inverted and the DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figure 14-4 shows the types of waveform output available.

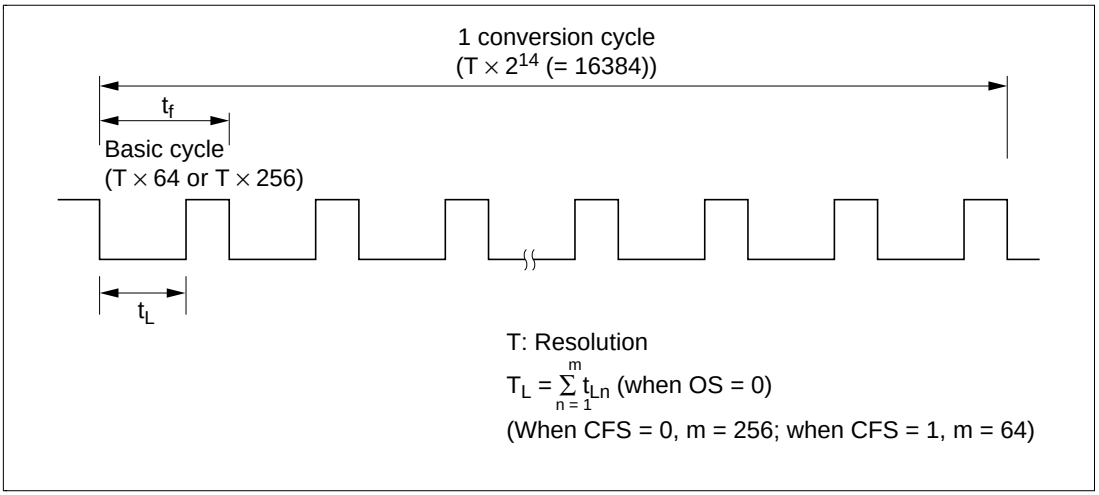


Figure 14-3 PWM D/A Operation

Table 14-4 summarizes the relationships of the CKS, CFS, and OS bit settings to the resolution, base cycle, and conversion cycle. The PWM output remains flat unless DADR contains at least a certain minimum value. Table 14-4 indicates the range of DADR settings that give an output waveform like the one in figure 14-3, and lists the conversion cycle length when low-order DADR bits are kept cleared to 0, reducing the conversion precision to 12 bits or 10 bits.

Table 14-4 Settings and Operation (Examples when ø = 10 MHz)

CKS	Resolution T (µs)	CFS	Base Cycle (µs)	Conversion Cycle (µs)	T_L (if OS = 0) T_H (if OS = 1)	Fixed DADR Bits: Precision (Bits)	Bit Data 3	2	1	0	Conversion Cycle* (µs)
0	0.1	0	6.4	1638.4	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					1638.4
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	409.6
						10	0	0	0	0	102.4
		1	25.6	1638.4	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					1638.4
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	409.6
						10	0	0	0	0	102.4
1	0.2	0	12.8	3276.8	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					3276.8
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	819.2
						10	0	0	0	0	204.8
		1	51.2	3276.8	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					3276.8
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	819.2
						10	0	0	0	0	204.8

Note: * This column indicates the conversion cycle when specific DADR bits are fixed.

1. OS = 0 (DADR corresponds to T_L)

   a. CFS = 0 [base cycle = resolution (T) × 64]

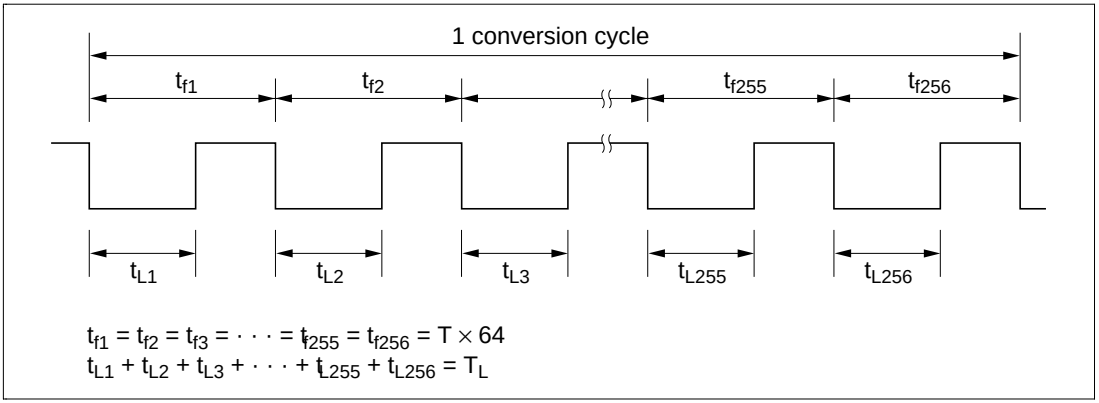


Figure 14-4 (1) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]

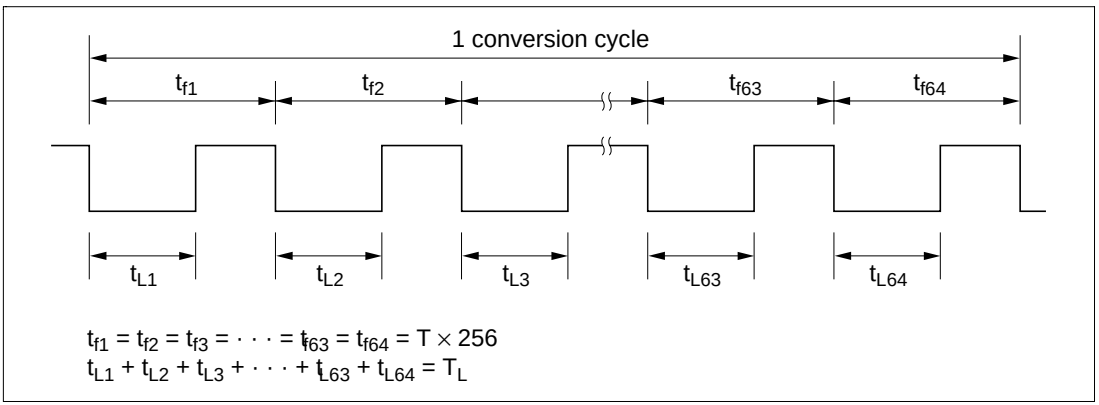
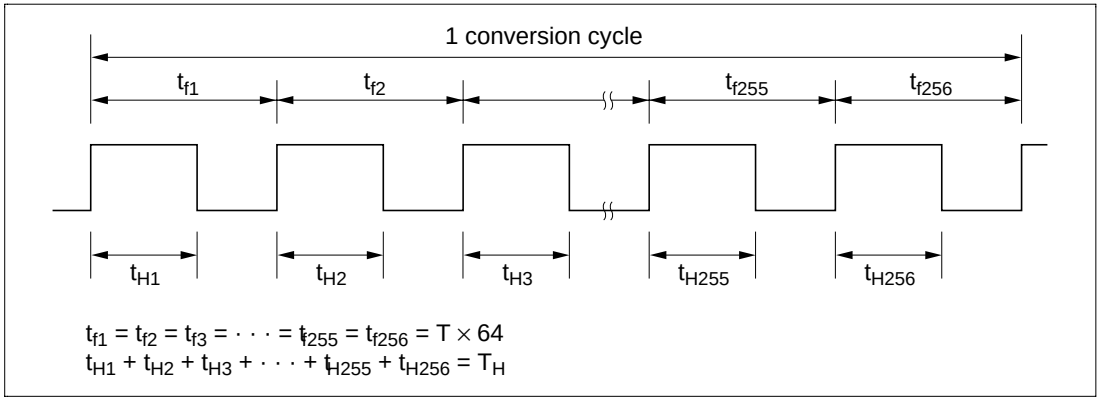


Figure 14-4 (2) Output Waveform

2. OS = 1 (DADR corresponds to T_H)

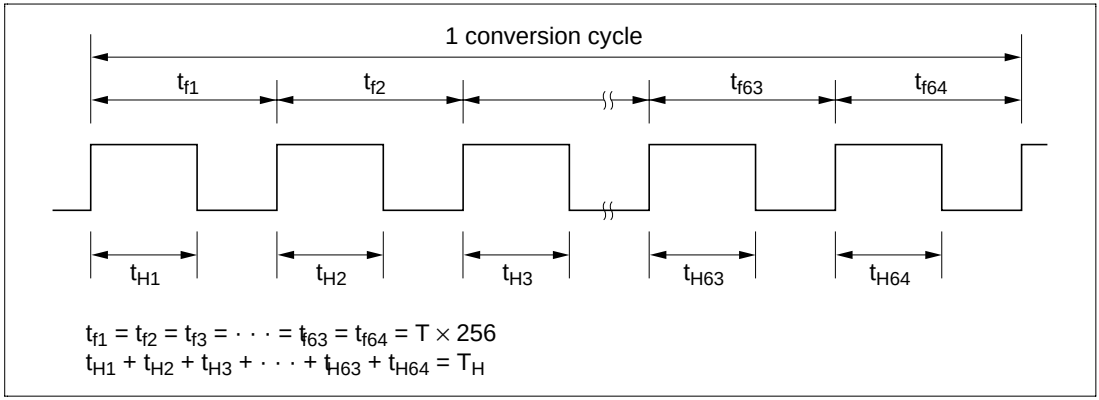
   a. CFS = 0 [base cycle = resolution (T) × 64]



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f255} = t_{f256} = T \times 64$
$t_{H1} + t_{H2} + t_{H3} + \cdots + t_{H255} + t_{H256} = T_H$

Figure 14-4 (3) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{H1} + t_{H2} + t_{H3} + \cdots + t_{H63} + t_{H64} = T_H$

Figure 14-4 (4) Output Waveform

Section 15 Watchdog Timer

15.1 Overview

The H8S/2633 Series has a two channel inbuilt watchdog timer, (WDT0/WDT1). The WDT outputs an overflow signal ($\overline{\text{WDTOVF}}$) if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow. At the same time, the WDT can also generate an internal reset signal for the H8S/2633 Series.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

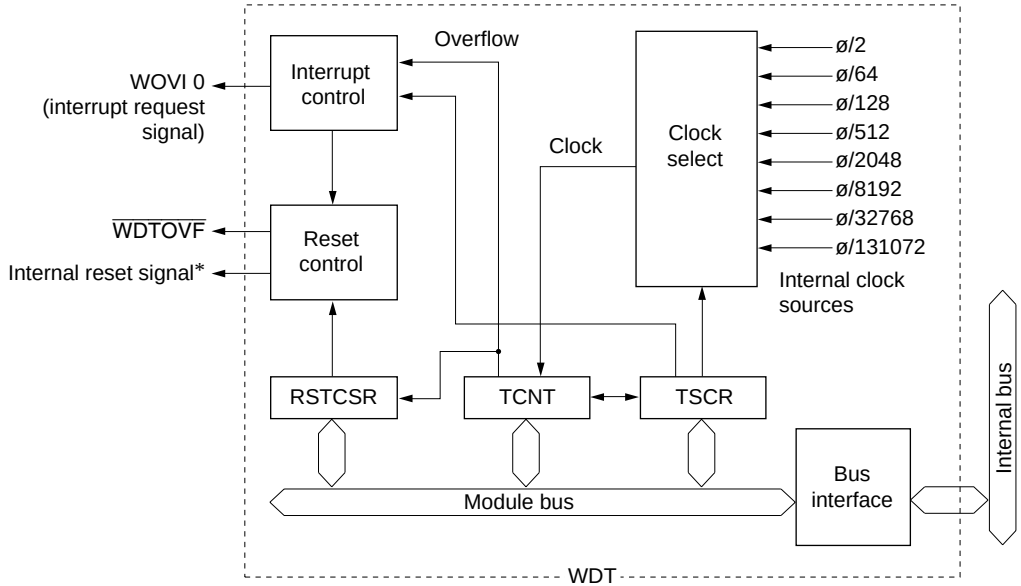
15.1.1 Features

WDT features are listed below.

- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output when in watchdog timer mode
  If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whether the LSI is internally reset or an NMI interrupt is generated at the same time.
  This internal reset is effected by either a power-on reset or a manual reset.
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt.
- WDT0 and WDT1 respectively allow eight and sixteen types of counter input clock to be selected
  The maximum interval of the WDT is given as a system clock cycle × 131072 × 256.
  A subclock may be selected for the input counter of WDT1.
  Where a subclock is selected, the maximum interval is given as a subclock cycle × 256 × 256.
- Selected clock can be output from the BUZZ output pin (WDT1)

15.1.2 Block Diagram

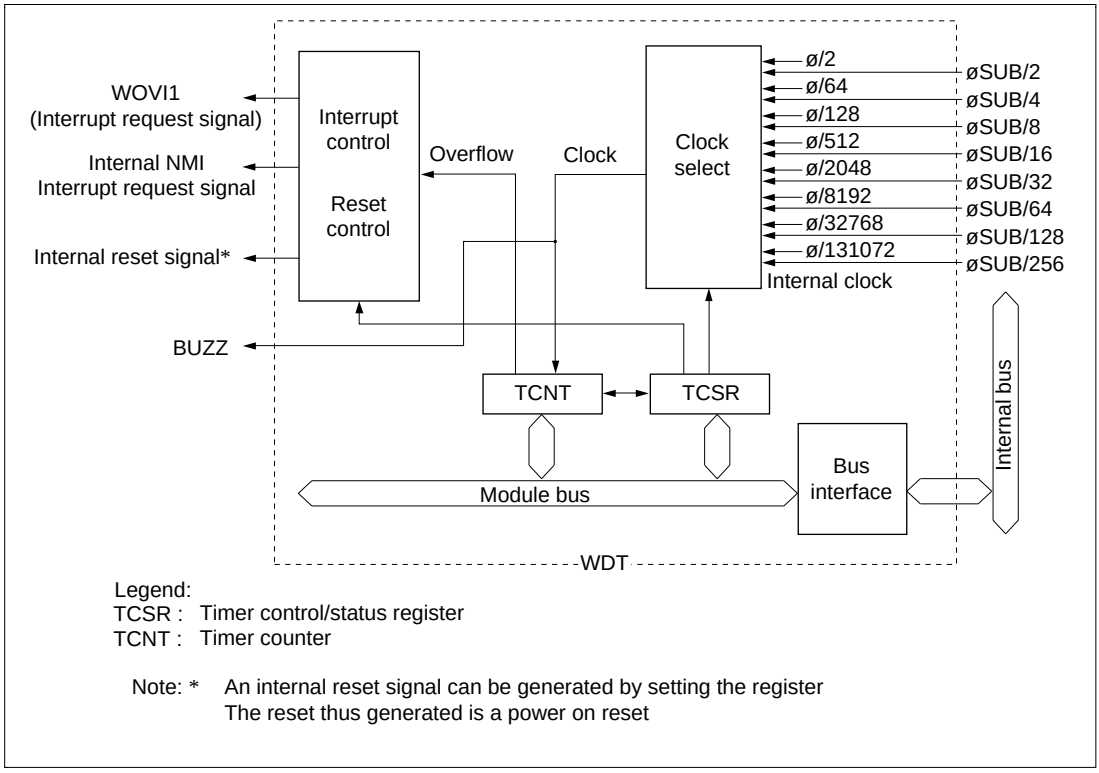
Figure 15-1 (a) and 15-1 (b) show a block diagram of the WDT.



Legend
TCSR : Timer control/status register
TCNT : Timer counter
RSTCSR : Reset control/status register

Note: * The type of internal reset signal depends on a register setting.
There are two alternative types of reset, namely power-on reset and manual reset.

Figure 15-1 (a) Block Diagram of WDT0



Legend:
TCSR : Timer control/status register
TCNT : Timer counter

Note: * An internal reset signal can be generated by setting the register
The reset thus generated is a power on reset

Figure 15-1 (b) Block Diagram of WDT1

15.1.3 Pin Configuration

Table 15-1 describes the WDT output pin.

Table 15-1 WDT Pin

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode
Buzzer output	BUZZ	Output	Outputs clock selected by watchdog timer (WDT1)

15.1.4 Register Configuration

Table 15-2 summarizes the WDT register configuration. These registers control clock selection, WDT mode switching, and the reset signal.

Table 15-2 WDT Registers

					Address*[1]	
Channel	**Name**	**Abbreviation**	**R/W**	**Initial Value**	**Write*[2]**	**Read**
0	Timer control/status register 0	TCSR0	R/(W)*[3]	H'18	H'FF74	H'FF74
	Timer counter 0	TCNT0	R/W	H'00	H'FF74	H'FF75
	Reset control/status register	RSTCSR	R/(W)*[3]	H'1F	H'FF76	H'FF77
1	Timer control/status register 1	TCSR1	R/(W)*[3]	H'00	H'FFA2	H'FFA2
	Timer counter 1	TCNT1	R/W	H'00	H'FFA2	H'FFA3
All	Pin function control register	PFCR	R/W	H'0D/H'00	H'FDEB	

Notes: *1 Lower 16 bits of the address.
*2 For details of write operations, see section 15.2.4, Notes on Register Access.
*3 Only a write of 0 is permitted to bit 7, to clear the flag.

15.2 Register Descriptions

15.2.1 Timer Counter (TCNT)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or an interval timer interrupt (WOVI) is generated, depending on the mode selected by the WT/$\overline{\text{IT}}$ bit in TCSR.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit is cleared to 0. It is not initialized in software standby mode.

Note: * TCNT is write-protected by a password to prevent accidental overwriting. For details see section 15.2.5, Notes on Register Access.

15.2.2 Timer Control/Status Register (TCSR)

TCSR0

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	:	0	0	0	1	1	0	0	0
R/W	:	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Note: * Only a 0 can be written, for flag clearing.

TCSR1

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{\text{IT}}$	TME	PSS	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only a 0 can be written, for flag clearing.

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

TCSR0 (TCSR1) is initialized to H'18 (H'00) by a reset and in hardware standby mode. It is not initialized in software standby mode.

Note: * TCSR is write-protected by a password to prevent accidental overwriting. For details see section 15.2.5, Notes on Register Access.

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00.

Bit 7 OVF	Description	
0	[Clearing conditions]	(Initial value)
	• Cleared when 0 is written to the TME bit (Only applies to WDT1)	
	• Cleared by reading TCSR when OVF = 1, then writing 0 to OVF	
1	[Setting condition]	
	When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. When TCNT overflows, WDT0 generates the $\overline{WDTOVF}$ signal when in watchdog timer mode, or a WOVI interrupt request to the CPU when in interval timer mode. WDT1 generates a reset or NMI interrupt request when in watchdog timer mode, or a WOVI interrupt request to the CPU when in interval timer mode.

WDT0 Mode Select

WDT0 WT/$\overline{IT}$	Description	
0	Interval timer mode: WDT0 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.	(Initial value)
1	Watchdog timer mode: WDT0 outputs a $\overline{WDTOVF}$ signal when the TCNT overflows.*	

Note: * For details on a TCNT overflow in watchdog timer mode, see section 15.2.3, Reset Control/Status Register (RSTCSR).

WDT1 Mode Select

WDT1 WT/$\overline{IT}$	Description	
0	Interval timer mode: WDT1 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.	(Initial value)
1	Watchdog timer mode: WDT1 requests a reset or an NMI interrupt from the CPU when the TCNT overflows.	

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5 TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT counts	

WDT0 TCSR Bit 4—Reserved Bit: This bit is always read as 1 and cannot be modified.

WDT1 TCSR Bit 4—Prescaler Select (PSS): This bit is used to select an input clock source for the TCNT of WDT1.

See the descriptions of Clock Select 2 to 0 for details.

WDT1 TCSR Bit 4

PSS	Description	
0	The TCNT counts frequency-division clock pulses of the ø based prescaler (PSM).	(Initial value)
1	The TCNT counts frequency-division clock pulses of the ø SUB-based prescaler (PSS).	

WDT0 TCSR Bit 3—Reserved Bit: This bit is always read as 1 and cannot be modified.

WDT1 TCSR Bit 3—Reset or NMI (RST/$\overline{\text{NMI}}$): This bit is used to choose between an internal reset request and an NMI request when the TCNT overflows during the watchdog timer mode.

Bit 3	
RTS/$\overline{\text{NMI}}$	**Description**
0	NMI request. (Initial value)
1	Internal reset request.

Bits 2 to 0: Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources, obtained by dividing the system clock (ø) or subclock (ø SUB), for input to TCNT.

WDT0 Input Clock Select

			Description	
Bit 2 CKS2	**Bit 1 CKS1**	**Bit 0 CKS0**	**Clock**	**Overflow Period* (where ø = 25 MHz)**
0	0	0	ø/2 (initial value)	20.4 µs
		1	ø/64	655.3 µs
	1	0	ø/128	1.3 ms
		1	ø/512	5.2 ms
1	0	0	ø/2048	20.9 ms
		1	ø/8192	83.8 ms
	1	0	ø/32768	335.5 ms
		1	ø/131072	1.34 s

Note: * An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.

WDT1 Input Clock Select

				Description	
Bit 4 PSS	**Bit 2 CKS2**	**Bit 1 CKS1**	**Bit 0 CKS0**	**Clock**	**Overflow Period* (where ø = 25 MHz) (where ø SUB = 32.768 kHz)**
0	0	0	0	ø/2 (initial value)	20.4 µs
			1	ø/64	655.3 µs
		1	0	ø/128	1.3 ms
			1	ø/512	5.2 ms
	1	0	0	ø/2048	20.9 ms
			1	ø/8192	83.8 ms
		1	0	ø/32768	335.5 ms
			1	ø/131072	1.34 s
1	0	0	0	øSUB/2	15.6 ms
			1	øSUB/4	31.3 ms
		1	0	øSUB/8	62.5 ms
			1	øSUB/16	125 ms
	1	0	0	øSUB/32	250 ms
			1	øSUB/64	500 ms
		1	0	øSUB/128	1 s
			1	øSUB/256	2 s

Note: * An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.

15.2.3 Reset Control/Status Register (RSTCSR)

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

Note: * Only 0 can be written, for flag clearing.

RSTCSR is an 8-bit readable/writable* register that controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal.

RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT internal reset signal caused by overflows.

Note: * RSTCSR is write-protected by a password to prevent accidental overwriting. For details see section 15.2.5, Notes on Register Access.

Bit 7—Watchdog Overflow Flag (WOVF): Indicates that TCNT has overflowed (changed from H'FF to H'00) during watchdog timer operation. This bit is not set in interval timer mode.

Bit 7

WOVF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading TCSR when WOVF = 1, then writing 0 to WOVF	
1	[Setting condition]	
	Set when TCNT overflows (changed from H'FF to H'00) during watchdog timer operation	

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is generated in the H8S/2633 Series if TCNT overflows during watchdog timer operation.

Bit 6

RSTE	Description	
0	Reset signal is not generated if TCNT overflows*	(Initial value)
1	Reset signal is generated if TCNT overflows	

Note: * The modules within the H8S/2633 Series are not reset, but TCNT and TCSR within the WDT are reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT overflows during watchdog timer operation.

For details of the types of reset, see section 4, Exception Handling.

Bit 5 RSTS	Description	
0	Power-on reset	(Initial value)
1	Manual reset	

Bits 4 to 0—Reserved: These bits are always read as 1 and cannot be modified.

15.2.4 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		CSS07	CSS36	BUZZE	LCASS	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable/writable register that performs address output control in external expanded mode.

Only bit 5 is described here. For details of the other bits, see section 7.2.6, Pin Function Control Register (PFCR).

Bit 5—BUZZ Output Enable (BUZZE): Enables or disables BUZZ output from the PF1 pin. The WDT1 input clock selected with bits PSS and CKS2 to CKS0 is output as the BUZZ signal.

Bit 5 BUZZE	Description	
0	Functions as PF1 I/O pin	(Initial value)
1	Functions as BUZZ output pin	

15.2.5 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written to by a word transfer instruction. They cannot be written to with byte instructions.

Figure 15-2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

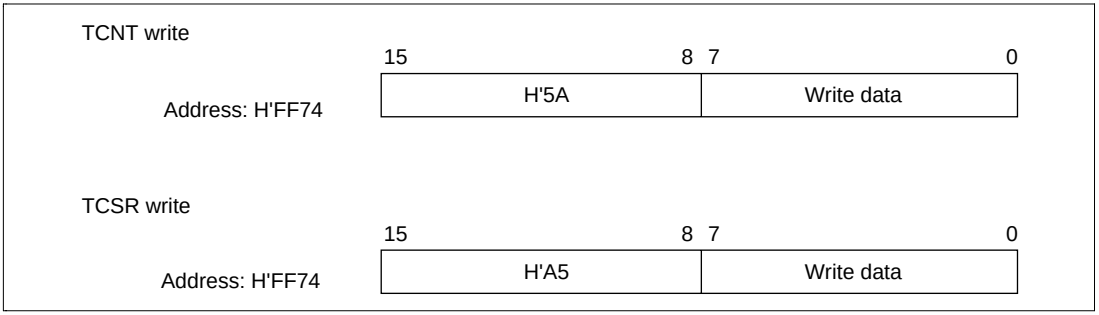


Figure 15-2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written to by word transfer instruction to address H'FF76. It cannot be written to with byte instructions.

Figure 15-3 shows the format of data written to RSTCSR. The method of writing 0 to the WOVF bit differs from that for writing to the RSTE and RSTS bits.

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and RSTS bits, but has no effect on the WOVF bit.

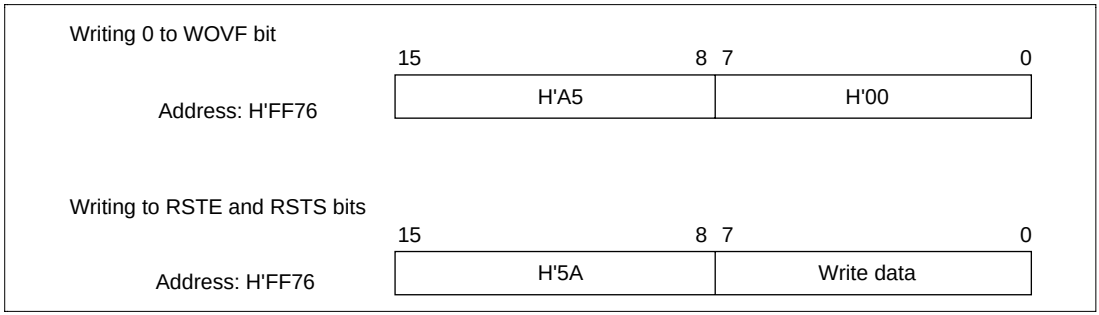


Figure 15-3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR (WDT0): These registers are read in the same way as other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

15.3 Operation

15.3.1 Watchdog Timer Operation

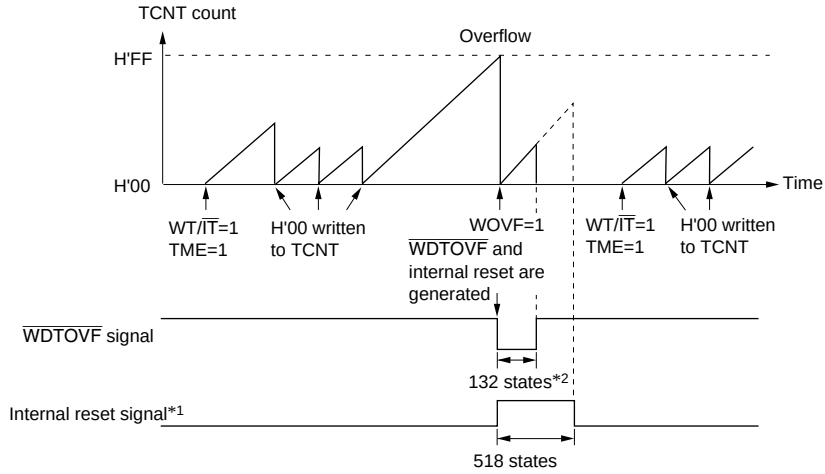
To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit in TCSR and TME bit to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system crash or other error, in the WDT0 the $\overline{WDTOVF}$ signal is output. This is shown in figure 15-4. This $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets the H8S/2633 Series internally is generated at the same time as the $\overline{WDTOVF}$ signal. This reset can be selected as a power-on reset or a manual reset, depending on the setting of the RSTS bit in RSTCSR. The internal reset signal is output for 518 states.

If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

In the case of WDT1, the chip is reset, or an NMI interrupt request is generated, for 516 system clock periods (516ø) (515 or 516 states when the clock source is øSUB (PSS = 1)). This is illustrated in figure 15-4 (b).

An NMI request from the watchdog timer and an interrupt request from the NMI pin are both treated as having the same vector. So, avoid handling an NMI request from the watchdog timer and an interrupt request from the NMI pin at the same time.

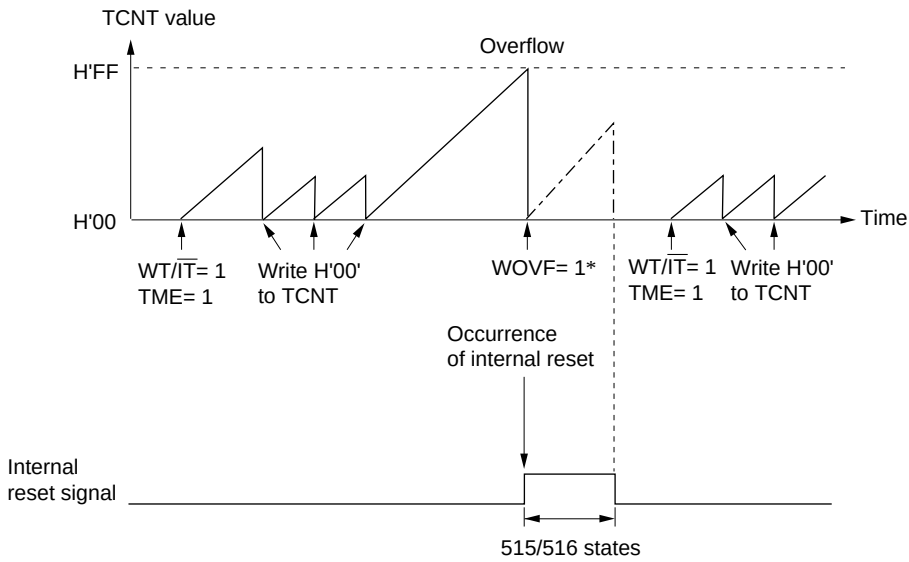


Legend

WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Notes: *1 The internal reset signal is generated only if the RSTE bit is set to 1.
*2 130 states when the RSTE bit is cleared to 0.

Figure 15-4 (a) WDT0 Watchdog Timer Operation



WT/$\overline{IT}$: Timer Mode Select bit
TME : Timer Enable bit

Note: * The WOVF bit is set to 1 and then cleared to 0 by an internal reset.

Figure 15-4 (b) WDT1 Operation in Watchdog Timer Mode

15.3.2 Interval Timer Operation

To use the WDT as an interval timer, clear the WT/$\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 15-5. This function can be used to generate interrupt requests at regular intervals.

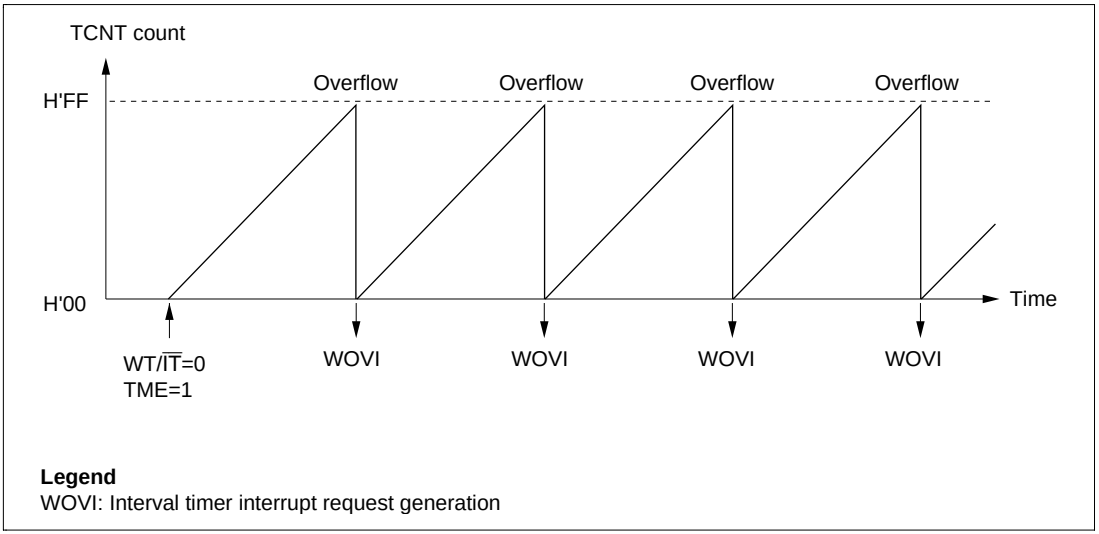


Figure 15-5 Interval Timer Operation

15.3.3 Timing of Setting Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 15-6.

With WDT1, the OVF bit of the TCSR is set to 1 and a simultaneous NMI interrupt is requested when the TCNT overflows if the NMI request has been chosen in the watchdog timer mode.

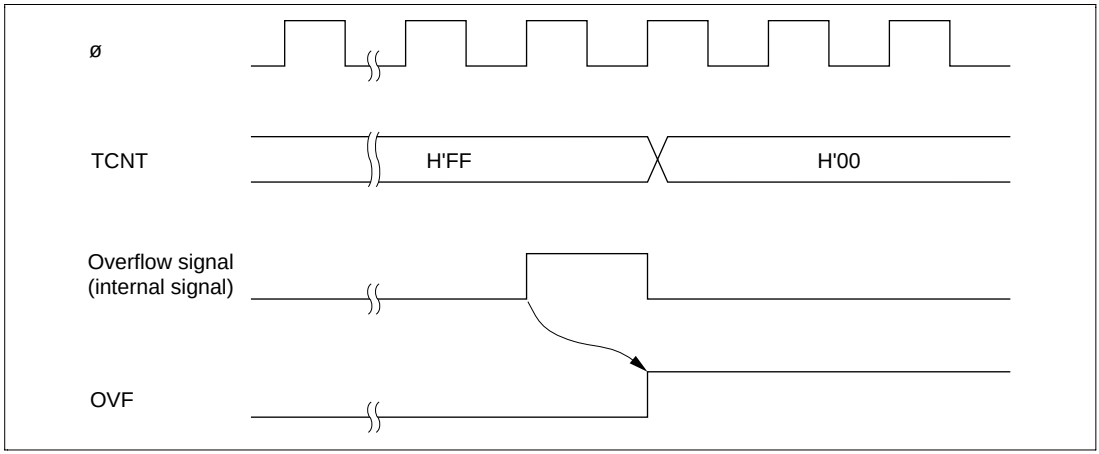


Figure 15-6 Timing of Setting of OVF

15.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

In the WDT0, the WOVF flag is set to 1 if TCNT overflows during watchdog timer operation. At the same time, the $\overline{\text{WDTOVF}}$ signal goes low. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire H8S/2633 Series chip. Figure 15-7 shows the timing in this case.

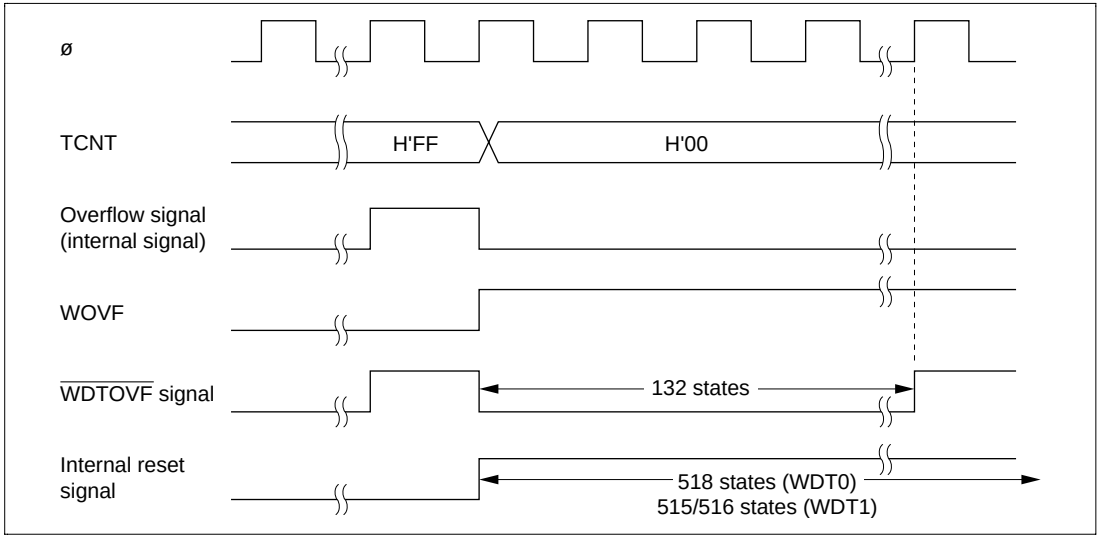


Figure 15-7 Timing of Setting of WOVF

15.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

If an NMI request has been chosen in the watchdog timer mode, an NMI request is generated when a TCNT overflow occurs.

15.5 Usage Notes

15.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 15-8 shows this operation.

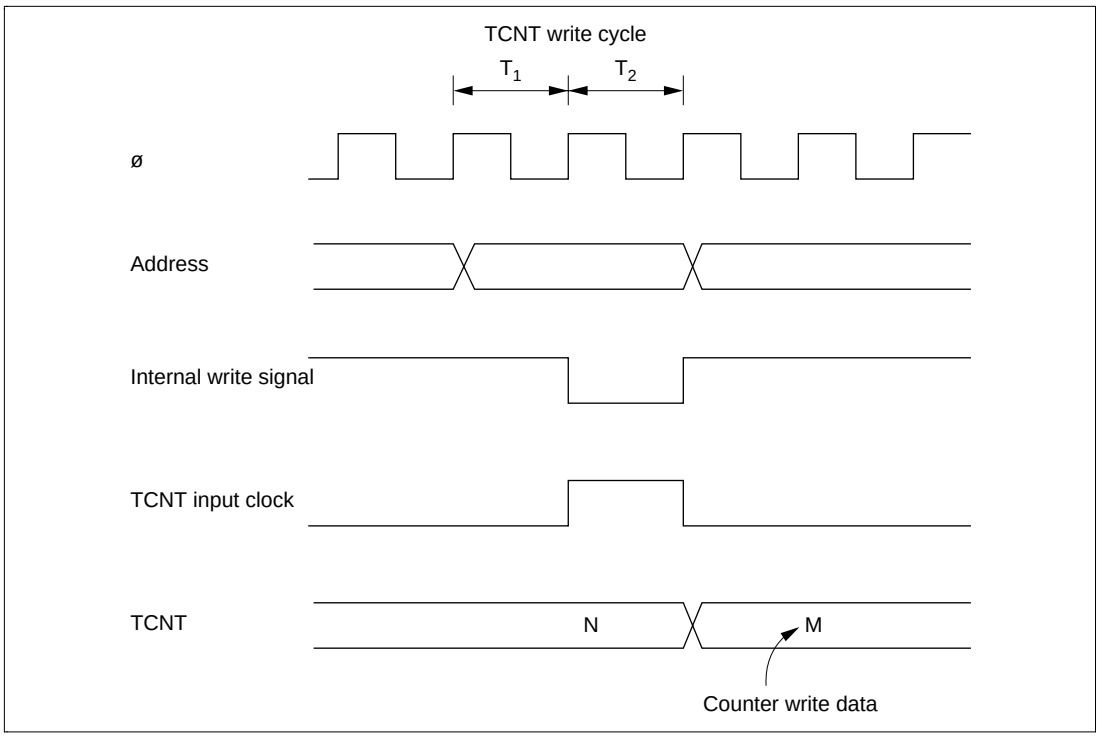


Figure 15-8 Contention between TCNT Write and Increment

15.5.2 Changing Value of PSS and CKS2 to CKS0

If bits PSS and CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits PSS and CKS2 to CKS0.

15.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

15.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin of the H8S/2633 Series, the H8S/2633 Series will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 15-9.

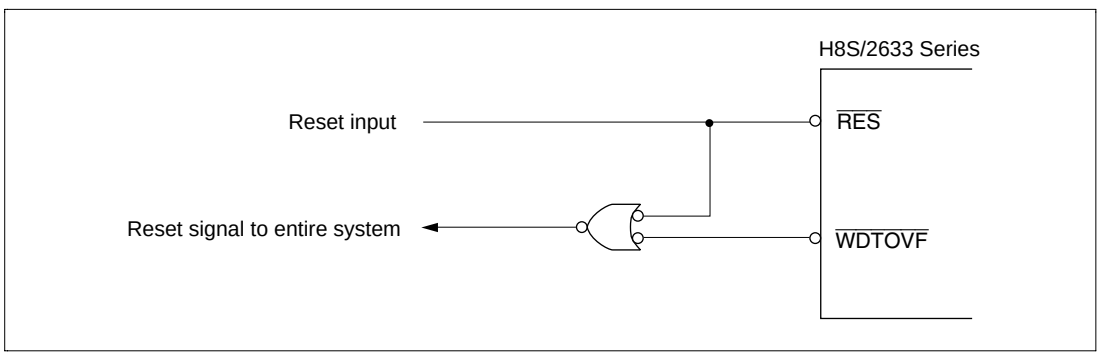


Figure 15-9 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

15.5.5 Internal Reset in Watchdog Timer Mode

The H8S/2633 Series is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCSR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF falg, therefore, read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

Section 16 Serial Communication Interface (SCI, IrDA)

16.1 Overview

The H8S/2633 is equipped with 5 independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

One of the five SCI channels is capable of sending and receiving IrDA communications waveforms (based on IrDA Version 1.0).

16.1.1 Features

SCI features are listed below.

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode
  - Serial data communication executed using asynchronous system in which synchronization is achieved character by character

    Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)
  - A multiprocessor communication function is provided that enables serial data communication with a number of processors
  - Choice of 12 serial data transfer formats

    Data length : 7 or 8 bits
    Stop bit length : 1 or 2 bits
    Parity : Even, odd, or none
    Multiprocessor bit : 1 or 0
  - Receive error detection : Parity, overrun, and framing errors
  - Break detection : Break can be detected by reading the RxD pin level directly in case of a framing error

  Clocked Synchronous mode
  - Serial data communication synchronized with a clock

    Serial data communication can be carried out with other chips that have a synchronous communication function
  - One serial data transfer format

    Data length : 8 bits

— Receive error detection : Overrun errors detected

- Full-duplex communication capability
  — The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously
  — Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data

- Choice of LSB-first or MSB-first transfer
  — Can be selected regardless of the communication mode* (except in the case of asynchronous mode 7-bit data)

Note: * Descriptions in this section refer to LSB-first transfer.

- On-chip baud rate generator allows any bit rate to be selected

- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin

- Four interrupt sources
  — Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, and receive error — that can issue requests independently
  — The transmit-data-empty interrupt and receive data full interrupts can activate the DMA controller (DMAC) or data transfer controller (DTC) to execute data transfer

- Module stop mode can be set
  — As the initial setting, SCI operation is halted. Register access is enabled by exiting module stop mode.

16.1.2 Block Diagram

Figure 16-1 shows a block diagram of the SCI.

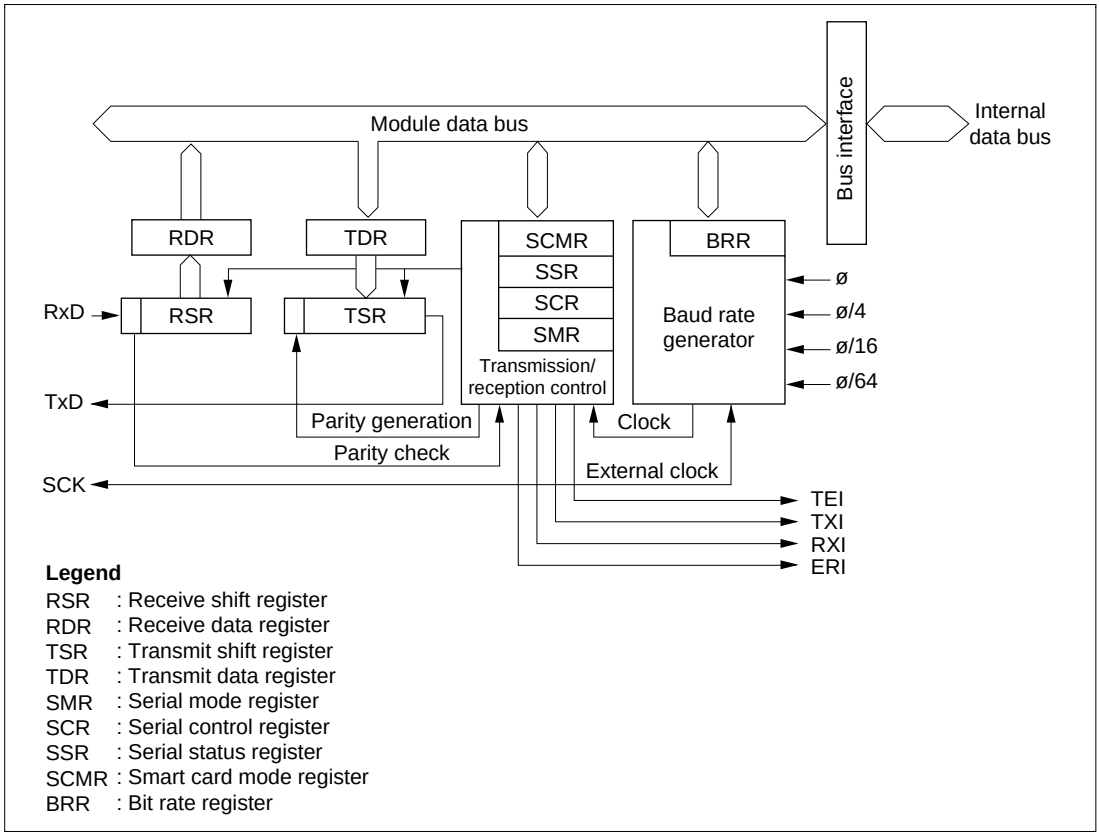


Figure 16-1 Block Diagram of SCI

16.1.3 Pin Configuration

Table 16-1 shows the serial pins for each SCI channel.

Table 16-1 SCI Pins

Channel	Pin Name	Symbol*	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0/IrRxD	Input	SCI0 receive data input (normal/IrDA)
	Transmit data pin 0	TxD0/IrTxD	Output	SCI0 transmit data output (normal/IrDA)
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data output
3	Serial clock pin 3	SCK3	I/O	SCI3 clock input/output
	Receive data pin 3	RxD3	Input	SCI3 receive data input
	Transmit data pin 3	TxD3	Output	SCI3 transmit data output
4	Serial clock pin 4	SCK4	I/O	SCI4 clock input/output
	Receive data pin 4	RxD4	Input	SCI4 receive data input
	Transmit data pin 4	TxD4	Output	SCI4 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

16.1.4 Register Configuration

The SCI has the internal registers shown in table 16-2. These registers are used to specify asynchronous mode or clocked synchronous mode, the data format , and the bit rate, and to control transmitter/receiver.

Table 16-2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78*[3]
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79*[3]
	Serial control register 0	SCR0	R/W	H'00	H'FF7A*[3]
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B*[3]
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FF7C*[3]
	Receive data register 0	RDR0	R	H'00	H'FF7D*[3]
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E*[3]
	IrDA control register	IrCR	R/W	H'00	H'FDB0
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80*[3]
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81*[3]
	Serial control register 1	SCR1	R/W	H'00	H'FF82*[3]
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83*[3]
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF84*[3]
	Receive data register 1	RDR1	R	H'00	H'FF85*[3]
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86*[3]
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*[2]	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
3	Serial mode register 3	SMR3	R/W	H'00	H'FDD0
	Bit rate register 3	BRR3	R/W	H'FF	H'FDD1
	Serial control register 3	SCR3	R/W	H'00	H'FDD2
	Transmit data register 3	TDR3	R/W	H'FF	H'FDD3
	Serial status register 3	SSR3	R/(W)*[2]	H'84	H'FDD4
	Receive data register 3	RDR3	R	H'00	H'FDD5
	Smart card mode register 3	SCMR3	R/W	H'F2	H'FDD6
4	Serial mode register 4	SMR4	R/W	H'00	H'FDD8
	Bit rate register 4	BRR4	R/W	H'FF	H'FDD9
	Serial control register 4	SCR4	R/W	H'00	H'FDDA
	Transmit data register 4	TDR4	R/W	H'FF	H'FDDB
	Serial status register 4	SSR4	R/(W)*[2]	H'84	H'FDDC
	Receive data register 4	RDR4	R	H'00	H'FDDD
	Smart card mode register 4	SCMR4	R/W	H'F2	H'FDDE
All	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9
	Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA

Notes: *1 Lower 16 bits of the address.

*2 Only 0 can be written, for flag clearing.

*3 Some of the SCI registers are allocated to the same addresses as other registers. The IICE bit of the serial timer control register X (SCRX) selects the respective registers.

16.2 Register Descriptions

16.2.1 Receive Shift Register (RSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

16.2.2 Receive Data Register (RDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

RDR is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored, and completes the receive operation. After this, RSR is receive-enabled.

Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

16.2.3 Transmit Shift Register (TSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

16.2.4 Transmit Data Register (TDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

16.2.5 Serial Mode Register (SMR)

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchronous mode as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In clocked synchronous mode with a multiprocessor format, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note:* When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronous mode, when parity addition and checking is disabled in asynchronous mode, and when a multiprocessor format is used.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: *1 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.

*2 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bits setting is only valid in asynchronous mode. If clocked synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

Bit 3

STOP	Description	
0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.	(Initial value)
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.	

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in clocked synchronous mode.

For details of the multiprocessor communication function, see section 16.3.3, Multiprocessor Communication Function.

Bit 2

MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ø, ø/4, ø/16, and ø/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 16.2.8, Bit Rate Register.

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ø clock	(Initial value)
	1	ø/4 clock	
1	0	ø/16 clock	
	1	ø/64 clock	

16.2.6 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7		
TIE	**Description**	
0	Transmit data empty interrupt (TXI) requests disabled	(Initial value)
1	Transmit data empty interrupt (TXI) requests enabled	

Note: TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive data full interrupt (RXI) request and receive error interrupt (ERI) request generation when serial receive data is transferred from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6

RIE	Description
0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled* (Initial value)
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Note:* RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5

TE	Description
0	Transmission disabled*[1] (Initial value)
1	Transmission enabled*[2]

Notes: *1 The TDRE flag in SSR is fixed at 1.

*2 In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.

SMR setting must be performed to decide the transfer format before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4

RE	Description
0	Reception disabled*[1] (Initial value)
1	Reception enabled*[2]

Notes: *1 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

*2 Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode.

SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when the MP bit in SMR is set to 1.

The MPIE bit setting is invalid in clocked synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end interrupt (TEI) request generation when there is no valid transmit data in TDR in MSB data transmission.

Bit 2

TEIE	Description	
0	Transmit end interrupt (TEI) request disabled*	(Initial value)
1	Transmit end interrupt (TEI) request enabled*	

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in clocked synchronous mode, and in the case of external clock operation (CKE1 = 1). Note that the SCI's operating mode must be decided using SMR before setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 16-9 in section 16.3, Operation.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*[1]
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output*[1]
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*[2]
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Initial value
*2 Outputs a clock of the same frequency as the bit rate.
*3 Inputs a clock with a frequency 16 times the bit rate.

16.2.7 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written, to clear the flag.

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from TDR to TSR and the next serial data can be written to TDR.

Bit 7 TDRE	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6

RDRF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1*2

Notes: *1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
*2 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (Initial value)*1 • When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data when reception ends, and the stop bit is 0 *2

Notes: *1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: *1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not transmitting, and in clocked synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (Initial value)
1	Data with a 1 multiprocessor bit is transmitted

16.2.8 Bit Rate Register (BRR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in standby mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 16-3 shows sample BRR settings in asynchronous mode, and table 16-4 shows sample BRR settings in clocked synchronous mode.

Table 16-3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	ø = 2 MHz			ø = 2.097152 MHz			ø = 2.4576 MHz			ø = 3 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

	ø = 3.6864 MHz			ø = 4 MHz			ø = 4.9152 MHz			ø = 5 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

Bit Rate (bit/s)	ø = 6 MHz			ø = 6.144 MHz			ø = 7.3728 MHz			ø = 8 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

Bit Rate (bit/s)	ø = 9.8304 MHz			ø = 10 MHz			ø = 12 MHz			ø = 12.288 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bit/s)	ø = 14 MHz n	N	Error (%)	ø = 14.7456 MHz n	N	Error (%)	ø = 16 MHz n	N	Error (%)	ø = 17.2032 MHz n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.13	0	13	0.00

Bit Rate (bit/s)	ø = 18 MHz n	N	Error (%)	ø = 19.6608 MHz n	N	Error (%)	ø = 20 MHz n	N	Error (%)	ø = 25 MHz n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	110	–0.02
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	–0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	0.15
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	–0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	0.15
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	–0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	0.15
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80	–0.47
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40	–0.76
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	1.73

Table 16-4 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	ø = 2 MHz		ø = 4 MHz		ø = 8 MHz		ø = 10 MHz		ø = 16 MHz		ø = 20 MHz		ø = 25 MHz	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—										
250	2	124	2	249	3	124	—	—	3	249				
500	1	249	2	124	2	249	—	—	3	124	—	—		
1 k	1	124	1	249	2	124	—	—	2	249	—	—	3	97
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124	2	155
5 k	0	99	0	199	1	99	1	124	1	199	1	249	2	77
10 k	0	49	0	99	0	199	0	249	1	99	1	124	1	155
25 k	0	19	0	39	0	79	0	99	0	159	0	199	0	249
50 k	0	9	0	19	0	39	0	49	0	79	0	99	0	124
100 k	0	4	0	9	0	19	0	24	0	39	0	49	0	62
250 k	0	1	0	3	0	7	0	9	0	15	0	19	0	24
500 k	0	0*	0	1	0	3	0	4	0	7	0	9	—	—
1 M			0	0*	0	1			0	3	0	4	—	—
2.5 M							0	0*			0	1	—	—
5 M											0	0*	—	—

Note: As far as possible, the setting should be made so that the error is no more than 1%.

Legend

Blank : Cannot be set.

— : Can be set, but there will be a degree of error.

* : Continuous transfer is not possible.

The BRR setting is found from the following formulas.

Asynchronous mode:

$$N = \frac{ø}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{ø}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

The bit rate error in asynchronous mode is found from the following formula:

$$\text{Error (\%)} = \left\{ \frac{ø \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 16-5 shows the maximum bit rate for each frequency in asynchronous mode. Tables 16-6 and 16-7 show the maximum bit rates with external clock input.

Table 16-5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ø (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0

Table 16-6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625

Table 16-7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7

16.2.9 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR selects LSB-first or MSB-first by means of bit SDIR. Except in the case of asynchronous mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial communication mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see 17.2.1, Smart Card Mode Register (SCMR).

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

This bit is valid when 8-bit data is used as the transmit/receive format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2

SINV	Description	
0	TDR contents are transmitted without modification Receive data is stored in RDR without modification	(Initial value)
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form	

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface operates as a normal SCI, 0 should be written in this bit.

Bit 0

SMIF	Description	
0	Operates as normal SCI (smart card interface function disabled)	(Initial value)
1	Smart card interface function enabled	

16.2.10 IrDA Control Register (IrCR)

Bit	:	7	6	5	4	3	2	1	0
		IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

IrCR is an 8-bit read/write register that selects the SCI0 function.

IrCR is initialized to H'00 when in hardware standby mode.

Bit 7—IrDA enable (IrE): Sets SCI0 input and output for normal SCI operation or IrDA operation.

Bit 7

IrE	Description	
0	TxD0/IrTxD and RxD0/IrRxD pins operate as TxD0 and RxD0.	(Initial value)
1	TxD0/IrTxD and RxD0/IrRxD pins operate as IrTxD and IrRxD.	

Bits 6 to 4—IrDA clock select 2 to 0 (IrCKS2 to IrCKS0): When the IrDA function is enabled, these bits set the width of the High pulse when encoding the IrTxD output pulse.

Bit 6	Bit 5	Bit 4		
IrCKS2	**IrCKS1**	**IrCKS0**	**Description**	
0	0	0	B×3/16 (three sixteenths of bit rate)	(Initial value)
		1	ø/2	
	1	0	ø/4	
		1	ø/8	
1	0	0	ø/16	
		1	ø/32	
	1	0	ø/64	
		1	ø/128	

Bits 3 to 0—Reserved: These bits are always read as 0 and cannot be modified.

16.2.11 Module Stop Control Registers B and C (MSTPCRB, MSTPCRC)

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	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

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	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

MSTPCRB and MSTPCRC are 8-bit readable/writable registers that perform module stop mode control.

Setting any of bits MSTPB7 to MSTBP5 and MSTPC7 and MSTPC6 to 1 stops SCI0 to SCI4 operating and enter module stop mode on completion of the bus cycle. For details, see section 24.5, Module Stop Mode.

MSTPCRB and MSTPCRC are initialized to H'FF by a reset and in hardware standby mode. They are not initialized by a manual reset and in software standby mode.

(1) Module Stop Control Register B (MSTPCRB)

Bit 7—Module Stop (MSTPB7): Specifies the SCI0 module stop mode.

Bit 7		
MSTPB7	**Description**	
0	SCI0 module stop mode is cleared	
1	SCI0 module stop mode is set	(Initial value)

Bit 6—Module Stop (MSTPB6): Specifies the SCI1 module stop mode.

Bit 6		
MSTPB6	**Description**	
0	SCI1 module stop mode is cleared	
1	SCI1 module stop mode is set	(Initial value)

Bit 5—Module Stop (MSTPB5): Specifies the SCI2 module stop mode.

Bit 5		
MSTPB5	**Description**	
0	SCI2 module stop mode is cleared	
1	SCI2 module stop mode is set	(Initial value)

(2) Module Stop Control Register C (MSTPCRC)

Bit 7—Module Stop (MSTPC7): Specifies the SCI3 module stop mode.

Bit 7		
MSTPC7	**Description**	
0	SCI3 module stop mode is cleared	
1	SCI3 module stop mode is set	(Initial value)

Bit 6—Module Stop (MSTPC6): Specifies the SCI4 module stop mode.

Bit 6

MSTPC6	Description	
0	SCI4 module stop mode is cleared	
1	SCI4 module stop mode is set	(Initial value)

16.3 Operation

16.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and clocked synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or clocked synchronous mode and the transmission format is made using SMR as shown in table 16-8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 16-9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same frequency as the bit rate can be output
  - — When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used)

Clocked Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source

— When internal clock is selected:

The SCI operates on the baud rate generator clock and a serial clock is output off-chip

— When external clock is selected:

The on-chip baud rate generator is not used, and the SCI operates on the input serial clock

Table 16-8 SMR Settings and Serial Transfer Format Selection

SMR Settings						SCI Transfer Format			
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3		Data	Multi Processor	Parity	Stop Bit
C/$\overline{A}$	CHR	MP	PE	STOP	Mode	Length	Bit	Bit	Length
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Clocked synchronous mode	8-bit data	No		None

Table 16-9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock	
C/$\overline{A}$	CKE1	CKE0	Mode	Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

16.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16-2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

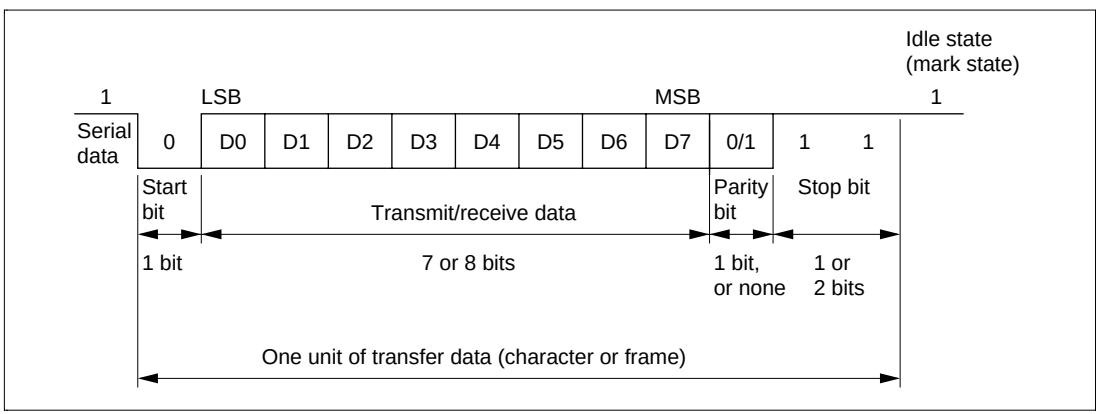


Figure 16-2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Data Transfer Format: Table 16-10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting.

Table 16-10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer 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

Legend

S : Start bit
STOP : Stop bit
P : Parity bit
MPB : Multiprocessor bit

Clock: Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 16-9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 16-3.

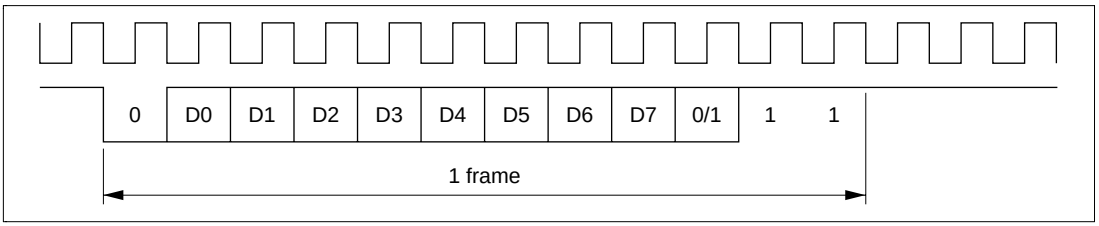


Figure 16-3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations:

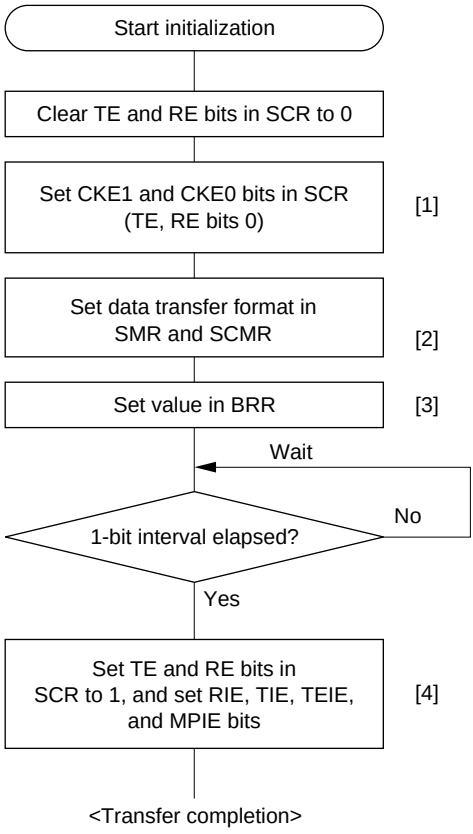
- SCI initialization (asynchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

  When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 16-4 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

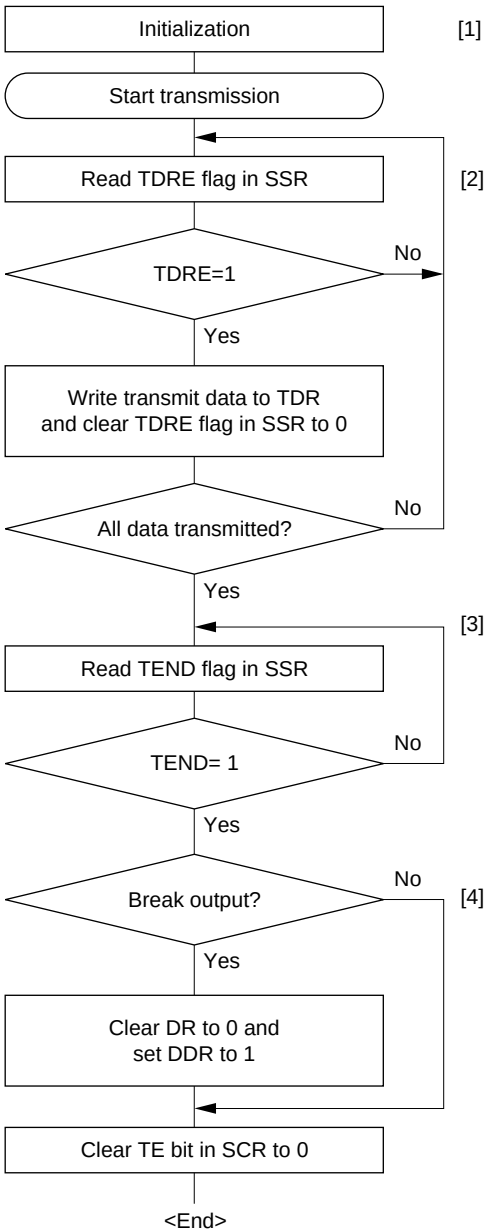
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 16-4 Sample SCI Initialization Flowchart

- Serial data transmission (asynchronous mode)

  Figure 16-5 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit data empty interrupt (TXI) request, and date is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 16-5 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Parity bit or multiprocessor bit:

One parity bit (even or odd parity), or one multiprocessor bit is output.

A format in which neither a parity bit nor a multiprocessor bit is output can also be selected.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 16-6 shows an example of the operation for transmission in asynchronous mode.

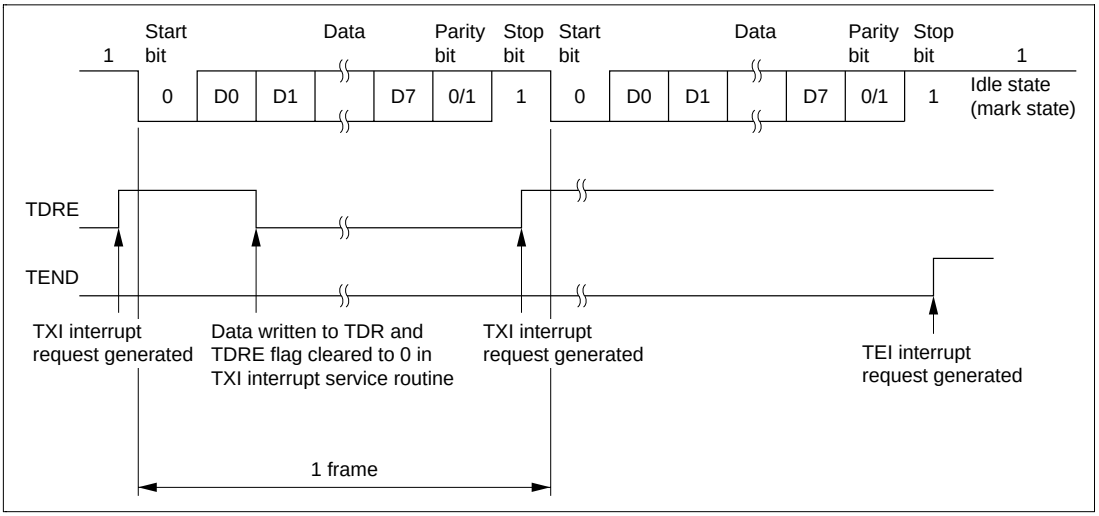
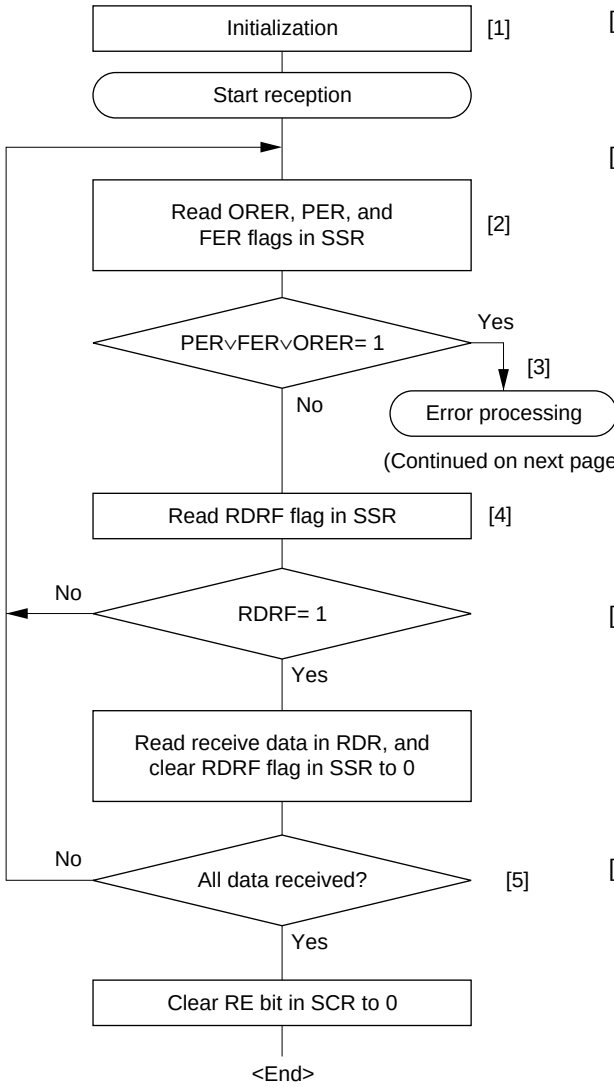


Figure 16-6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

- Serial data reception (asynchronous mode)

  Figure 16-7 shows a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DMAC or DTC is activated by an RXI interrupt and the RDR value is read.

Figure 16-7 Sample Serial Reception Data Flowchart

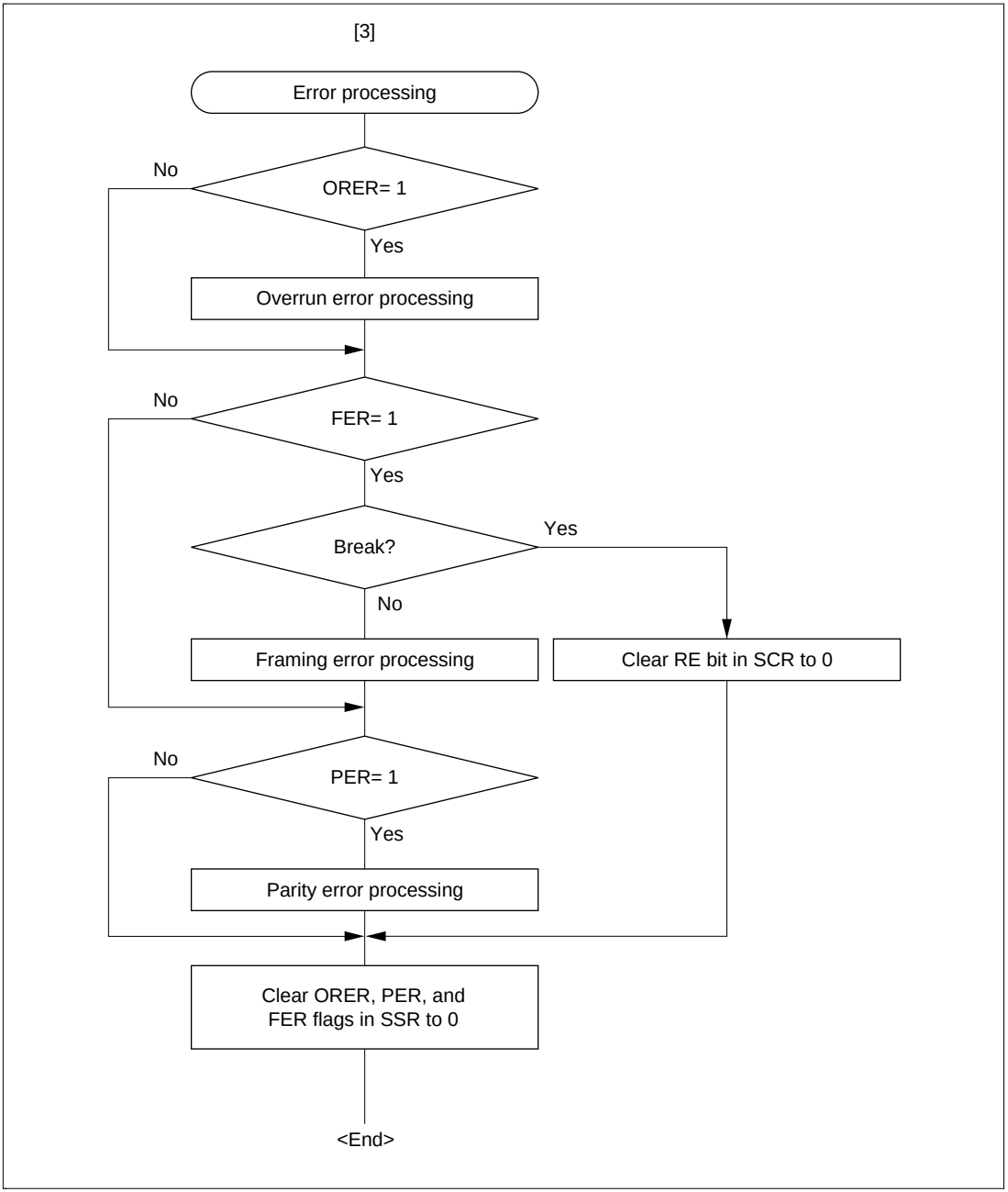


Figure 16-7 Sample Serial Reception Data Flowchart (cont)

In serial reception, the SCI operates as described below.

[1] The SCI monitors the transmission line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.

[2] The received data is stored in RSR in LSB-to-MSB order.

[3] The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks.

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{E}$ bit in SMR.

[b] Stop bit check:

The SCI checks whether the stop bit is 1.

If there are two stop bits, only the first is checked.

[c] Status check:

The SCI checks whether the RDRF flag is 0, indicating that the receive data can be transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

If a receive error* is detected in the error check, the operation is as shown in table 16-11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 16-11 Receive Errors and Conditions for Occurrence

Receive Error	Abbreviation	Occurrence Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR.
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR.
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR.

Figure 16-8 shows an example of the operation for reception in asynchronous mode.

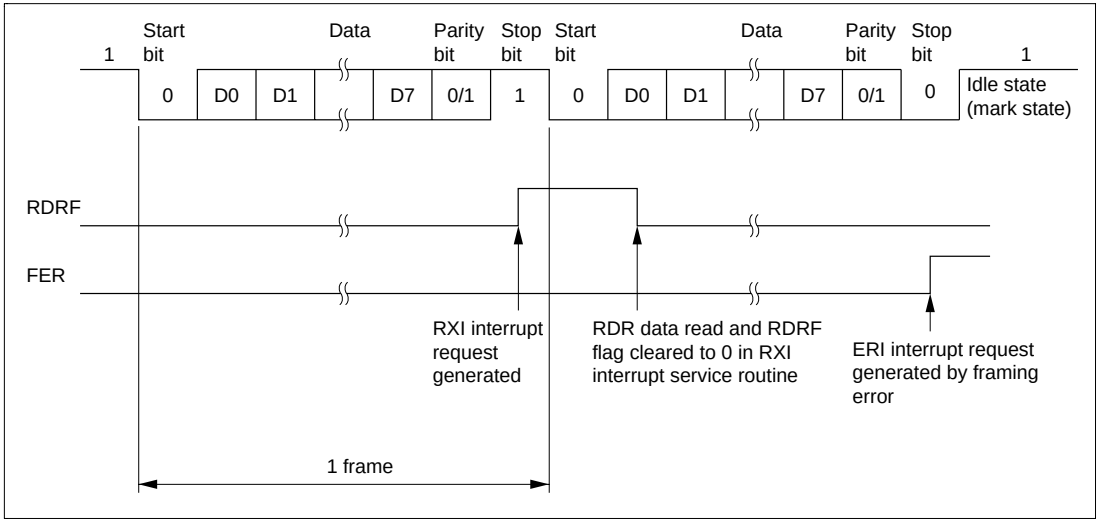


Figure 16-8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

16.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing transmission lines.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station , and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 16-9 shows an example of inter-processor communication using the multiprocessor format.

Data Transfer Format: There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 16-10.

Clock: See the section on asynchronous mode.

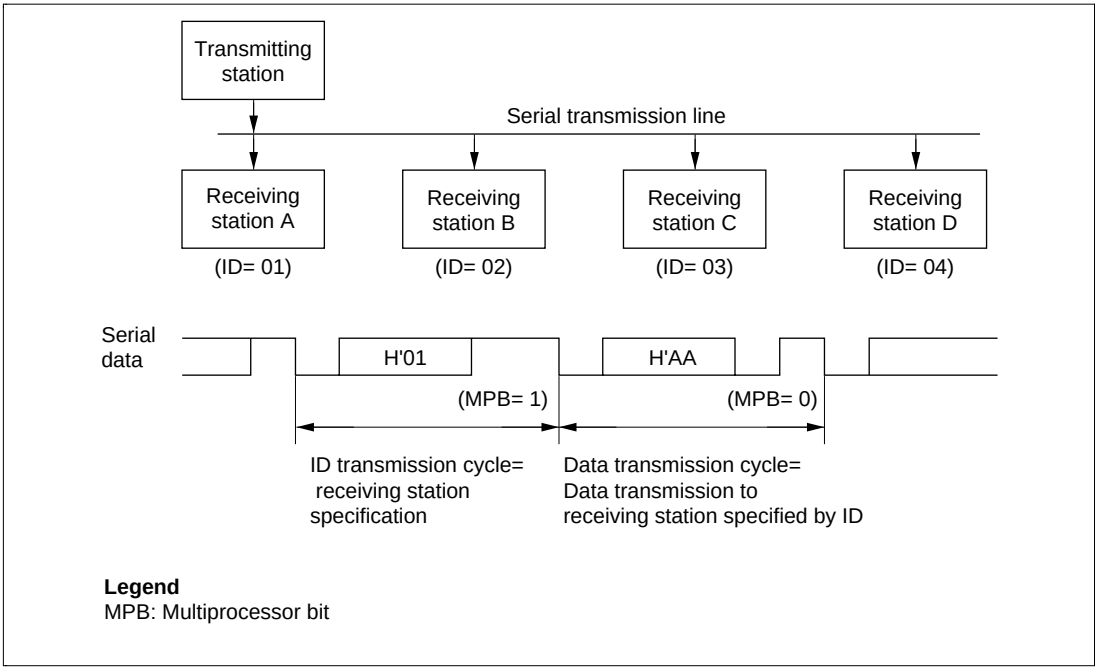


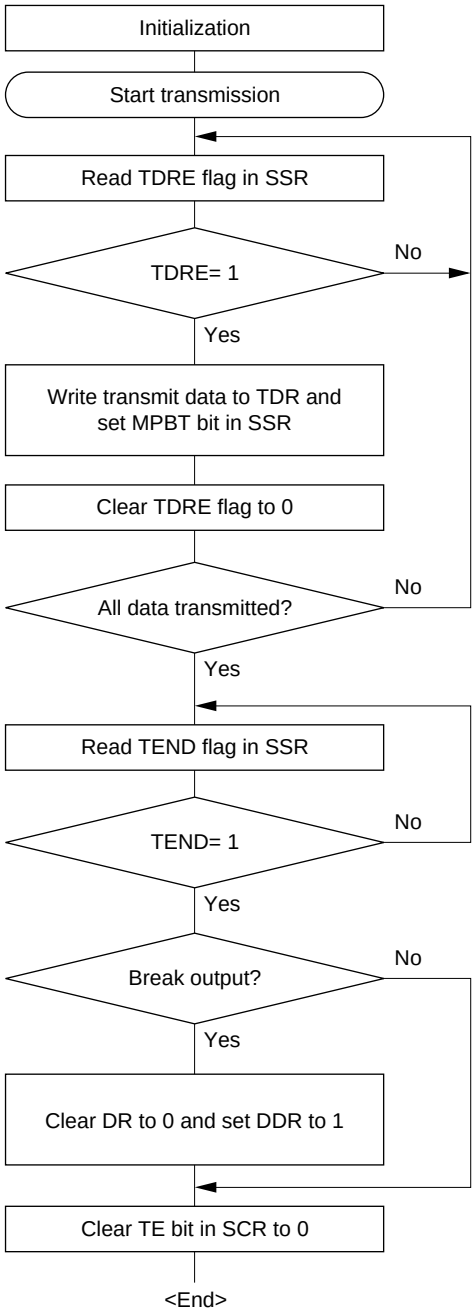
Figure 16-9 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations:

- Multiprocessor serial data transmission

  Figure 16-10 shows a sample flowchart for multiprocessor serial data transmission.

  The following procedure should be used for multiprocessor serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 16-10 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Multiprocessor bit

One multiprocessor bit (MPBT value) is output.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a transmission end interrupt (TEI) request is generated.

Figure 16-11 shows an example of SCI operation for transmission using the multiprocessor format.

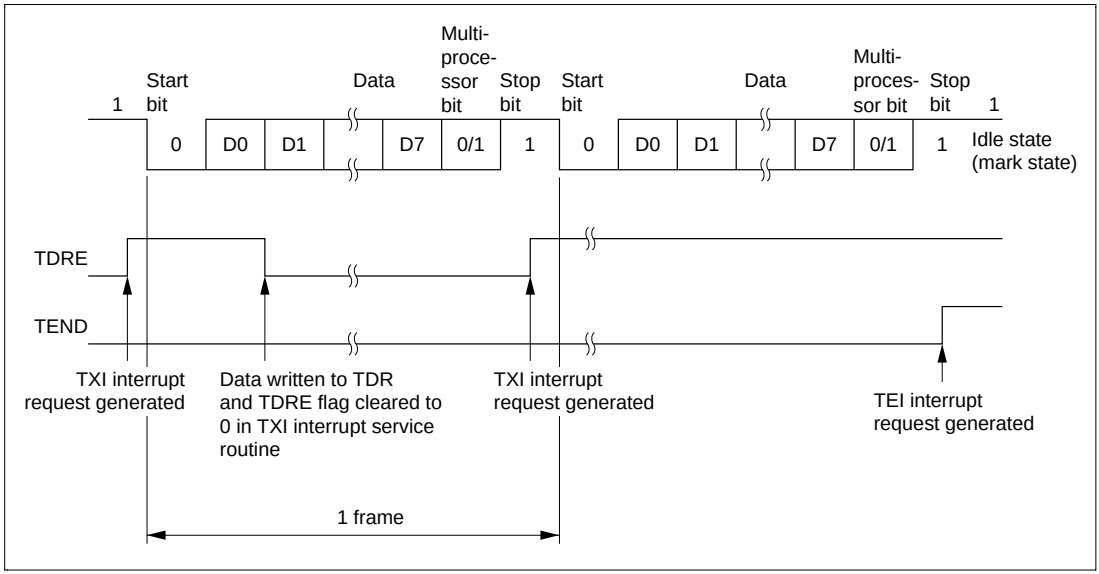
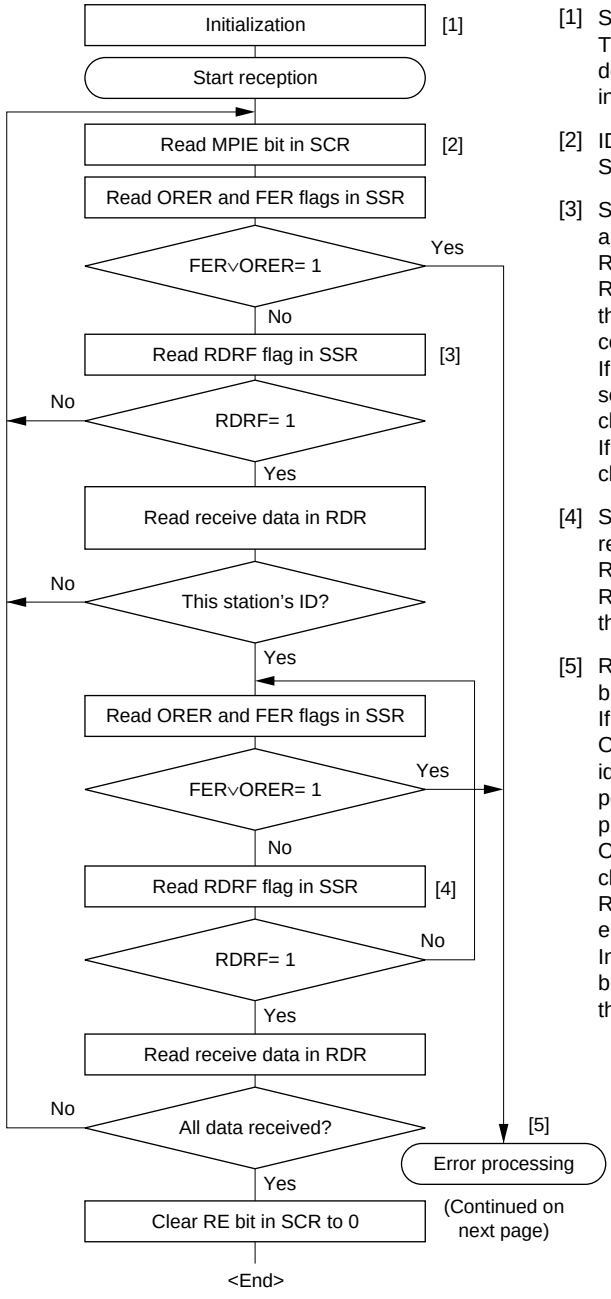


Figure 16-11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

- Multiprocessor serial data reception

  Figure 16-12 shows a sample flowchart for multiprocessor serial reception.

  The following procedure should be used for multiprocessor serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 16-12 Sample Multiprocessor Serial Reception Flowchart

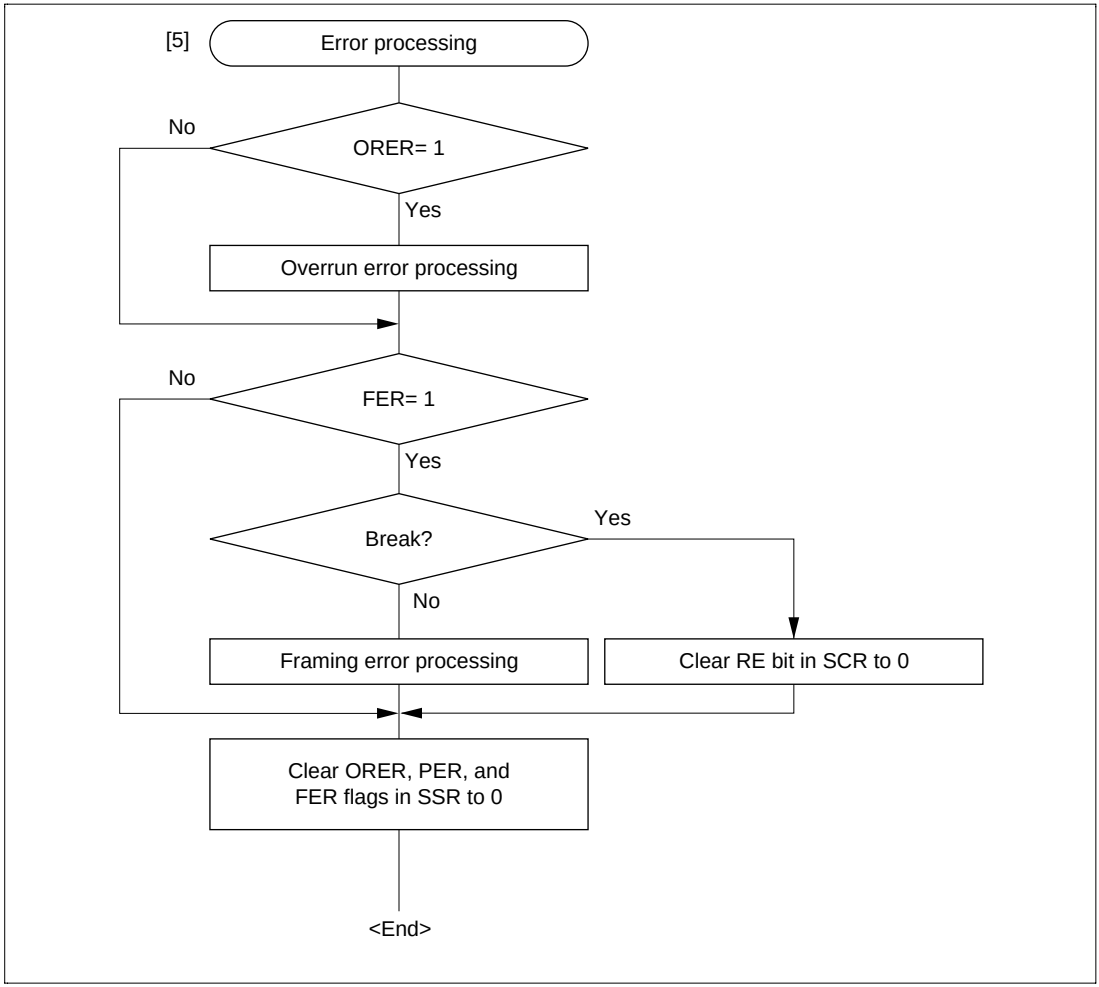


Figure 16-12 Sample Multiprocessor Serial Reception Flowchart (cont)

Figure 16-13 shows an example of SCI operation for multiprocessor format reception.

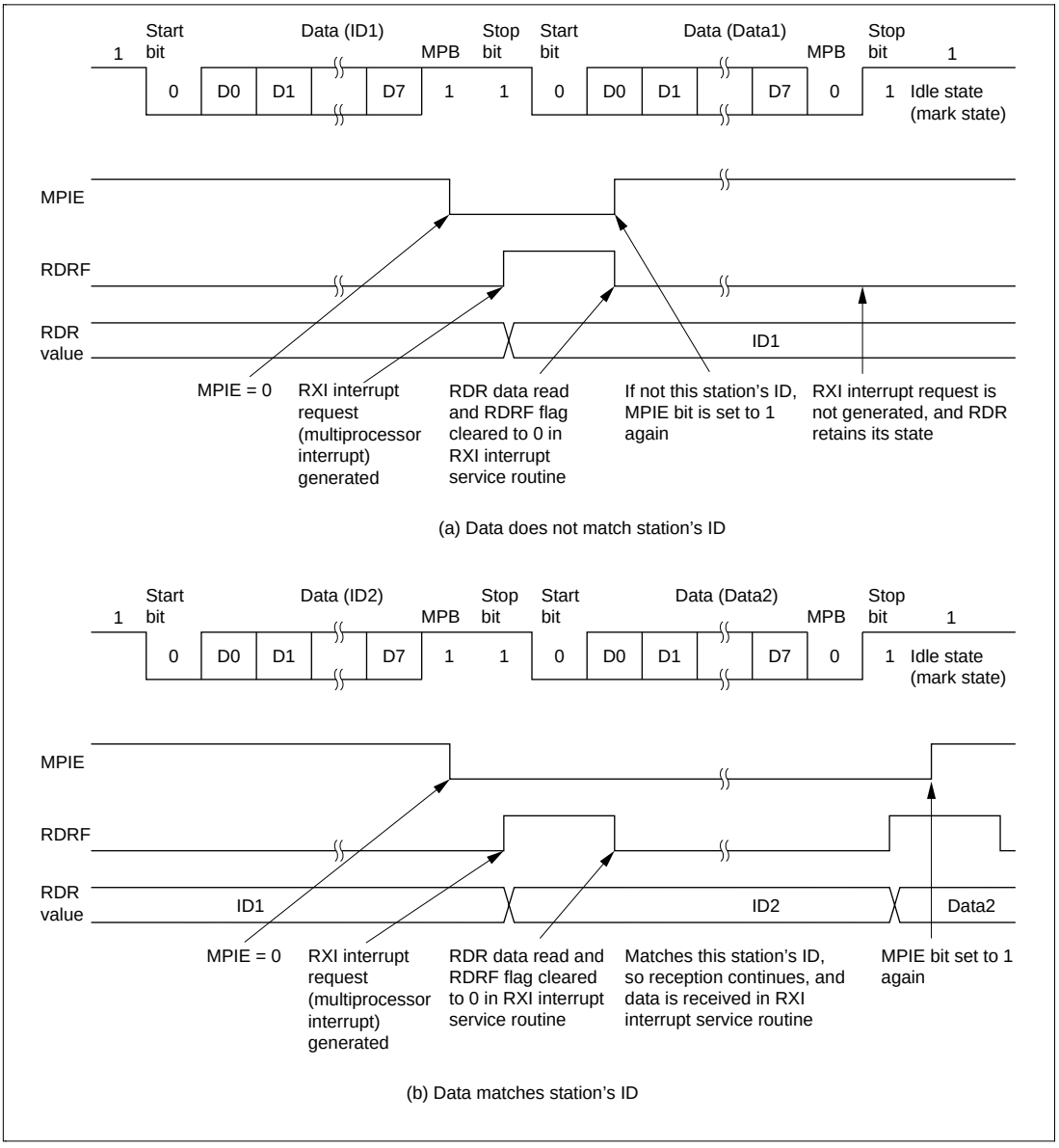


Figure 16-13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

16.3.4 Operation in Clocked Synchronous Mode

In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16-14 shows the general format for clocked synchronous serial communication.

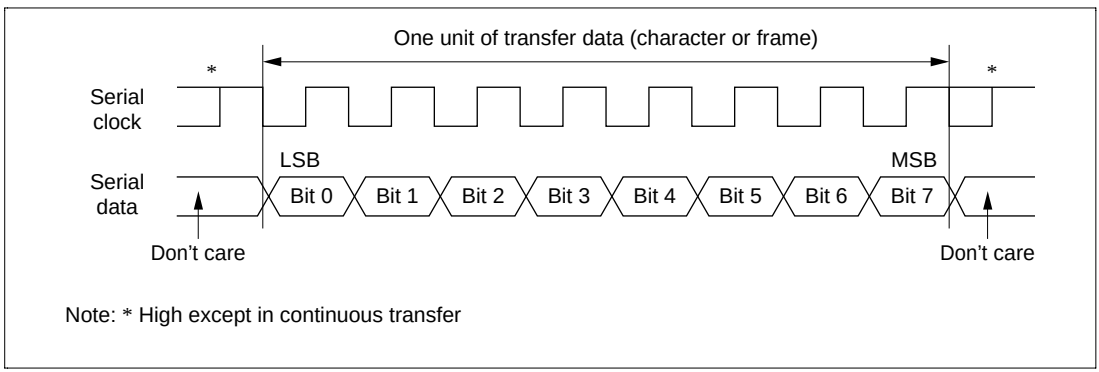


Figure 16-14 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. Data confirmation is guaranteed at the rising edge of the serial clock.

In clocked serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the transmission line holds the MSB state.

In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format: A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Clock: Either an internal clock generated by the on-chip baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 16-9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you want to perform receive operations in units of one character, you should select an external clock as the clock source.

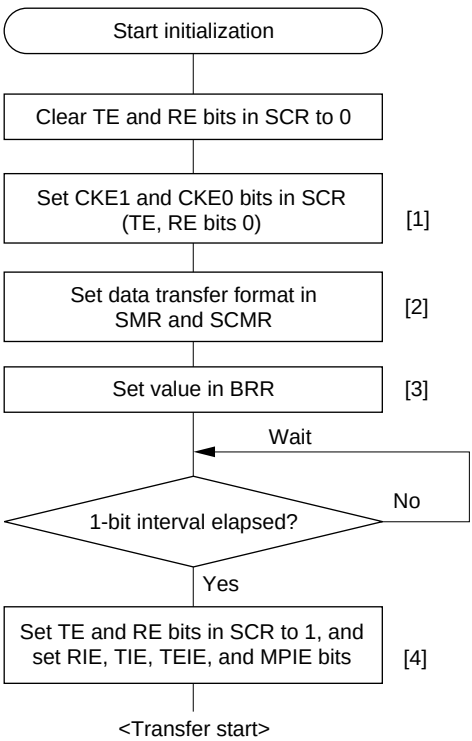
Data Transfer Operations:

- SCI initialization (clocked synchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

  Figure 16-15 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

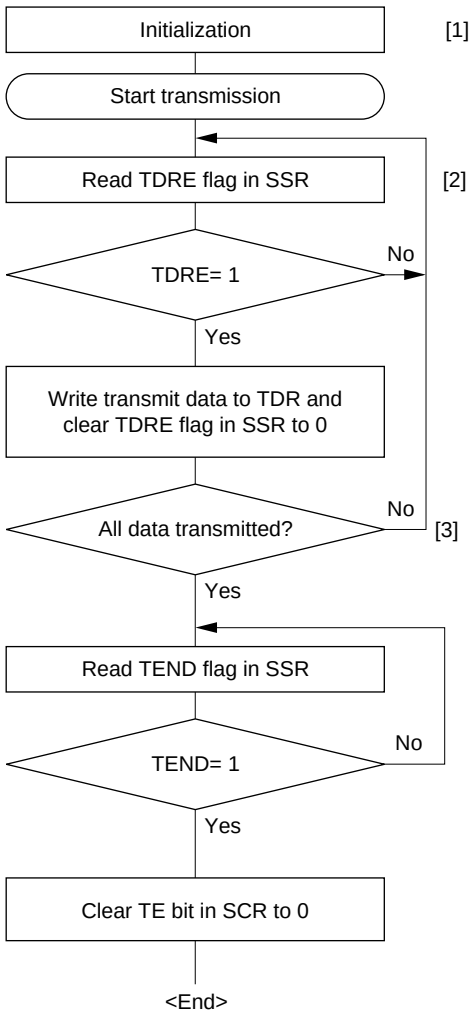
Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 16-15 Sample SCI Initialization Flowchart

- Serial data transmission (clocked synchronous mode)

  Figure 16-16 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 16-16 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an external clock has been specified, data is output synchronized with the input clock.

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and ending with the MSB (bit 7).

[3] The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is sent, and the TxD pin maintains its state.

If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

[4] After completion of serial transmission, the SCK pin is fixed high.

Figure 16-17 shows an example of SCI operation in transmission.

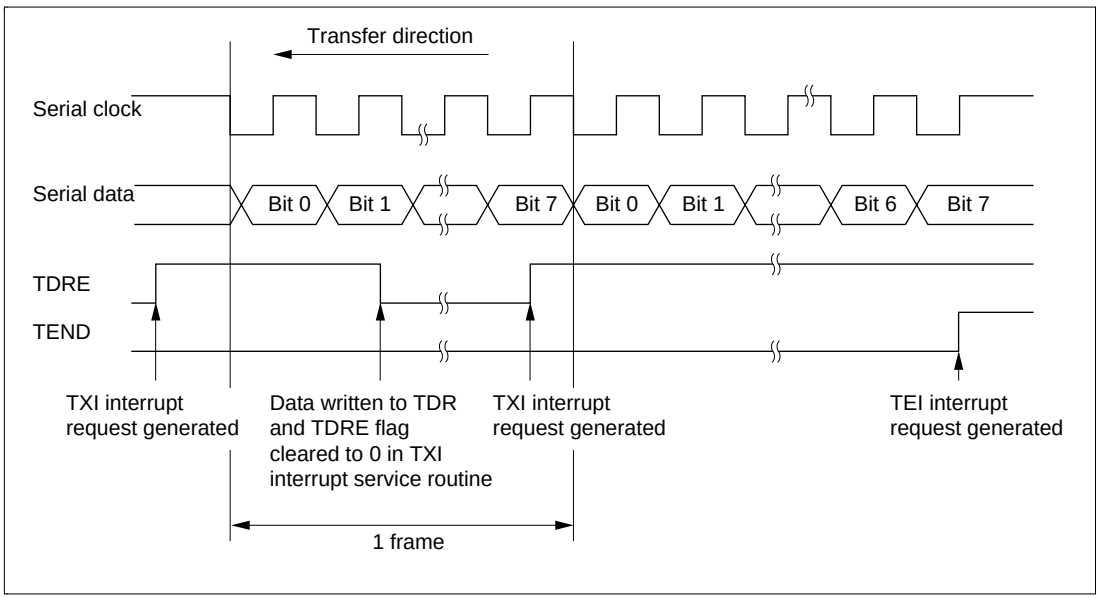


Figure 16-17 Example of SCI Operation in Transmission

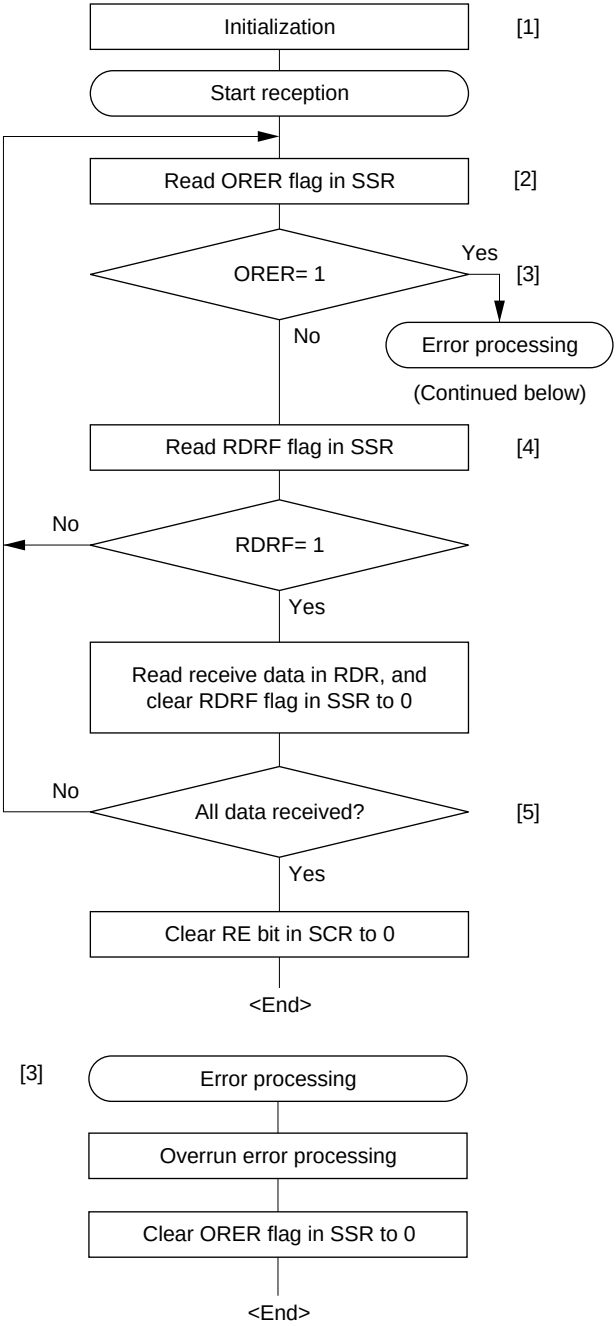
- Serial data reception (clocked synchronous mode)

  Figure 16-18 shows a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.

  When changing the operating mode from asynchronous to clocked synchronous, be sure to check that the ORER, PER, and FER flags are all cleared to 0.

  The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit nor receive operations will be possible.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 16-18 Sample Serial Reception Flowchart

In serial reception, the SCI operates as described below.

[1] The SCI performs internal initialization in synchronization with serial clock input or output.

[2] The received data is stored in RSR in LSB-to-MSB order.

After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 16-11.

Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive error interrupt (ERI) request is generated.

Figure 16-19 shows an example of SCI operation in reception.

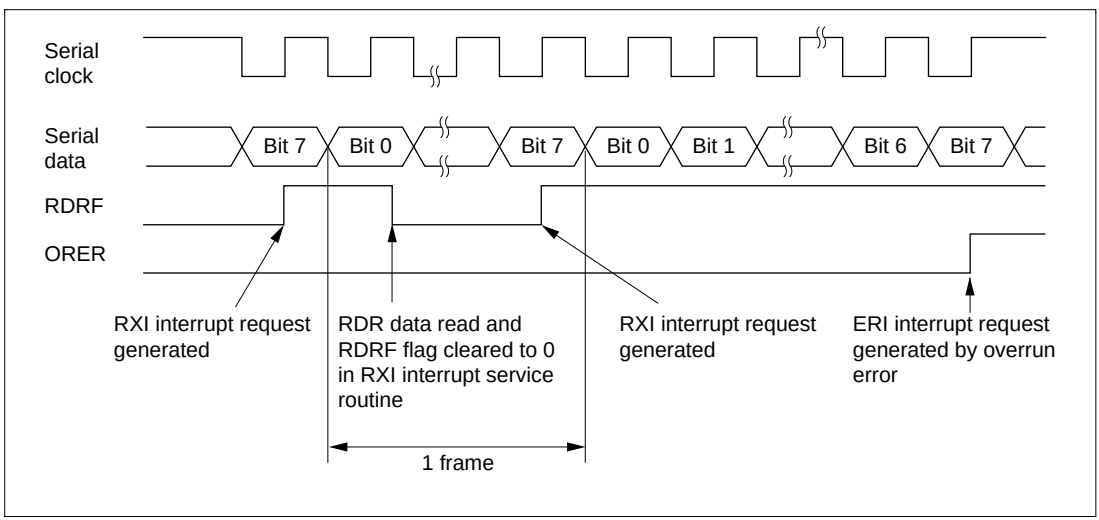
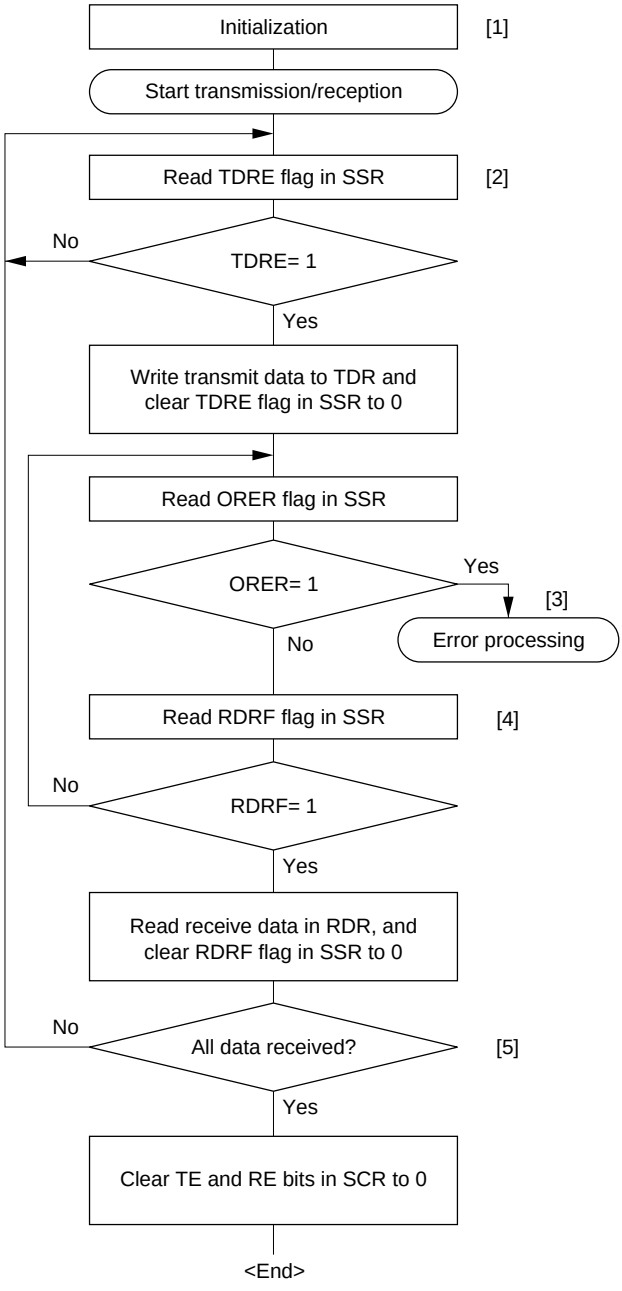


Figure 16-19 Example of SCI Operation in Reception

- Simultaneous serial data transmission and reception (clocked synchronous mode)

  Figure 16-20 shows a sample flowchart for simultaneous serial transmit and receive operations.

  The following procedure should be used for simultaneous serial data transmit and receive operations.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 16-20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

16.3.5 IrDA Operation

Figure 16-21 is a block diagram of the IrDA.

When the IrE bit of IrCR is set to enable the IrDA function, the TxD0/RxD0 signals of SCI channel 0 are encoded and decoded with waveforms conforming to the IrDA standard version 1.0 (IrTxD/IrRxD pins). Connecting these to an infrared transmitter/receiver allows the realization of infrared transmission and reception conforming to an IrDA standard version 1.0 system.

In an IrDA standard version 1.0 system, communication is initiated at a transfer rate of 9600bps. The rate is subsequently varied as required. The IrDA interface of this LSI does not have an internal function for automatically varying the transfer rate. The transfer rate must be varied using software.

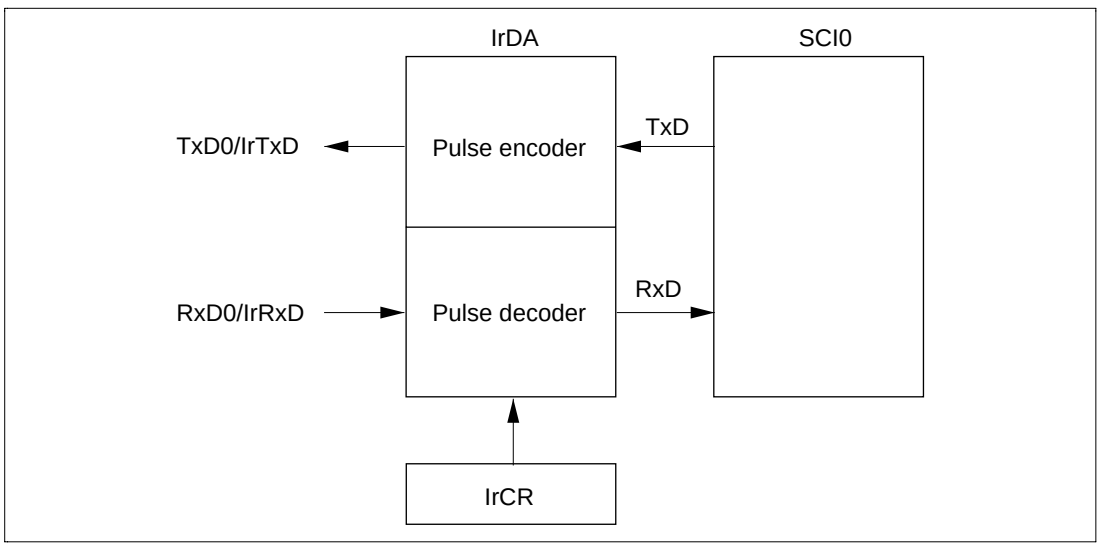


Figure 16-21 IrDA Block Diagram

(1) Transmission

When transmitting, the signal (UART frame) output from the SCI is converted by the IrDA interface into an IR frame (see Figure 16-22).

When the value of the serial data is “0”, a High pulse that has 3/16ths the width of the bit rate (the duration of 1 bit width) is output (default). Note that the High pulse can also be changed by altering the settings of IrCR IrCKS2 to IrCKS0.

As per the standard, the High pulse width is a minimum of 1.41 µs, the maximum is (3/16 + 2.5%) × bit rate, or (3/16 × bit rate) + 1.08 µs. With a 20MHz system clock ø, the minimum High pulse width can be set to 1.6 µs, which is greater than the 1.41 µs required by the standard.

When the value of the serial data is “1”, no pulse is output.

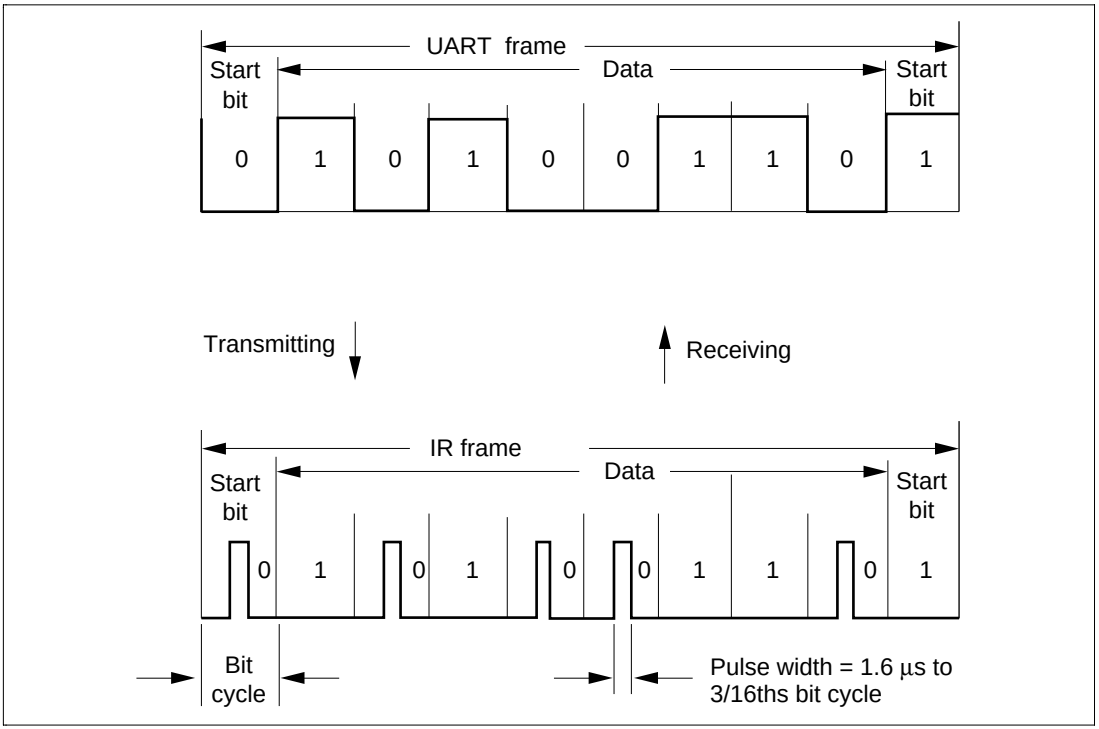


Figure 16-22 IrDA Transmit and Receive Operations

(2) Receiving

When receiving, the IR frame data is converted into UART frames by the IrDA interface and input to the SCI.

When a High pulse is detected, “0” is output. If there is no pulse for the duration of 1 bit, “1” is output. Pulses of less than the minimum pulse width of 1.41 µs are also recognized as “0” data.

(3) Selecting High Pulse Width

Table 16-12 shows the settings of IrCKS2 to IrCKS0 (for the minimum pulse width), at various LSI operating frequencies, and various bit rates to set the pulse width when transmitting with a pulse width less than 3/16ths of the bit rate.

Table 16-12 Setting Bits IrCKS2 to IrCKS0

	Bit Rate (bps) (Upper Row) / Bit Cycle × 3/16 (µs) (Lower Row)					
	2400	**9600**	**19200**	**38400**	**57600**	**115200**
Operating Frequency (MHz)	**78.13**	**19.53**	**9.77**	**4.88**	**3.26**	**1.63**
2	010	010	010	010	010	—
2.097152	010	010	010	010	010	—
2.4576	010	010	010	010	010	—
3	011	011	011	011	011	—
3.6864	011	011	011	011	011	011
4.9152	011	011	011	011	011	011
5	011	011	011	011	011	011
6	100	100	100	100	100	100
6.144	100	100	100	100	100	100
7.3728	100	100	100	100	100	100
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101
25	110	110	110	110	110	110

Legend

—: SCI cannot be set to this bit rate.

16.4 SCI Interrupts

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 16-13 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in the SCR. Each kind of interrupt request is sent to the interrupt controller independently.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DMAC or DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC. The DMAC or DTC cannot be activated by a TEI interrupt request.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt can activate the DMAC or DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. The DMAC or DTC cannot be activated by an ERI interrupt request.

Note that the DMAC cannot be activated by interrupts of SCI channels 2 to 4.

Table 16-13 SCI Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation	DMAC Activation	Priority*
0	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	High
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
3	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
4	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	Low

Note: * This table shows the initial state immediately after a reset. Relative priorities among channels can be changed by means of the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt may have priority for acceptance, with the result that the TDRE and TEND flags are cleared. Note that the TEI interrupt will not be accepted in this case.

16.5 Usage Notes

The following points should be noted when using the SCI.

Relation between Writes to TDR and the TDRE Flag

The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously

If a number of receive errors occur at the same time, the state of the status flags in SSR is as shown in table 16-14. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 16-14 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR to RDR	Receive Error Status
1	1	0	0	X	Overrun error
0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Notes: ○: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

Break Detection and Processing (Asynchronous Mode Only): When framing error (FER) detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the parity error flag (PER) may also be set.

Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

Sending a Break (Asynchronous Mode Only): The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Consequently, DDR and DR for the port corresponding to the TxD pin are first set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only): Transmission cannot be started when a receive error flag (ORER, PER, or FER) 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 also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode: In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock. This is illustrated in figure 16-23.

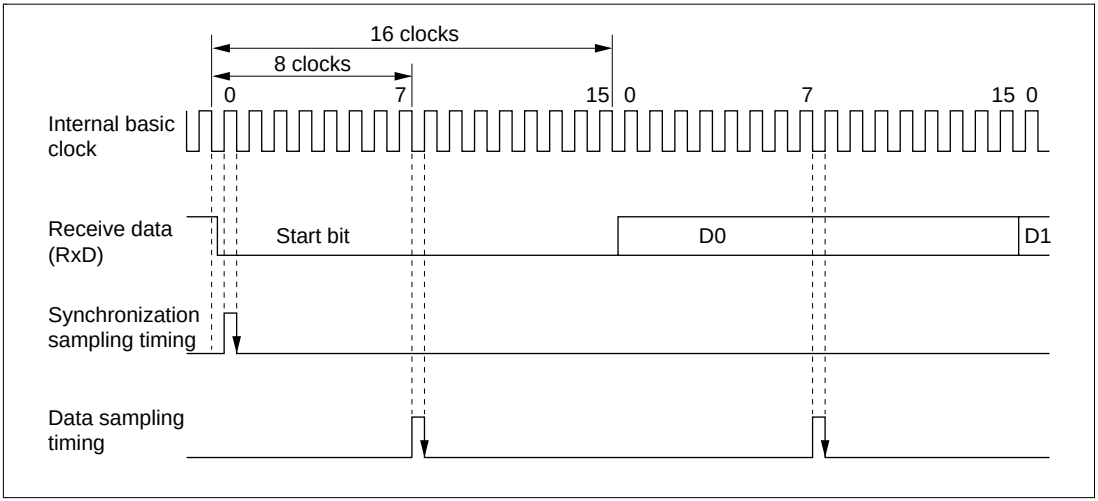


Figure 16-23 Receive Data Sampling Timing in Asynchronous Mode

Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

... Formula (1)

Where M : Reception margin (%)
N : Ratio of bit rate to clock (N = 16)
D : Clock duty (D = 0 to 1.0)
L : Frame length (L = 9 to 12)
F : Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin of 46.875% is given by formula (2) below.

When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$

$$= 46.875\%$$ ... Formula (2)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Restrictions on Use of DMAC or DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ø clock cycles after TDR is updated by the DMAC or DTC. Misoperation may occur if the transmit clock is input within 4 ø clocks after TDR is updated. (Figure 16-24)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI reception end interrupt (RXI).

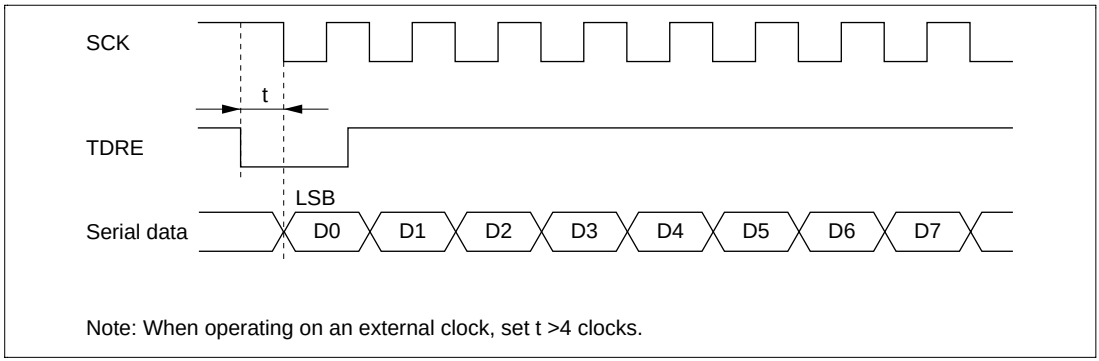


Figure 16-24 Example of Clocked Synchronous Transmission by DTC

Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode depend on the port settings, and becomes high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined. When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read -> TDR write -> TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization. Figure 16-25 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 16-26 and 16-27.

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. RSR, RDR, and SSR are reset. If a transition is made without stopping operation, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

  Figure 16-28 shows a sample flowchart for mode transition during reception.

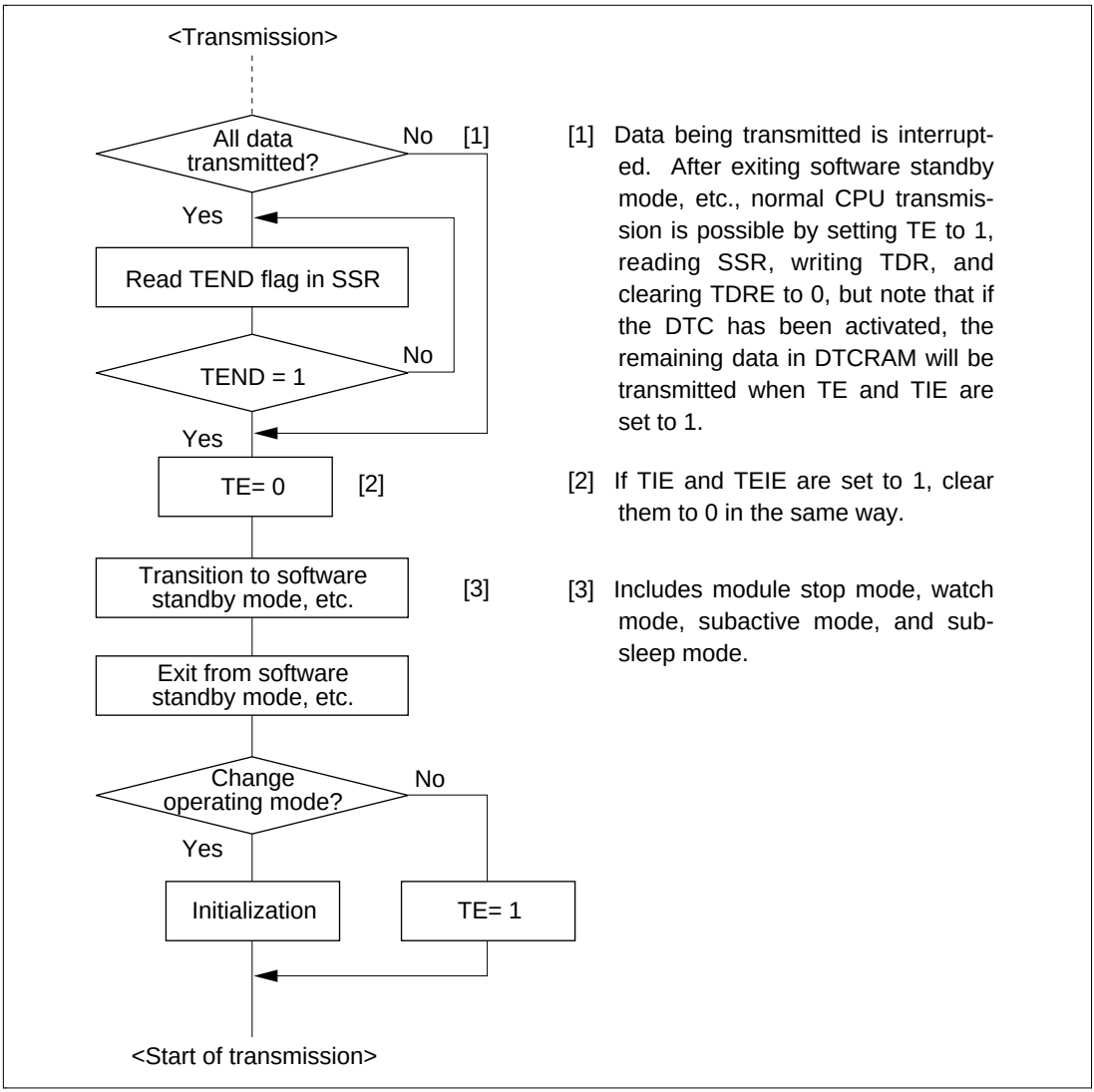


Figure 16-25 Sample Flowchart for Mode Transition during Transmission

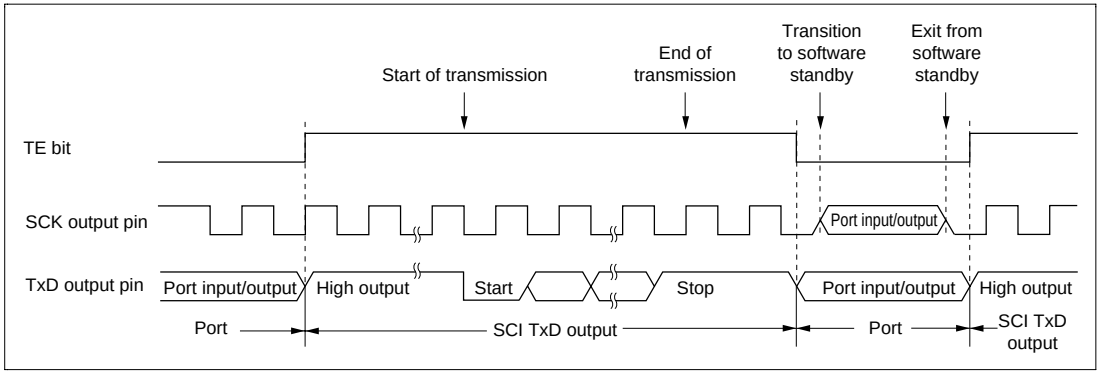


Figure 16-26 Asynchronous Transmission Using Internal Clock

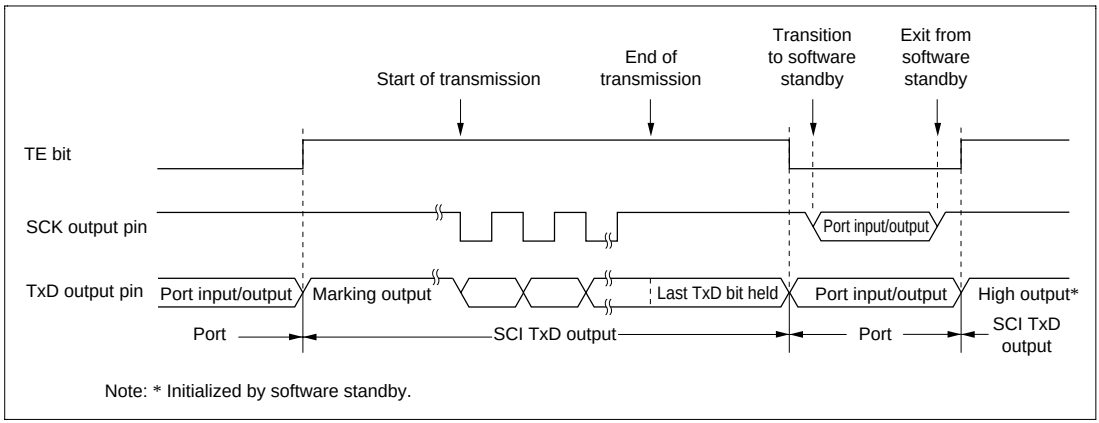


Figure 16-27 Synchronous Transmission Using Internal Clock

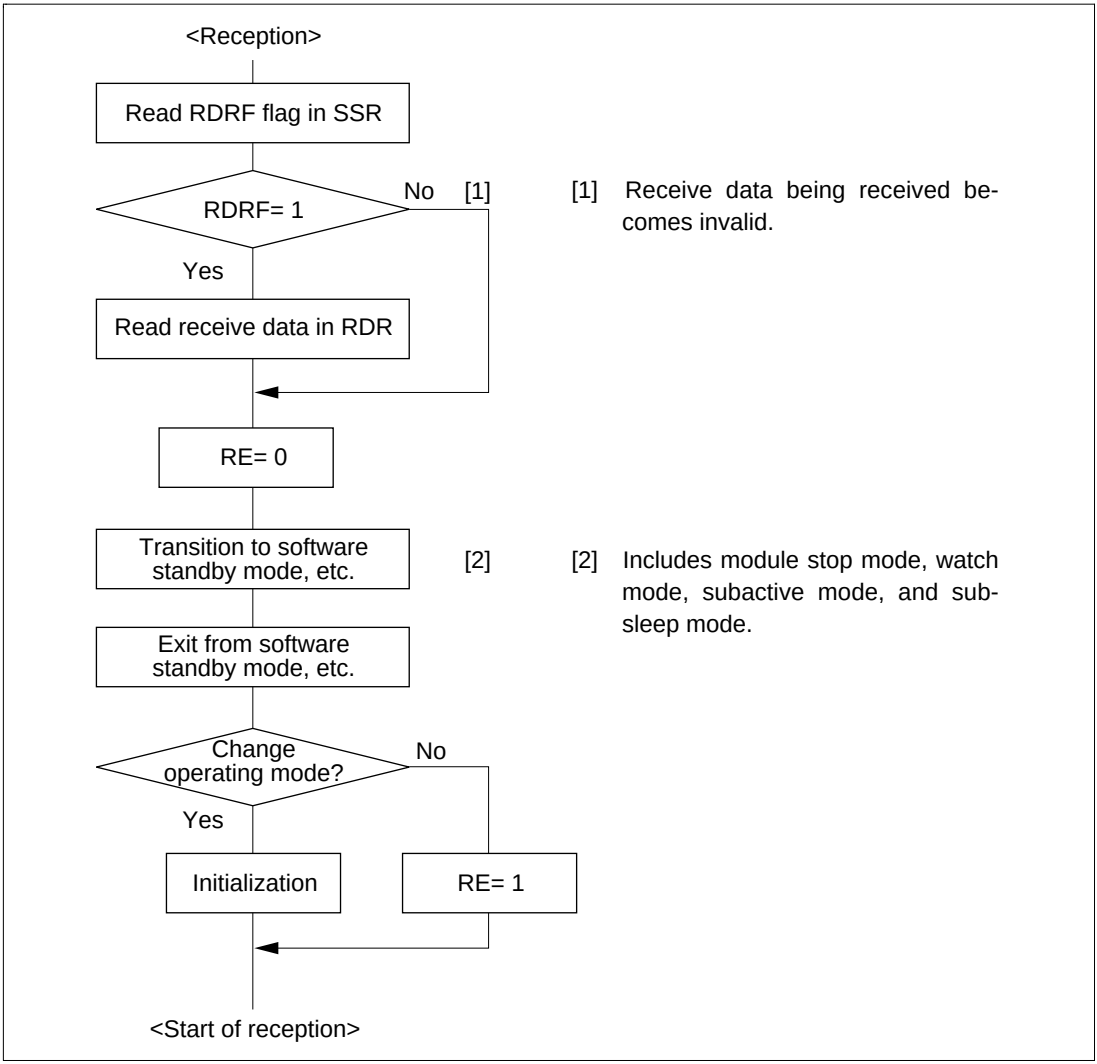


Figure 16-28 Sample Flowchart for Mode Transition during Reception

Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 16-29)

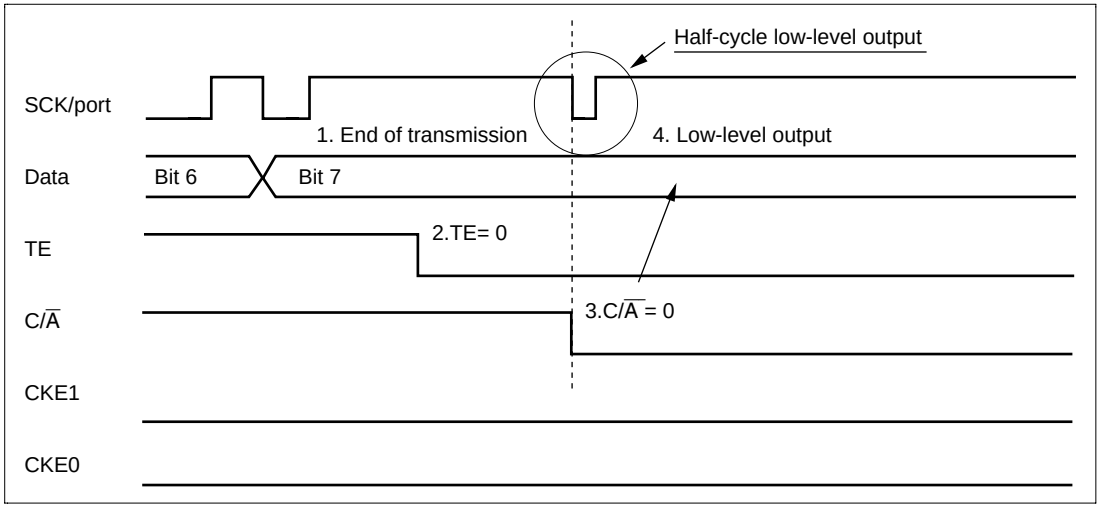


Figure 16-29 Operation when Switching from SCK Pin Function to Port Pin Function

- Sample Procedure for Avoiding Low-Level Output: As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

  With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

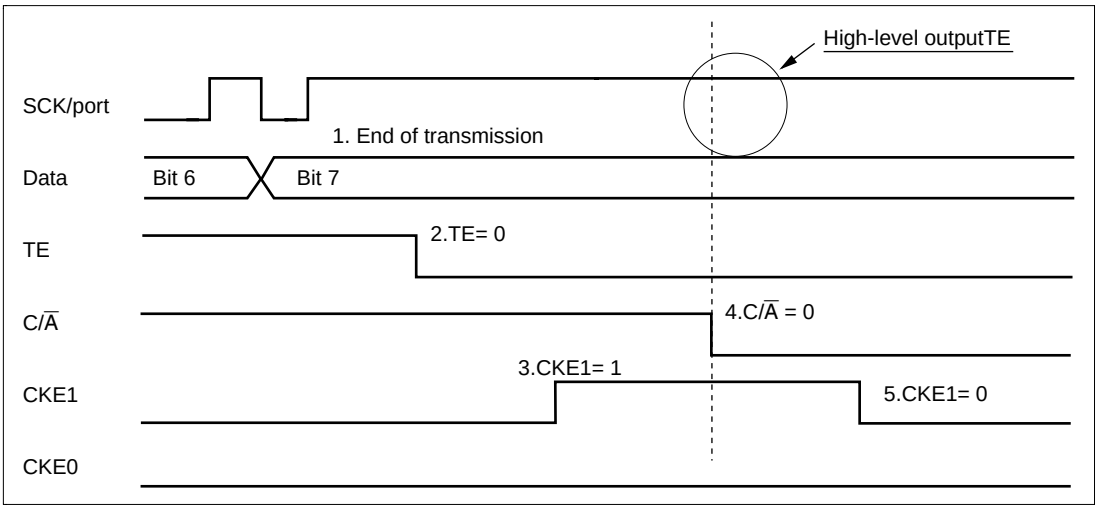


Figure 16-30 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 17 Smart Card Interface

17.1 Overview

SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

17.1.1 Features

Features of the Smart Card interface supported by the H8S/2633 Series are as follows.

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/receive error) that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate the DMA controller (DMAC) or data transfer controller (DTC) to execute data transfer

17.1.2 Block Diagram

Figure 17-1 shows a block diagram of the Smart Card interface.

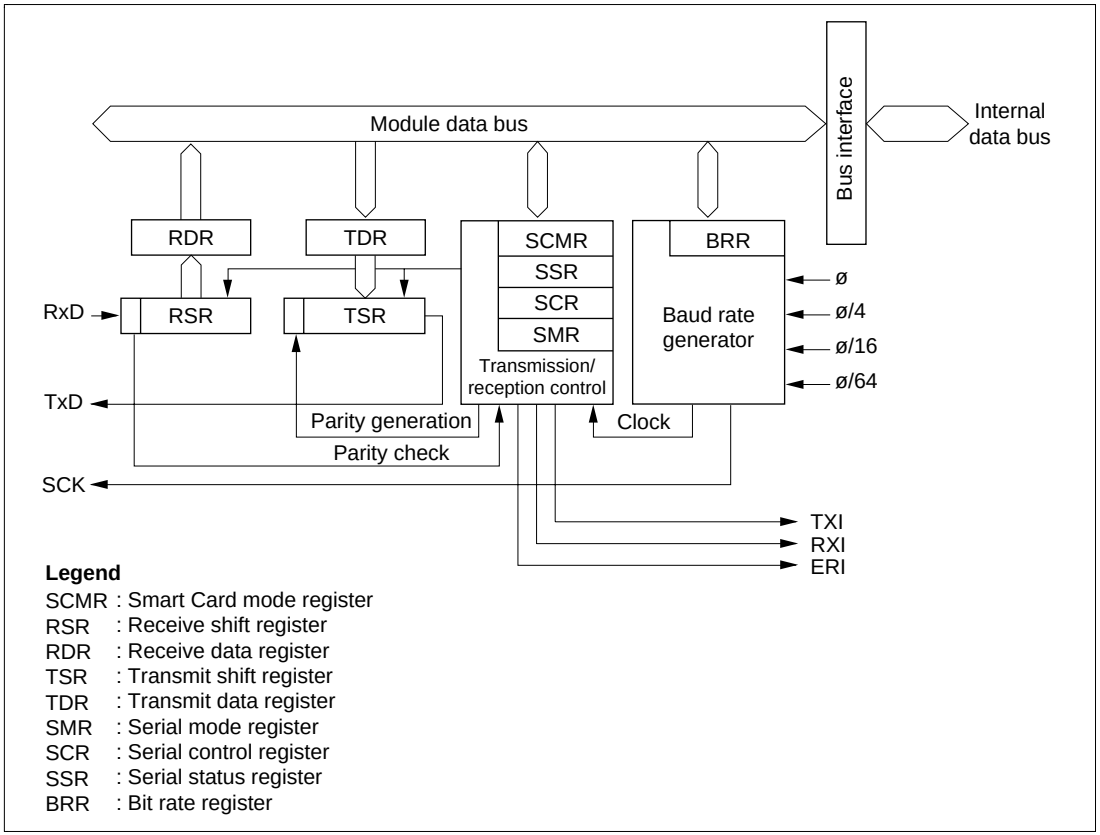


Figure 17-1 Block Diagram of Smart Card Interface

17.1.3 Pin Configuration

Table 17-1 shows the Smart Card interface pin configuration.

Table 17-1 Smart Card Interface Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data output
3	Serial clock pin 3	SCK3	I/O	SCI3 clock input/output
	Receive data pin 3	RxD3	Input	SCI3 receive data input
	Transmit data pin 3	TxD3	Output	SCI3 transmit data output
4	Serial clock pin 4	SCK4	I/O	SCI4 clock input/output
	Receive data pin 4	RxD4	Input	SCI4 receive data input
	Transmit data pin 4	TxD4	Output	SCI4 transmit data output

17.1.4 Register Configuration

Table 17-2 shows the registers used by the Smart Card interface. Details of BRR, TDR, RDR, and MSTPCR are the same as for the normal SCI function: see the register descriptions in section 16, Serial Communication Interface.

Table 17-2 Smart Card Interface Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*[2]	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
3	Serial mode register 3	SMR3	R/W	H'00	H'FDD0
	Bit rate register 3	BRR3	R/W	H'FF	H'FDD1
	Serial control register 3	SCR3	R/W	H'00	H'FDD2
	Transmit data register 3	TDR3	R/W	H'FF	H'FDD3
	Serial status register 3	SSR3	R/(W)*[2]	H'84	H'FDD4
	Receive data register 3	RDR3	R	H'00	H'FDD5
	Smart card mode register 3	SCMR3	R/W	H'F2	H'FDD6
4	Serial mode register 4	SMR4	R/W	H'00	H'FDD8
	Bit rate register 4	BRR4	R/W	H'FF	H'FDD9
	Serial control register 4	SCR4	R/W	H'00	H'FDDA
	Transmit data register 4	TDR4	R/W	H'FF	H'FDDB
	Serial status register 4	SSR4	R/(W)*[2]	H'84	H'FDDC
	Receive data register 4	RDR4	R	H'00	H'FDDD
	Smart card mode register 4	SCMR4	R/W	H'F2	H'FDDE
All	Module stop control register B, C	MSTPCRB	R/W	H'FF	H'FDE9
		MSTPCRC	R/W	H'FF	H'FDEA

Notes: *1 Lower 16 bits of the address.

*2 Can only be written with 0 for flag clearing.

17.2 Register Descriptions

Registers added with the Smart Card interface and bits for which the function changes are described here.

17.2.1 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR is an 8-bit readable/writable register that selects the Smart Card interface function.

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. This function is used together with the SDIR bit for communication with an inverse convention card. The SINV bit does not affect the logic level of the parity bit. For parity-related setting procedures, see section 17.3.4, Register Settings.

Bit 2

SINV	Description	
0	TDR contents are transmitted as they are	(Initial value)
	Receive data is stored as it is in RDR	
1	TDR contents are inverted before being transmitted	
	Receive data is stored in inverted form in RDR	

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the Smart Card interface function.

Bit 0

SMIF	Description	
0	Smart Card interface function is disabled	(Initial value)
1	Smart Card interface function is enabled	

17.2.2 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written, to clear these flags.

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this, the setting conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 16.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In Smart Card interface mode, bit 4 indicates the status of the error signal sent back from the receiving end in transmission. Framing errors are not detected in Smart Card interface mode.

Bit 4

ERS	Description
0	Normal reception, with no error signal [Clearing conditions] (Initial value) • Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	Error signal sent from receiver indicating detection of parity error [Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 16.2.7, Serial Status Register (SSR).

However, the setting conditions for the TEND bit, are as shown below.

Bit 2

TEND	Description
0	Transmission is in progress [Clearing conditions] (Initial value) • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	Transmission has ended [Setting conditions] • Upon reset, and in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 1 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 1

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

17.2.3 Serial Mode Register (SMR)

Bit	:	7	6	5	4	3	2	1	0
		GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When the smart card interface is used, be sure to make the 1 setting shown for bit 5.

The function of bits 7, 6, 3, and 2 of SMR changes in Smart Card interface mode.

Bit 7—GSM Mode (GM): Sets the smart card interface function to GSM mode.

This bit is cleared to 0 when the normal smart card interface is used. In GSM mode, this bit is set to 1, the timing of setting of the TEND flag that indicates transmission completion is advanced and clock output control mode addition is performed. The contents of the clock output control mode addition are specified by bits 1 and 0 of the serial control register (SCR).

Bit 7

GM	Description
0	Normal smart card interface mode operation (Initial value) • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output ON/OFF control (set by SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bit 6—Block Transfer Mode (BLK): Selects block transfer mode.

Bit 6	
BLK	**Description**
0	Normal Smart Card interface mode operation • Error signal transmission/detection and automatic data retransmission performed • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not performed • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

Bits 3 and 2—Basic Clock Pulse 1 and 2 (BCP1, BCP0): These bits specify the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface.

Bit 3	Bit 2		
BCP1	**BCP0**	**Description**	
0	1	32 clock periods	(Initial value)
	0	64 clock periods	
1	1	372 clock periods	
	0	256 clock periods	

Bits 5, 4, 1, and 0: Operate in the same way as for the normal SCI. For details, see section 16.2.5, serial mode register (SMR).

17.2.4 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In smart card interface mode, the function of bits 1 and 0 of SCR changes when bit 7 of the serial mode register (SMR) is set to 1.

Bits 7 to 2—Operate in the same way as for the normal SCI.

For details, see section 16.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin.

In smart card interface mode, in addition to the normal switching between clock output enabling and disabling, the clock output can be specified as to be fixed high or low.

SCMR	SMR	SCR Setting		
SMIF	**C/$\overline{A}$, GM**	**CKE1**	**CKE0**	**SCK Pin Function**
0	See the SCI			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Outputs clock as SCK output pin
1	1	0	0	Operates as SCK output pin, with output fixed low
1	1	0	1	Outputs clock as SCK output pin
1	1	1	0	Operates as SCK output pin, with output fixed high
1	1	1	1	Outputs clock as SCK output pin

17.3 Operation

17.3.1 Overview

The main functions of the Smart Card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer. (except in block transfer mode)
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

17.3.2 Pin Connections

Figure 17-2 shows a schematic diagram of Smart Card interface related pin connections.

In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor.

When the clock generated on the Smart Card interface is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. No connection is needed if the IC card uses an internal clock.

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

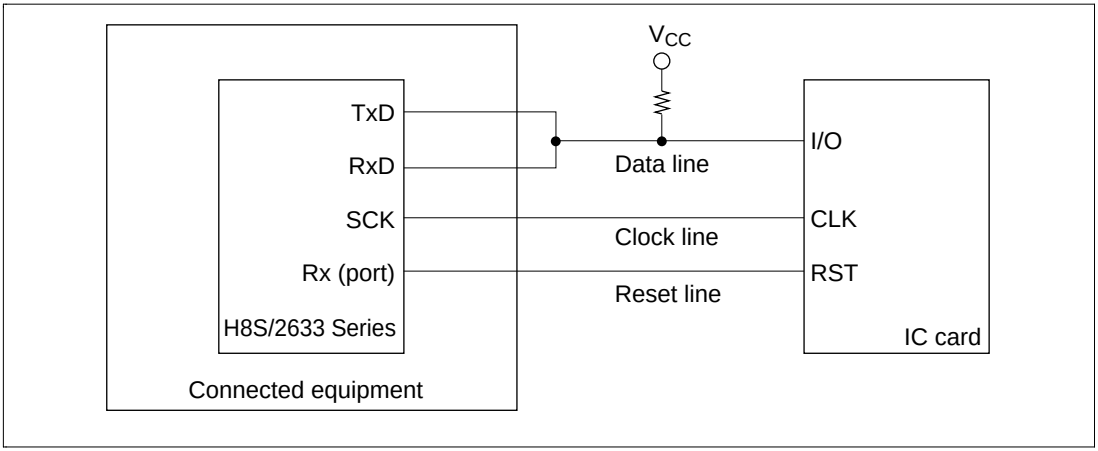


Figure 17-2 Schematic Diagram of Smart Card Interface Pin Connections

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

17.3.3 Data Format

(1) Normal Transfer Mode

Figure 17-3 shows the normal Smart Card interface data format. In reception in this mode, a parity check is carried out on each frame, and if an error is detected an error signal is sent back to the transmitting end, and retransmission of the data is requested. If an error signal is sampled during transmission, the same data is retransmitted.

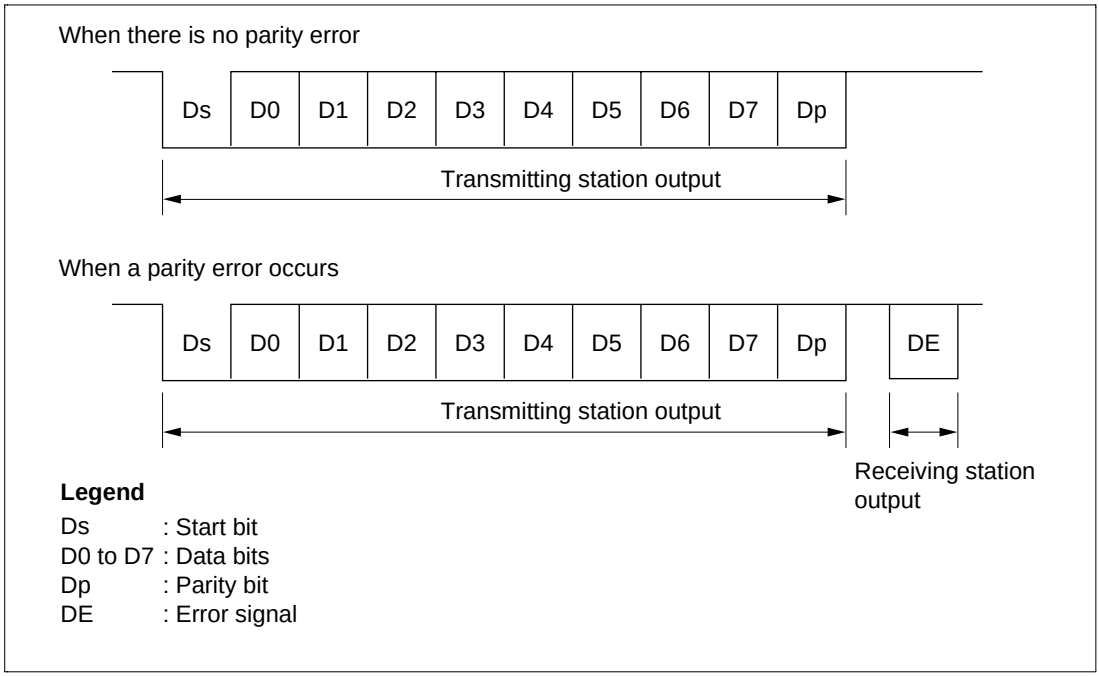


Figure 17-3 Normal Smart Card Interface Data Format

The operation sequence is as follows.

[1] When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check.

If there is no parity error and the data is received normally, the receiving station waits for reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving station places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit the next data frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the erroneous data.

(2) Block Transfer Mode

The operation sequence in block transfer mode is as follows.

[1] When the data line in not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.

[4] After reception, a parity error check is carried out, but an error signal is not output even if an error has occurred. When an error occurs reception cannot be continued, so the error flag should be cleared to 0 before the parity bit of the next frame is received.

[5] The transmitting station proceeds to transmit the next data frame.

17.3.4 Register Settings

Table 17-3 shows a bit map of the registers used by the smart card interface.

Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described below.

Table 17-3 Smart Card Interface Register Settings

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SMR	GM	BLK	1	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	TIE	RIE	TE	RE	0	0	CKE1*	CKE0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	—	—	—	—	SDIR	SINV	—	SMIF

Notes: — : Unused bit.
*: The CKE1 bit must be cleared to 0 when the GM bit in SMR is cleared to 0.

SMR Setting: The GM bit is cleared to 0 in normal smart card interface mode, and set to 1 in GSM mode. The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. Bits BCP1 and BCP0 select the number of basic clock periods in a 1-bit transfer interval. For details, see section 17.3.5, Clock.

The BLK bit is cleared to 0 in normal smart card interface mode, and set to 1 in block transfer mode.

BRR Setting: BRR is used to set the bit rate. See section 17.3.5, Clock, for the method of calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the normal SCI. For details, see section 16, Serial Communication Interface.

Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared to 0, set these bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When the GM bit in SMR is set to 1, clock output is performed. The clock output can also be fixed high or low.

Smart Card Mode Register (SCMR) Setting:

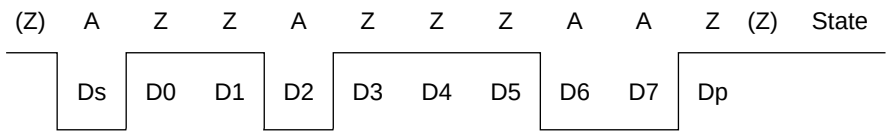
The SDIR bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SINV bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SMIF bit is set to 1 in the case of the Smart Card interface.

Examples of register settings and the waveform of the start character are shown below for the two types of IC card (direct convention and inverse convention).

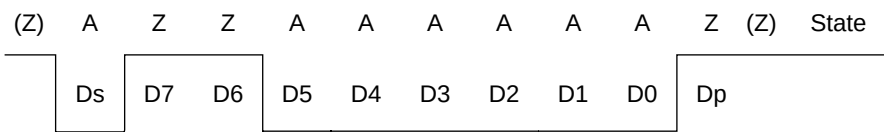
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B.

The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F.

The parity bit is 0, corresponding to state Z, since even parity is stipulated for the Smart Card.

With the H8S/2633 Series, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (the same applies to both transmission and reception).

17.3.5 Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with BRR and the CKS1, CKS0, BCP1 and BCP0 bits in SMR. The formula for calculating the bit rate is as shown below. Table 17-5 shows some sample bit rates.

If clock output is selected by setting CKE0 to 1, a clock is output from the SCK pin. The clock frequency is determined by the bit rate and the setting of bits BCP1 and BCP0.

$$B = \frac{ø}{S \times 2^{2n+1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ø = Operating frequency (MHz)
n = See table 17-4
S = Number of internal clocks in 1-bit period, set by BCP1 and BCP0

Table 17-4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 17-5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0 and S = 372)

	ø (MHz)							
N	10.00	10.714	13.00	14.285	16.00	18.00	20.00	25.00
0	13441	14400	17473	19200	21505	24194	26882	33602
1	6720	7200	8737	9600	10753	12097	13441	16801
2	4480	4800	5824	6400	7168	8065	8961	11201

Note: Bit rates are rounded to the nearest whole number.

The method of calculating the value to be set in the bit rate register (BRR) from the operating frequency and bit rate, on the other hand, is shown below. N is an integer, $0 \leq N \leq 255$, and the smaller error is specified.

$$N = \frac{ø}{S \times 2^{2n+1} \times B} \times 10^6 - 1$$

Table 17-6 Examples of BRR Settings for Bit Rate B (bit/s) (When n = 0 and S = 372)

	ø (MHz)															
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00		20.00		25.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99	2	6.60	3	12.49

Table 17-7 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ø (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left(\frac{ø}{S \times 2^{2n+1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

17.3.6 Data Transfer Operations

Initialization: Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

[1] Clear the TE and RE bits in SCR to 0.

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, CKS0 bits in SMR. Set the PE bit to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

Serial Data Transmission: As data transmission in smart card mode involves error signal sampling and retransmission processing, the processing procedure is different from that for the normal SCI. Figure 17-4 shows a flowchart for transmitting, and figure 17-5 shows the relation between a transmit operation and the internal registers.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ERS error flag in SSR is cleared to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set to 1.

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit operation. The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.

With the above processing, interrupt servicing or data transfer by the DMAC or DTC is possible.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit data empty interrupt (TXI) request will be generated. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transfer error interrupt (ERI) request will be generated.

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 17-6.

If the DMAC or DTC is activated by a TXI request, the number of bytes set in the DMAC or DTC can be transmitted automatically, including automatic retransmission.

For details, see Interrupt Operation and Data Transfer Operation by DTC below.

Note: For block transfer mode, see section 16.3.2, Operation in Asynchronous Mode.

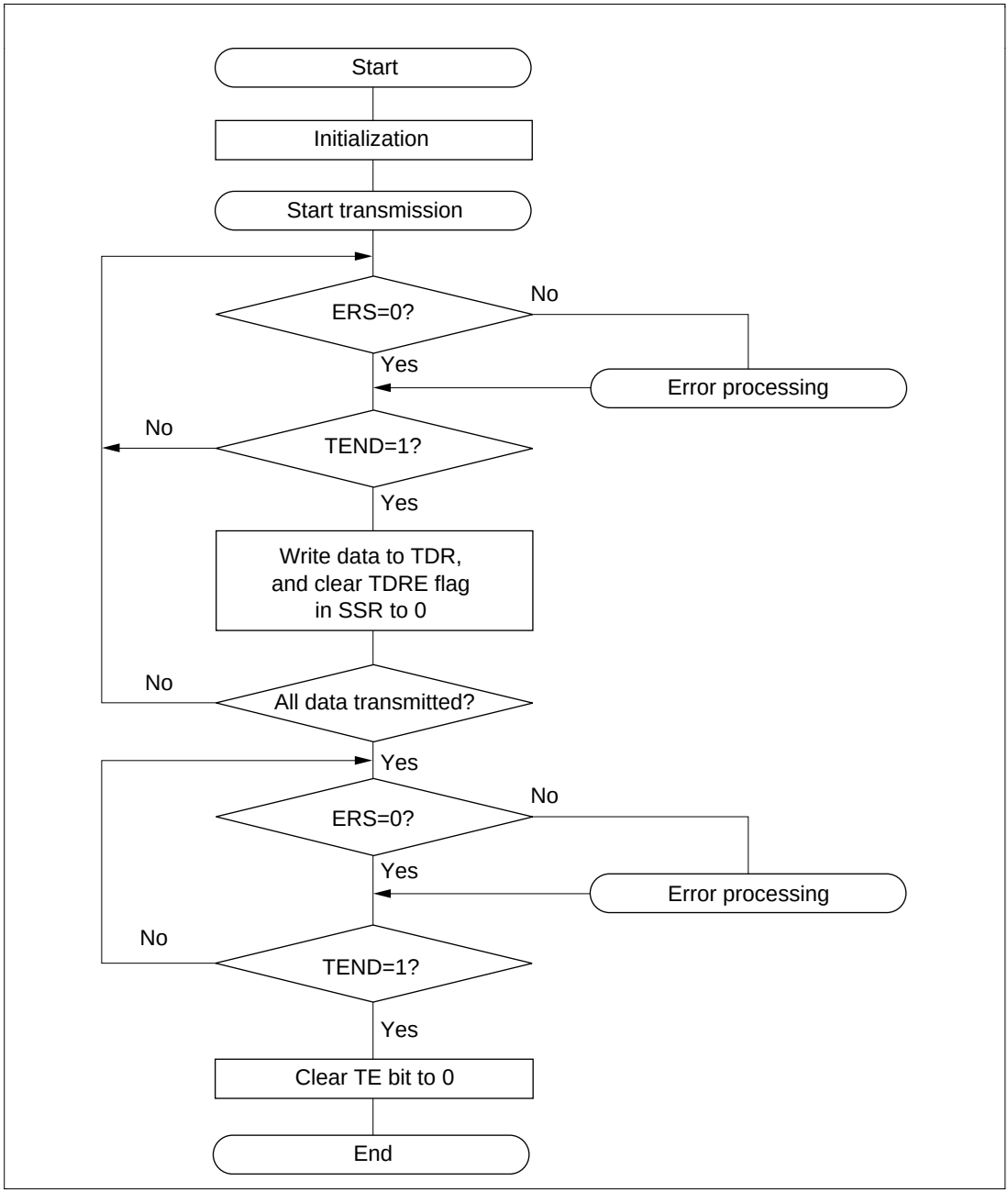
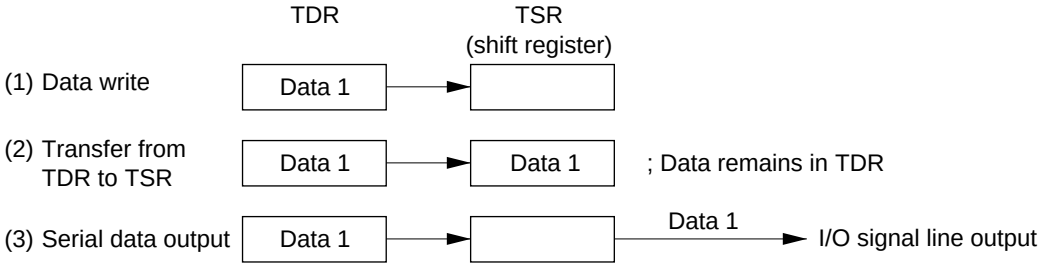


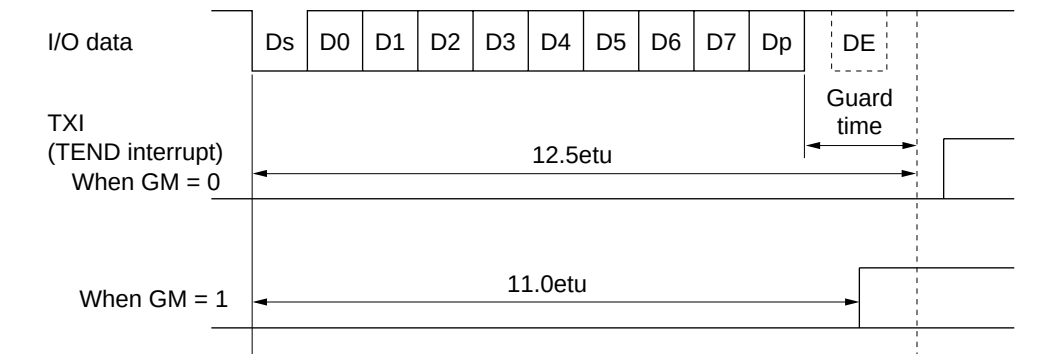
Figure 17-4 Example of Transmission Processing Flow



In case of normal transmission: TEND flag is set
In case of transmit error: ERS flag is set
Steps (2) and (3) above are repeated until the TEND flag is set

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in LSB-first transmission, D0 in MSB-first transmission) of the next transfer data to be transmitted has been completed.

Figure 17-5 Relation Between Transmit Operation and Internal Registers



Legend
Ds : Start bit
D0 to D7 : Data bits
Dp : Parity bit
DE : Error signal

Figure 17-6 TEND Flag Generation Timing in Transmission Operation

Serial Data Reception (Except Block Transfer Mode): Data reception in Smart Card mode uses the same processing procedure as for the normal SCI. Figure 17-7 shows an example of the transmission processing flow.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either is set, perform the appropriate receive error processing, then clear both the ORER and the PER flag to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2].

[6] To end reception, clear the RE bit to 0.

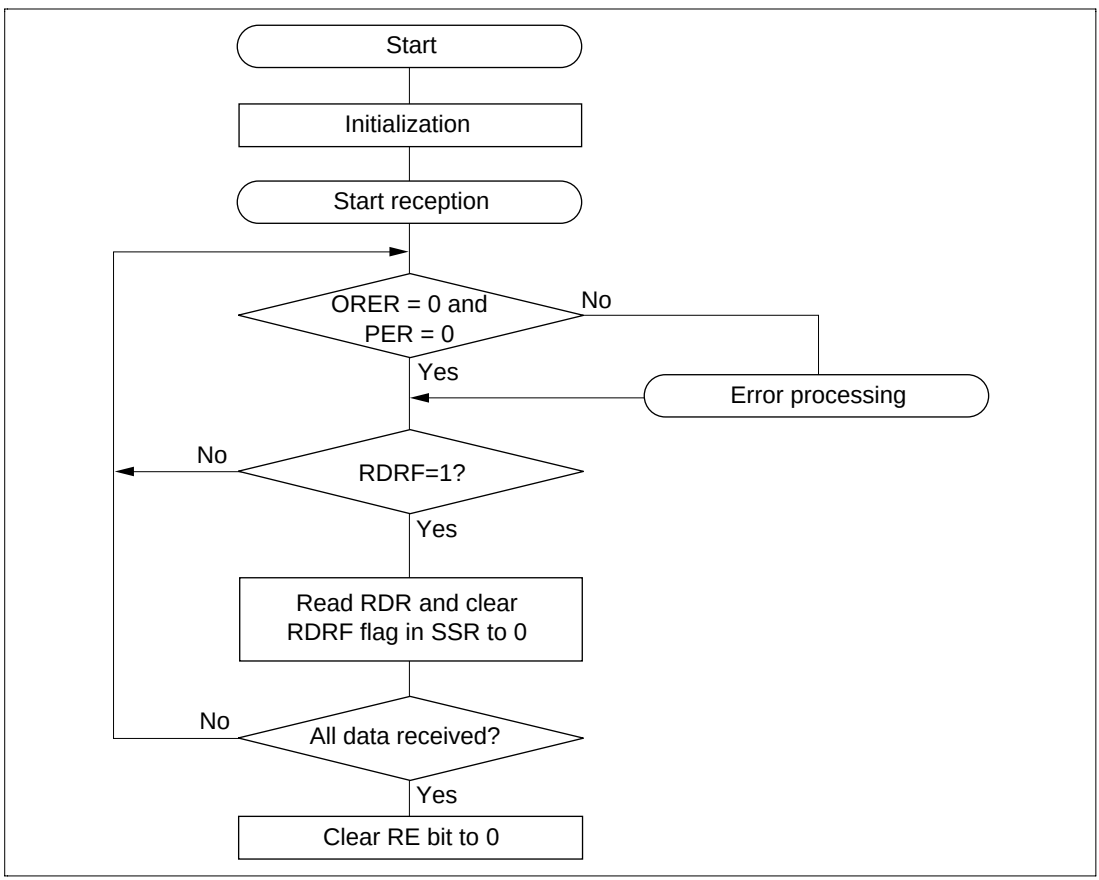


Figure 17-7 Example of Reception Processing Flow

With the above processing, interrupt servicing or data transfer by the DMAC or DTC is possible.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive data full interrupt (RXI) request will be generated. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transfer error interrupt (ERI) request will be generated.

If the DMAC or DTC is activated by an RXI request, the receive data in which the error occurred is skipped, and only the number of bytes of receive data set in the DMAC or DTC are transferred.

For details, see Interrupt Operation and Data Transfer Operation by DTC below.

If a parity error occurs during reception and the PER is set to 1, the received data is still transferred to RDR, and therefore this data can be read.

Note: For block transfer mode, see section 16.3.2, Operation in Asynchronous Mode.

Mode Switching Operation: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output Level: When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 17-8 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

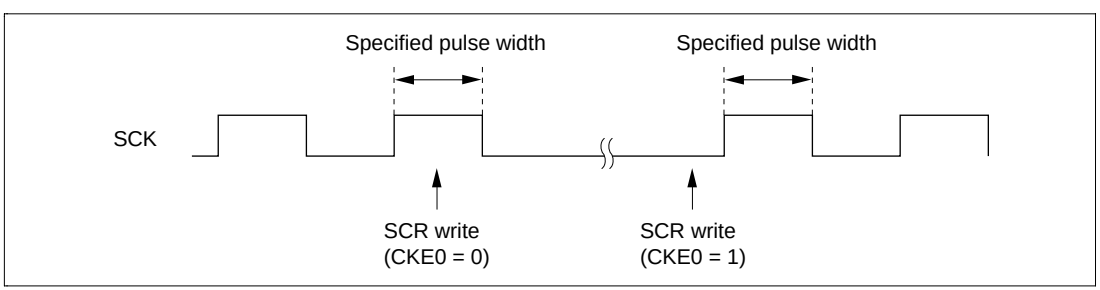


Figure 17-8 Timing for Fixing Clock Output Level

Interrupt Operation (Except Block Transfer Mode): There are three interrupt sources in smart card interface mode: transmit data empty interrupt (TXI) requests, transfer error interrupt (ERI)

requests, and receive data full interrupt (RXI) requests. The transmit end interrupt (TEI) request is not used in this mode.

When the TEND flag in SSR is set to 1, a TXI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated.

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request is generated. The relationship between the operating states and interrupt sources is shown in table 17-8.

Note: For block transfer mode, see section 16.4, SCI Interrupts.

Table 17-8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DMAC Activation	DTC Activation
Transmit Mode	Normal operation	TEND	TIE	TXI	Possible	Possible
	Error	ERS	RIE	ERI	Not possible	Not possible
Receive Mode	Normal operation	RDRF	RIE	RXI	Possible	Possible
	Error	PER, ORER	RIE	ERI	Not possible	Not possible

Data Transfer Operation by DMAC or DTC: In smart card mode, as with the normal SCI, transfer can be carried out using the DMAC or DTC. In a transmit operation, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DMAC or DTC activation source, the DMAC or DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, TEND remains cleared to 0 and the DMAC is not activated. Therefore, the SCI and DMAC will automatically transmit the specified number of bytes, including retransmission in the event of an error. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DMAC or DTC, it is essential to set and enable the DMAC or DTC before carrying out SCI setting. For details of the DMAC or DTC setting procedures, see section 8, DMA Controller (DMAC) and section 9, Data Transfer Controller (DTC).

In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DMAC or DTC activation source, the DMAC or DTC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DMAC or

DTC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DMAC or DTC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

Note: For block transfer mode, see section 16.4, SCI Interrupts.

17.3.7 Operation in GSM Mode

Switching the Mode: When switching between smart card interface mode and software standby mode, the following switching procedure should be followed in order to maintain the clock duty.

- When changing from smart card interface mode to software standby mode

[1] Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.

[2] Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.

[3] Write 0 to the CKE0 bit in SCR to halt the clock.

[4] Wait for one serial clock period.
During this interval, clock output is fixed at the specified level, with the duty preserved.

[5] Make the transition to the software standby state.

- When returning to smart card interface mode from software standby mode

[6] Exit the software standby state.

[7] Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

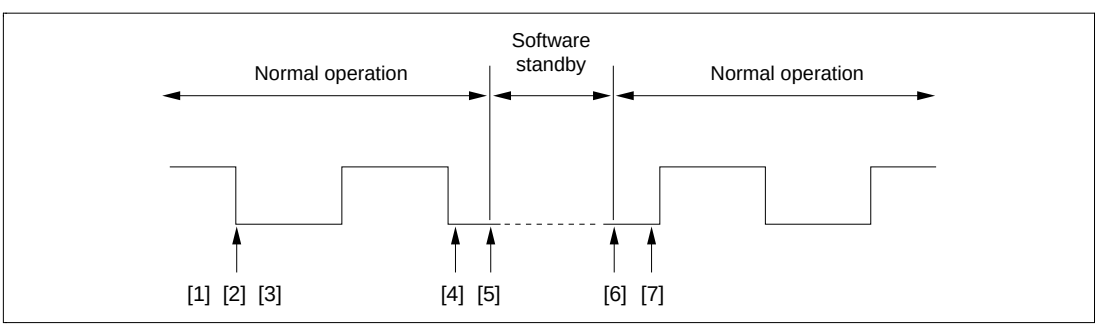


Figure 17-9 Clock Halt and Restart Procedure

Powering On: To secure the clock duty from power-on, the following switching procedure should be followed.

[1] The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.

[2] Fix the SCK pin to the specified output level with the CKE1 bit in SCR.

[3] Set SMR and SCMR, and switch to smart card mode operation.

[4] Set the CKE0 bit in SCR to 1 to start clock output.

17.3.8 Operation in Block Transfer Mode

Operation in block transfer mode is the same as in SCI asynchronous mode, except for the following points. For details, see section 16.3.2, Operation in Asynchronous Mode.

(1) Data Format

The data format is 8 bits with parity. There is no stop bit, but there is a 2-bit (1-bit or more in reception) error guard time.

Also, except during transmission (with start bit, data bits, and parity bit), the transmission pins go to the high-impedance state, so the signal lines must be fixed high with a pull-up resistor.

(2) Transmit/Receive Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock. The number of basic clock periods in a 1-bit transfer interval can be set to 32, 64, 372, or 256 with bits BCP1 and BCP0. For details, see section 17.3.5, Clock.

(3) ERS (FER) Flag

As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transmission and reception is not performed, this flag is always cleared to 0.

17.4 Usage Notes

The following points should be noted when using the SCI as a Smart Card interface.

Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode: In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (as determined by bits BCP1 and BCP0).

In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock. Figure 17-10 shows the receive data sampling timing when using a clock of 372 times the transfer rate.

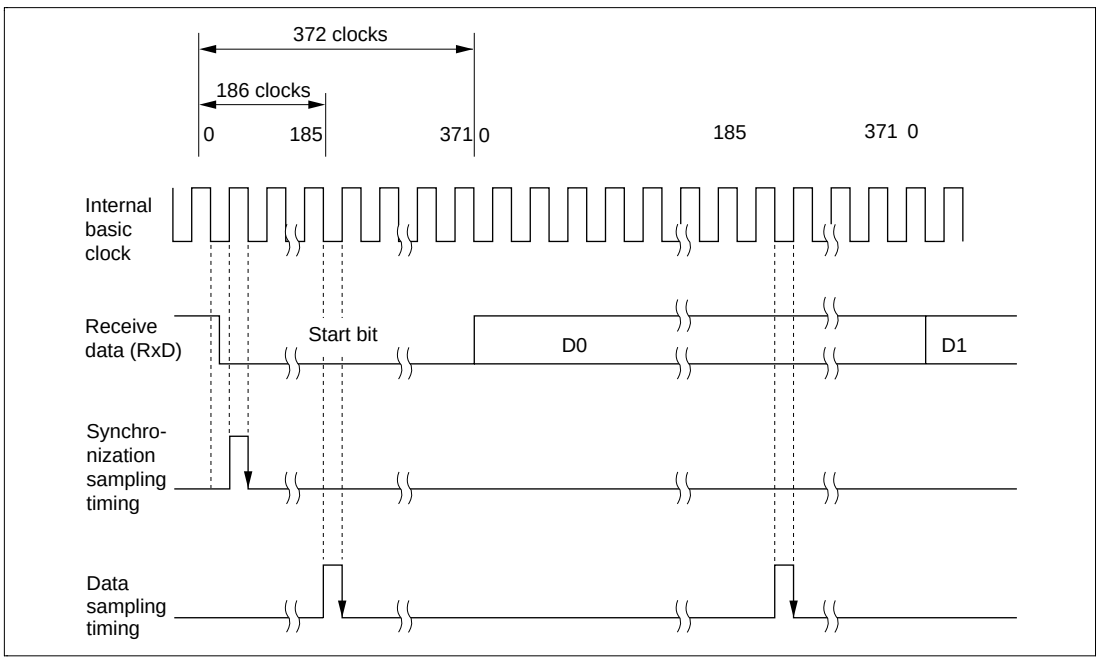


Figure 17-10 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate)

Thus the reception margin in asynchronous mode is given by the following formula.

Formula for reception margin in smart card interface mode

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

When D = 0.5 and F = 0,
$M = (0.5 - 1/2 \times 372) \times 100\%$
$= 49.866\%$

Retransfer Operations (Except Block Transfer Mode): Retransfer operations are performed by the SCI in receive mode and transmit mode as described below.

- Retransfer operation when SCI is in receive mode
  Figure 17-11 illustrates the retransfer operation when the SCI is in receive mode.

[1] If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1.

[4] If no error is found when the received parity bit is checked, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.
If DMAC or DTC data transfer by an RXI source is enabled, the contents of RDR can be read automatically. When the RDR data is read by the DMAC or DTC, the RDRF flag is automatically cleared to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the timing for error signal transmission.

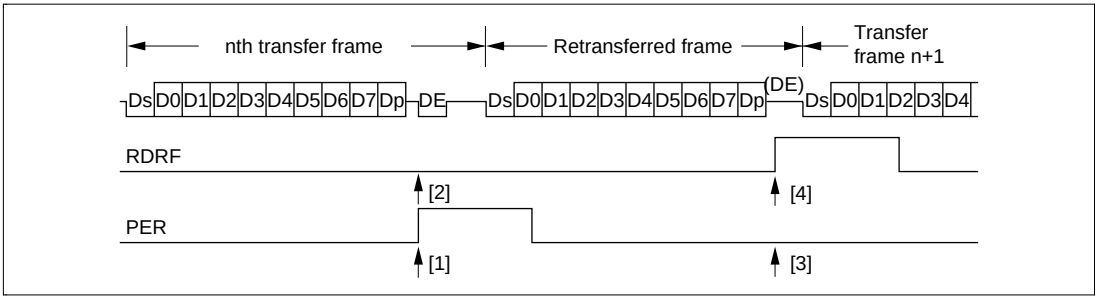


Figure 17-11 Retransfer Operation in SCI Receive Mode

- Retransfer operation when SCI is in transmit mode

  Figure 17-12 illustrates the retransfer operation when the SCI is in transmit mode.

[6] If an error signal is sent back from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[7] The TEND bit in SSR is not set for a frame for which an error signal indicating an abnormality is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.

[9] If an error signal is not sent back from the receiving end, transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DMAC and DTC by means of the TXI source is enabled, the next data can be written to TDR automatically. When data is written to TDR by the DMAC or DTC, the TDRE bit is automatically cleared to 0.

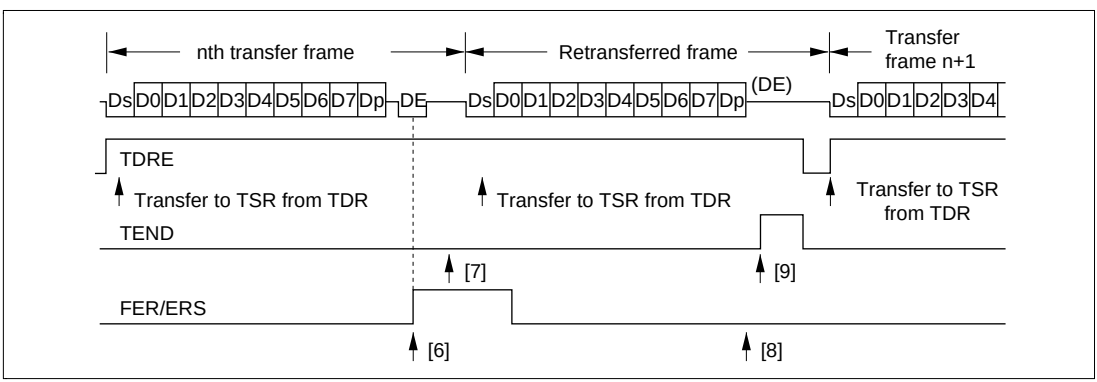


Figure 17-12 Retransfer Operation in SCI Transmit Mode

Section 18 I^2C Bus Interface [Option]

A two-channel I^2C bus interface is available as an option in the H8S/2633 Series. The I^2C bus interface is not available for the H8S/2633 Series. Observe the following notes when using this option.

1. For mask-ROM versions, a W is added to the part number in products in which this optional function is used.

   Examples: HD6432633WF

2. The product number is identical for F-ZTAT versions. However, be sure to inform your Hitachi sales representative if you will be using this option.

18.1 Overview

A two-channel I^2C bus interface is available for the H8S/2633 Series as an option. The I^2C bus interface conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

Each I^2C bus interface channel uses only one data line (SDA) and one clock line (SCL) to transfer data, saving board and connector space.

18.1.1 Features

- Selection of addressing format or non-addressing format
  - — I^2C bus format: addressing format with acknowledge bit, for master/slave operation
  - — Serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)
  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.

- Wait function in slave mode (I^2C bus format)

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.

- Three interrupt sources
  - — Data transfer end (including transmission mode transition with I^2C bus format and address reception after loss of master arbitration)
  - — Address match: when any slave address matches or the general call address is received in slave receive mode (I^2C bus format)
  - — Stop condition detection

- Selection of 16 internal clocks (in master mode)

- Direct bus drive (with SCL and SDA pins)
  - — Two pins—P35/SCL0 and P34/SDA0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.
  - — Two pins—P33/SCL1 and P32/SDA1—(normally CMOS pins) function as NMOS-only outputs when the bus drive function is selected.

18.1.2 Block Diagram

Figure 18-1 shows a block diagram of the I^2C bus interface.

Figure 18-2 shows an example of I/O pin connections to external circuits. Channel 0 I/O pins are NMOS open drains, and it is possible to apply voltages in excess of the power supply (PV_{CC}) voltage for this LSI. Set the upper limit of voltage applied to the power supply (PV_{CC}) power supply range + 0.3 V, i.e. 5.8 V. Channel 1 I/O pins are driven solely by NMOS, so in terms of appearance they carry out the same operations as an NMOS open drain. However, the voltage which can be applied to the I/O pins depends on the voltage of the power supply (PV_{CC}) of this LSI.

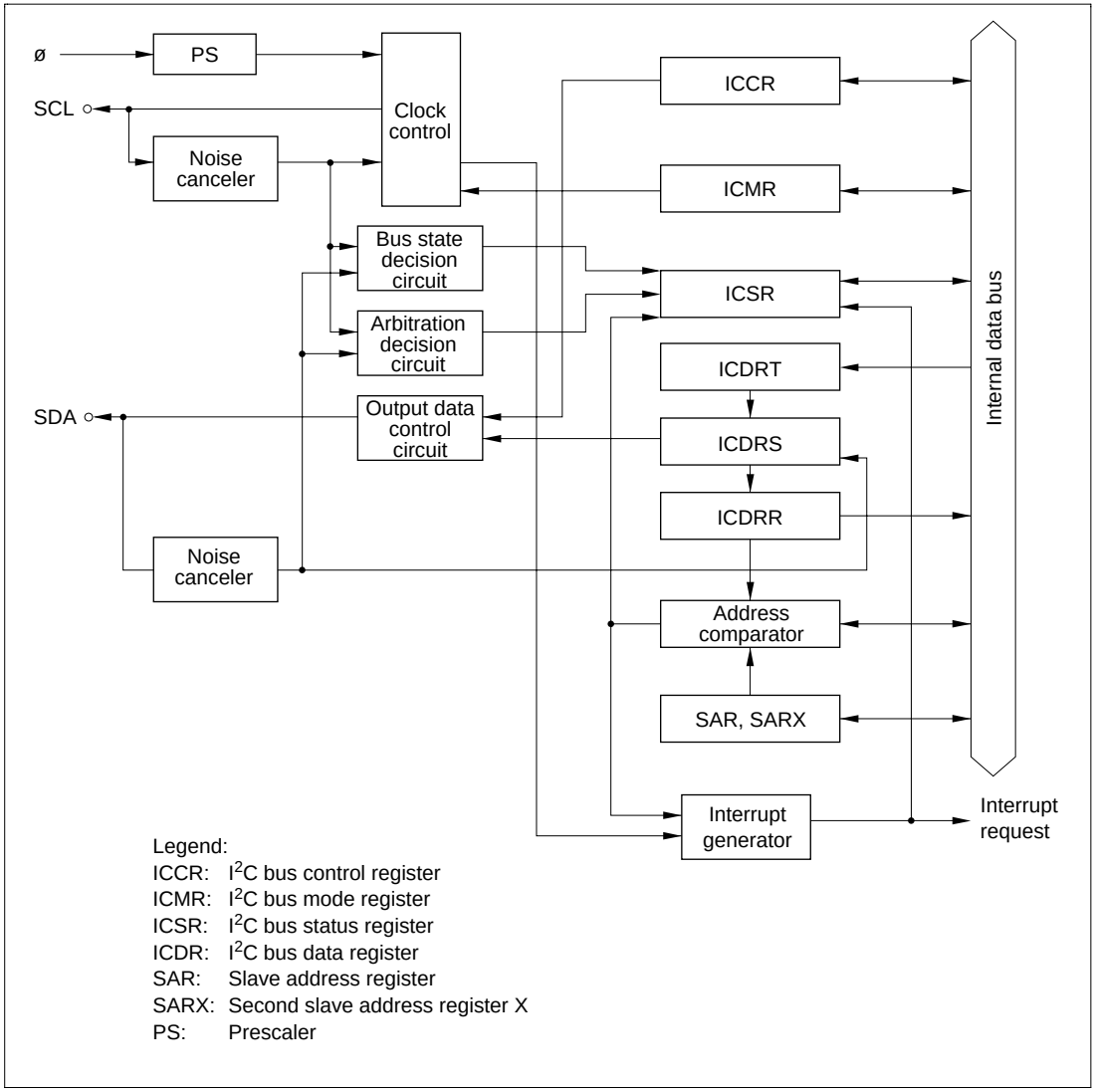


Figure 18-1 Block Diagram of I²C Bus Interface

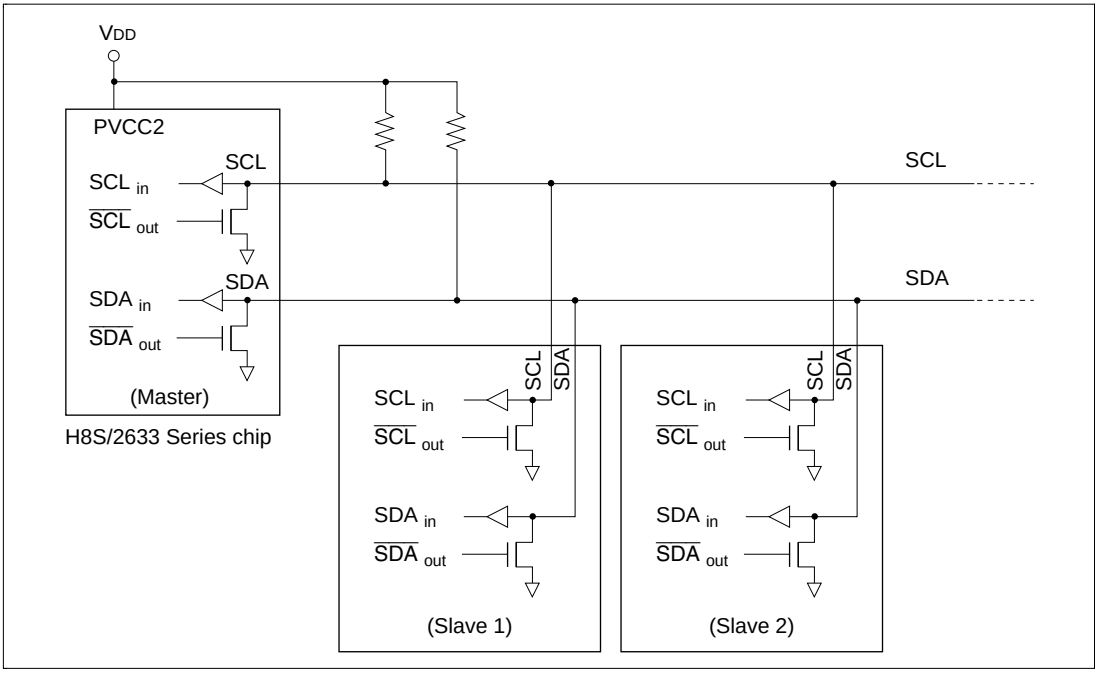


Figure 18-2 I²C Bus Interface Connections (Example: H8S/2633 Series Chip as Master)

18.1.3 Input/Output Pins

Table 18-1 summarizes the input/output pins used by the I²C bus interface.

Table 18-1 I²C Bus Interface Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock	SCL0	I/O	IIC0 serial clock input/output
	Serial data	SDA0	I/O	IIC0 serial data input/output
1	Serial clock	SCL1	I/O	IIC1 serial clock input/output
	Serial data	SDA1	I/O	IIC1 serial data input/output

Note: In the text, the channel subscript is omitted, and only SCL and SDA are used.

18.1.4 Register Configuration

Table 18-2 summarizes the registers of the I²C bus interface.

Table 18-2 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	I²C bus control register	ICCR0	R/W	H'01	H'FF78*[3]
	I²C bus status register	ICSR0	R/W	H'00	H'FF79*[3]
	I²C bus data register	ICDR0	R/W	—	H'FF7E*[2, 3]
	I²C bus mode register	ICMR0	R/W	H'00	H'FF7F*[2, 3]
	Slave address register	SAR0	R/W	H'00	H'FF7F*[2, 3]
	Second slave address register	SARX0	R/W	H'01	H'FF7E*[2, 3]
1	I²C bus control register	ICCR1	R/W	H'01	H'FF80*[3]
	I²C bus status register	ICSR1	R/W	H'00	H'FF81*[3]
	I²C bus data register	ICDR1	R/W	—	H'FF86*[2, 3]
	I²C bus mode register	ICMR1	R/W	H'00	H'FF87*[2, 3]
	Slave address register	SAR1	R/W	H'00	H'FF87*[2, 3]
	Second slave address register	SARX1	R/W	H'01	H'FF86*[2, 3]
Common	Serial control register X	SCRX	R/W	H'00	H'FDB4
	DDC switch register	DDCSWR	R/W	H'0F	H'FDB5
	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.

*2 The register that can be written or read depends on the ICE bit in the I²C bus control register. The slave address register can be accessed when ICE = 0, and the I²C bus mode register can be accessed when ICE = 1.

*3 The I²C bus interface registers are assigned to the same addresses as other registers. Register selection is performed by means of the IICE bit in the serial control register X (SCRX).

18.2 Register Descriptions

18.2.1 I²C Bus Data Register (ICDR)

Bit	:	7	6	5	4	3	2	1	0
		ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	:	7	6	5	4	3	2	1	0
		ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R	R	R	R	R	R	R	R

- ICDRS

Bit	:	7	6	5	4	3	2	1	0
		ICDRS7	ICDRS6	ICDRR5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	—	—	—	—	—	—	—	—

- ICDRT

Bit	:	7	6	5	4	3	2	1	0
		ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit	:							—	—
								TDRE	RDRF
Initial value	:							0	0
R/W	:							—	—

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If IIC is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If IIC is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	The next transmit data is in ICDR (ICDRT), or transmission cannot be started (Initial value) [Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmit mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I^2C bus format or serial format selected • When a stop condition is detected with the I^2C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	The next transmit data can be written in ICDR (ICDRT) [Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I^2C bus format or serial format selected • When using formatless mode in transmit mode (TRS = 1) • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • When a switch is made from receive mode (TRS = 0) to transmit mode (TRS = 1) after detection of a start condition

RDRF	Description
0	The data in ICDR (ICDRR) is invalid (Initial value) [Clearing condition] When ICDR (ICDRR) receive data is read in receive mode
1	The ICDR (ICDRR) receive data can be read [Setting condition] When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

18.2.2 Slave Address Register (SAR)

Bit	:	7	6	5	4	3	2	1	0
		SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAR is an 8-bit readable/writable register that stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SAR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 to 1—Slave Address (SVA6 to SVA0): Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select (FS): Used together with the FSX bit in SARX and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only

The FS bit also specifies whether or not SAR slave address recognition is performed in slave mode.

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	
SW	**FS**	**FSX**	**Operating Mode**
0	0	0	I^2C bus format • SAR and SARX slave addresses recognized
		1	I^2C bus format (Initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	—	—	Must not be set.

18.2.3 Second Slave Address Register (SARX)

Bit	:	7	6	5	4	3	2	1	0
		SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	:	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

SARX is an 8-bit readable/writable register that stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SARX is initialized to H'01 by a reset and in hardware standby mode.

Bits 7 to 1—Second Slave Address (SVAX6 to SVAX0): Set a unique address in bits SVAX6 to SVAX0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select X (FSX): Used together with the FS bit in SAR and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only

The FSX bit also specifies whether or not SARX slave address recognition is performed in slave mode. For details, see the description of the FS bit in SAR.

18.2.4 I^2C Bus Mode Register (ICMR)

Bit	:	7	6	5	4	3	2	1	0
		MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the master mode transfer clock frequency and the transfer bit count. ICMR is assigned to the same address as SAR. ICMR can be written and read only when the ICE bit is set to 1 in ICCR.

ICMR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—MSB-First/LSB-First Select (MLS): Selects whether data is transferred MSB-first or LSB-first.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

Do not set this bit to 1 when the I^2C bus format is used.

Bit 7

MLS	Description	
0	MSB-first	(Initial value)
1	LSB-first	

Bit 6—Wait Insertion Bit (WAIT): Selects whether to insert a wait between the transfer of data and the acknowledge bit, in master mode with the I^2C bus format. When WAIT is set to 1, after the fall of the clock for the final data bit, the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.

The setting of this bit is invalid in slave mode.

Bit 6

WAIT	Description	
0	Data and acknowledge bits transferred consecutively	(Initial value)
1	Wait inserted between data and acknowledge bits	

Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): These bits, together with the IICX1 (channel 1) or IICX0 (channel 0) bit in the SCRX register, select the serial clock frequency in master mode. They should be set according to the required transfer rate.

SCRX Bit 5 or 6	Bit 5	Bit 4	Bit 3		Transfer Rate					
IICX	CKS2	CKS1	CKS0	Clock	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz	ø = 25 MHz
0	0	0	0	ø/28	179 kHz	286 kHz	357 kHz	571 kHz*	714 kHz*	893 kHz*
			1	ø/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*	625 kHz*
		1	0	ø/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*	521 kHz*
			1	ø/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz	391 kHz
	1	0	0	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz	313 kHz
			1	ø/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz	250 kHz
		1	0	ø/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz	223 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
1	0	0	0	ø/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz	446 kHz*
			1	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz	313 kHz
		1	0	ø/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz	260 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
	1	0	0	ø/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz	156 kHz
			1	ø/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz	125 kHz
		1	0	ø/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz	112 kHz
			1	ø/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz	97.7 kHz

Note: * These rates are outside the ranges stipulated in the I^2C bus interface specifications (normal mode: max. 100 kHz, high-speed mode: max. 400 kHz).

Bits 2 to 0—Bit Counter (BC2 to BC0): Bits BC2 to BC0 specify the number of bits to be transferred next. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low..

The bit counter is initialized to 000 by a reset and when a start condition is detected. The value returns to 000 at the end of a data transfer, including the acknowledge bit.

Bit 2	Bit 1	Bit 0	Bits/Frame		
BC2	**BC1**	**BC0**	**Synchronous Serial Format**	**I²C Bus Format**	
0	0	0	8	9	(Initial value)
		1	1	2	
	1	0	2	3	
		1	3	4	
1	0	0	4	5	
		1	5	6	
	1	0	6	7	
		1	7	8	

18.2.5 I²C Bus Control Register (ICCR)

Bit	:	7	6	5	4	3	2	1	0
		ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

Note: * Only 0 can be written, for flag clearing.

ICCR is an 8-bit readable/writable register that enables or disables the I²C bus interface, enables or disables interrupts, selects master or slave mode and transmission or reception, enables or disables acknowledgement, confirms the I²C bus interface bus status, issues start/stop conditions, and performs interrupt flag confirmation.

ICCR is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—I²C Bus Interface Enable (ICE): Selects whether or not the I²C bus interface is to be used. When ICE is set to 1, port pins function as SCL and SDA input/output pins and transfer operations are enabled. When ICE is cleared to 0, the I²C bus interface module is halted and its internal states are cleared.

The SAR and SARX registers can be accessed when ICE is 0. The ICMR and ICDR registers can be accessed when ICE is 1.

Bit 7

ICE	Description	
0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function	(Initial value)
	I²C bus interface module internal states initialized	
	SAR and SARX can be accessed	
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus)	
	ICMR and ICDR can be accessed	

Bit 6—I²C Bus Interface Interrupt Enable (IEIC): Enables or disables interrupts from the I²C bus interface to the CPU.

Bit 6

IEIC	Description	
0	Interrupts disabled	(Initial value)
1	Interrupts enabled	

Bit 5—Master/Slave Select (MST)

Bit 4—Transmit/Receive Select (TRS)

MST selects whether the I²C bus interface operates in master mode or slave mode.

TRS selects whether the I²C bus interface operates in transmit mode or receive mode.

In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. In slave receive mode with the addressing format (FS = 0 or FSX = 0), hardware automatically selects transmit or receive mode according to the R/W bit in the first frame after a start condition.

Modification of the TRS bit during transfer is deferred until transfer of the frame containing the acknowledge bit is completed, and the changeover is made after completion of the transfer.

MST and TRS select the operating mode as follows.

Bit 5	Bit 4		
MST	**TRS**	**Operating Mode**	
0	0	Slave receive mode	(Initial value)
	1	Slave transmit mode	
1	0	Master receive mode	
	1	Master transmit mode	

Bit 5

MST	Description	
0	Slave mode	(Initial value)
	[Clearing conditions] 1. When 0 is written by software 2. When bus arbitration is lost after transmission is started in I^2C bus format master mode	
1	Master mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing condition 2) 2. When 1 is written in MST after reading MST = 0 (in case of clearing condition 2)	

Bit 4

TRS	Description	
0	Receive mode	(Initial value)
	[Clearing conditions] 1. When 0 is written by software (in cases other than setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (in case of clearing condition 3) 3. When bus arbitration is lost after transmission is started in I^2C bus format master mode 4. When the SW bit in DDCSWR changes from 1 to 0	
1	Transmit mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing conditions 3 and 4) 2. When 1 is written in TRS after reading TRS = 0 (in case of clearing conditions 3 and 4) 3. When a 1 is received as the R/W bit of the first frame in I^2C bus format slave mode	

Bit 3—Acknowledge Bit Judgement Selection (ACKE): Specifies whether the value of the acknowledge bit returned from the receiving device when using the I^2C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0.

In the H8S/2633 Series, the DTC can be used to perform continuous transfer. The DTC is activated when the IRTR interrupt flag is set to 1 (IRTR is one of two interrupt flags, the other being IRIC). When the ACKE bit is 0, the TDRE, IRIC, and IRTR flags are set on completion of data transmission, regardless of the value of the acknowledge bit. When the ACKE bit is 1, the TDRE, IRIC, and IRTR flags are set on completion of data transmission when the acknowledge bit is 0, and the IRIC flag alone is set on completion of data transmission when the acknowledge bit is 1.

When the DTC is activated, the TDRE, IRIC, and IRTR flags are cleared to 0 after the specified number of data transfers have been executed. Consequently, interrupts are not generated during continuous data transfer, but if data transmission is completed with a 1 acknowledge bit when the ACKE bit is set to 1, the DTC is not activated and an interrupt is generated, if enabled.

Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit 3

ACKE	Description	
0	The value of the acknowledge bit is ignored, and continuous transfer is performed	(Initial value)
1	If the acknowledge bit is 1, continuous transfer is interrupted	

Bit 2—Bus Busy (BBSY): The BBSY flag can be read to check whether the I^2C bus (SCL, SDA) is busy or free. In master mode, this bit is also used to issue start and stop conditions.

A high-to-low transition of SDA while SCL is high is recognized as a start condition, setting BBSY to 1. A low-to-high transition of SDA while SCL is high is recognized as a stop condition, clearing BBSY to 0.

To issue a start condition, use a MOV instruction to write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, use a MOV instruction to write 0 in BBSY and 0 in SCP. It is not possible to write to BBSY in slave mode; the I^2C bus interface must be set to master transmit mode before issuing a start condition. MST and TRS should both be set to 1 before writing 1 in BBSY and 0 in SCP.

Bit 2		
BBSY	**Description**	
0	Bus is free	(Initial value)
	[Clearing condition]	
	When a stop condition is detected	
1	Bus is busy	
	[Setting condition]	
	When a start condition is detected	

Bit 1—I²C Bus Interface Interrupt Request Flag (IRIC): Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set to 1 at the end of a data transfer, when a slave address or general call address is detected in slave receive mode, when bus arbitration is lost in master transmit mode, and when a stop condition is detected. IRIC is set at different times depending on the FS bit in SAR and the WAIT bit in ICMR. See section 18.3.6, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.

IRIC is cleared by reading IRIC after it has been set to 1, then writing 0 in IRIC.

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

Bit 1

IRIC	Description
0	Waiting for transfer, or transfer in progress (Initial value) [Clearing conditions] 1. When 0 is written in IRIC after reading IRIC = 1 2. When ICDR is written or read by the DTC (When the TDRE or RDRF flag is cleared to 0) (This is not always a clearing condition; see the description of DTC operation for details)
1	Interrupt requested [Setting conditions] • I^2C bus format master mode 1. When a start condition is detected in the bus line state after a start condition is issued (when the TDRE flag is set to 1 because of first frame transmission) 2. When a wait is inserted between the data and acknowledge bit when WAIT = 1 3. At the end of data transfer (at the rise of the 9th transmit/receive clock pulse, or at the fall of the 8th transmit/receive clock pulse when using wait insertion) 4. When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) 5. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • I^2C bus format slave mode 1. When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 2. When the general call address is detected (when FS = 0 and the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 3. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) 4. When a stop condition is detected (when the STOP or ESTP flag is set to 1) • Synchronous serial format 1. At the end of data transfer (when the TDRE or RDRF flag is set to 1) 2. When a start condition is detected with serial format selected When any other condition arises in which the TDRE or RDRF flag is set to 1

When, with the I^2C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set. The IRTR flag (the DTC start request flag) is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I^2C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set. The IRIC and IRTR flags are not cleared at the end of the specified number of transfers in continuous transfer using the DTC. The TDRE or RDRF flag is cleared, however, since the specified number of ICDR reads or writes have been completed.

Table 18-3 shows the relationship between the flags and the transfer states.

Table 18-3 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

Bit 0—Start Condition/Stop Condition Prohibit (SCP): Controls the issuing of start and stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

Bit 0	
SCP	**Description**
0	Writing 0 issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1 (Initial value) Writing is ignored

18.2.6 I^2C Bus Status Register (ICSR)

Bit	:	7	6	5	4	3	2	1	0
		ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written, for flag clearing.

ICSR is an 8-bit readable/writable register that performs flag confirmation and acknowledge confirmation and control.

ICSR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Error Stop Condition Detection Flag (ESTP): Indicates that a stop condition has been detected during frame transfer in I^2C bus format slave mode.

Bit 7

ESTP	Description	
0	No error stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in ESTP after reading ESTP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I^2C bus format slave mode Error stop condition detected [Setting condition] When a stop condition is detected during frame transfer	
	• In other modes No meaning	

Bit 6—Normal Stop Condition Detection Flag (STOP): Indicates that a stop condition has been detected after completion of frame transfer in I^2C bus format slave mode.

Bit 6

STOP	Description	
0	No normal stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in STOP after reading STOP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I^2C bus format slave mode Normal stop condition detected [Setting condition] When a stop condition is detected after completion of frame transfer	
	• In other modes No meaning	

Bit 5—I^2C Bus Interface Continuous Transmission/Reception Interrupt Request Flag (IRTR): Indicates that the I^2C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.

IRTR flag setting is performed when the TDRE or RDRF flag is set to 1. IRTR is cleared by reading IRTR after it has been set to 1, then writing 0 in IRTR. IRTR is also cleared automatically when the IRIC flag is cleared to 0.

Bit 5

IRTR	Description	
0	Waiting for transfer, or transfer in progress	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in IRTR after reading IRTR = 1	
	2. When the IRIC flag is cleared to 0	
1	Continuous transfer state	
	[Setting condition]	
	• In I^2C bus interface slave mode When the TDRE or RDRF flag is set to 1 when AASX = 1	
	• In other modes When the TDRE or RDRF flag is set to 1	

Bit 4—Second Slave Address Recognition Flag (AASX): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.

AASX is cleared by reading AASX after it has been set to 1, then writing 0 in AASX. AASX is also cleared automatically when a start condition is detected.

Bit 4

AASX	Description	
0	Second slave address not recognized	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in AASX after reading AASX = 1	
	2. When a start condition is detected	
	3. In master mode	
1	Second slave address recognized	
	[Setting condition]	
	When the second slave address is detected in slave receive mode and FSX = 0	

Bit 3—Arbitration Lost (AL): This flag indicates that arbitration was lost in master mode. The I^2C bus interface monitors the bus. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

AL is cleared by reading AL after it has been set to 1, then writing 0 in AL. In addition, AL is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 3

AL	Description
0	Bus arbitration won (Initial value) [Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AL after reading AL = 1
1	Arbitration lost [Setting conditions] 1. If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode 2. If the internal SCL line is high at the fall of SCL in master transmit mode

Bit 2—Slave Address Recognition Flag (AAS): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.

AAS is cleared by reading AAS after it has been set to 1, then writing 0 in AAS. In addition, AAS is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 2

AAS	Description
0	Slave address or general call address not recognized (Initial value) [Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AAS after reading AAS = 1 3. In master mode
1	Slave address or general call address recognized [Setting condition] When the slave address or general call address is detected in slave receive mode and FS = 0

Bit 1—General Call Address Recognition Flag (ADZ): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).

ADZ is cleared by reading ADZ after it has been set to 1, then writing 0 in ADZ. In addition, ADZ is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 1

ADZ	Description	
0	General call address not recognized	(Initial value)
	[Clearing conditions]	
	1. When ICDR data is written (transmit mode) or read (receive mode)	
	2. When 0 is written in ADZ after reading ADZ = 1	
	3. In master mode	
1	General call address recognized	
	[Setting condition]	
	When the general call address is detected in slave receive mode and (FSX = 0 or FS = 0)	

Bit 0—Acknowledge Bit (ACKB): Stores acknowledge data. In transmit mode, after the receiving device receives data, it returns acknowledge data, and this data is loaded into ACKB. In receive mode, after data has been received, the acknowledge data set in this bit is sent to the transmitting device.

When this bit is read, in transmission (when TRS = 1), the value loaded from the bus line (returned by the receiving device) is read. In reception (when TRS = 0), the value set by internal software is read.

Bit 0

ACKB	Description	
0	Receive mode: 0 is output at acknowledge output timing	(Initial value)
	Transmit mode: Indicates that the receiving device has acknowledged the data (signal is 0)	
1	Receive mode: 1 is output at acknowledge output timing	
	Transmit mode: Indicates that the receiving device has not acknowledged the data (signal is 1)	

18.2.7 Serial Control Register X (SCRX)

Bit	:	7	6	5	4	3	2	1	0
		—	IICX1	IICX0	IICE	FLSHE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCRX is an 8-bit readable/writable register that controls register access, the I^2C interface operating mode (when the on-chip IIC option is included), and on-chip flash memory control (F-ZTAT versions). If a module controlled by SCRX is not used, do not write 1 to the corresponding bit.

SCRX is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Reserved: Do not set 1.

Bit 6—I^2C Transfer Select 1 (IICX1): This bit, together with bits CKS2 to CKS0 in ICMR of IIC1, selects the transfer rate in master mode. For details, see section 18.2.4, I^2C Bus Mode Register (ICMR).

Bit 5—I^2C Transfer Select 0 (IICX0): This bit, together with bits CKS2 to CKS0 in ICMR of IIC0, selects the transfer rate in master mode. For details, see section 18.2.4, I^2C Bus Mode Register (ICMR).

Bit 4—I^2C Master Enable (IICE): Controls CPU access to the I^2C bus interface data and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR).

Bit 4

IICE	Description	
0	CPU access to I^2C bus interface data and control registers is disabled	(Initial value)
1	CPU access to I^2C bus interface data and control registers is enabled	

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls the operation of the flash memory in F-ZTAT versions. For details, see section 22, ROM.

Bits 2 to 0—Reserved: Do not set 1.

18.2.8 DDC Switch Register (DDCSWR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	CLR3	CLR2	CLR1	CLR0
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	W*2	W*2	W*2	W*2

Notes: 1. Should always be written with 0.
2. Always read as 1.

DDCSWR is an 8-bit readable/writable register that is used to initialize the IIC module.

DDCSWR is initialized to H'0F by a reset and in hardware standby mode.

Bits 7 to 4—Reserved: Should always be written with 0.

Bits 3 to 0—IIC Clear 3 to 0 (CLR3 to CLR0): These bits control initialization of the internal state of IIC0 and IIC1.

These bits can only be written to; if read they will always return a value of 1.

When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module(s), and the internal state of the IIC module(s) is initialized.

The write data for these bits is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.

When clearing is required again, all the bits must be written to in accordance with the setting.

Bit 3	Bit 2	Bit 1	Bit 0	
CLR3	**CLR2**	**CLR1**	**CLR0**	**Description**
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latches cleared
1	—	—	—	Invalid setting

18.2.9 Module Stop Control Register B (MSTPCRB)

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	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

MSTPCRB is an 8-bit readable/writable register that perform module stop mode control.

When the MSTPB4 or MSTPB3 bit is set to 1, operation of the corresponding IIC channel is halted at the end of the bus cycle, and a transition is made to module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRB is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 4—Module Stop (MSTPB4): Specifies IIC channel 0 module stop mode.

Bit 4

MSTPB4	Description	
0	IIC channel 0 module stop mode is cleared	
1	IIC channel 0 module stop mode is set	(Initial value)

Bit 3—Module Stop (MSTPB3): Specifies IIC channel 1 module stop mode.

Bit 3

MSTPB3	Description	
0	IIC channel 1 module stop mode is cleared	
1	IIC channel 1 module stop mode is set	(Initial value)

18.3 Operation

18.3.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats with an acknowledge bit. These are shown in figures 18-3 (a) and (b). The first frame following a start condition always consists of 8 bits.

The serial format is a non-addressing format with no acknowledge bit. Although start and stop conditions must be issued, this format can be used as a synchronous serial format. This is shown in figure 18-4.

Figure 18-5 shows the I²C bus timing.

The symbols used in figures 18-3 to 18-5 are explained in table 18-4.

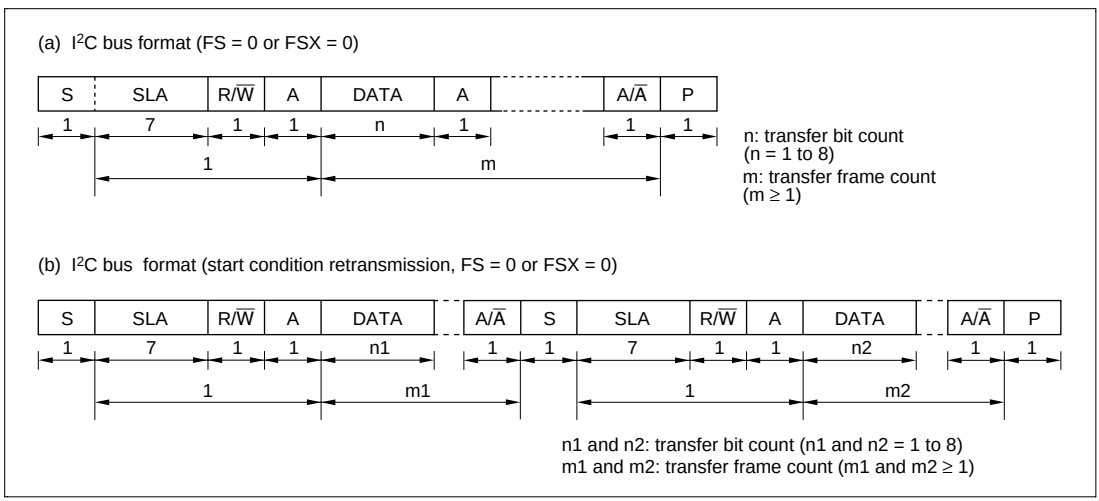


Figure 18-3 I²C Bus Data Formats (I²C Bus Formats)

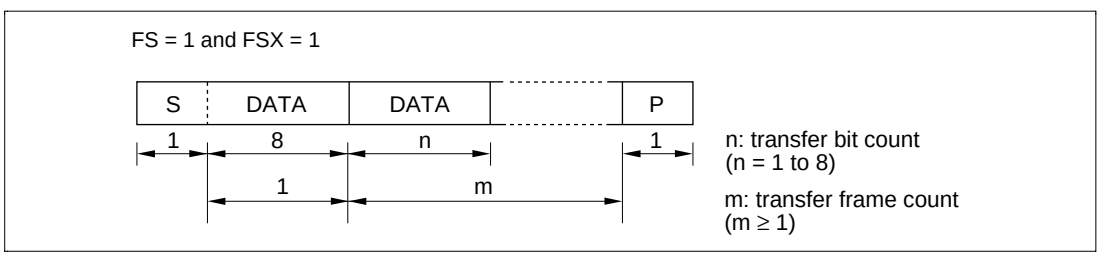


Figure 18-4 I²C Bus Data Format (Serial Format)

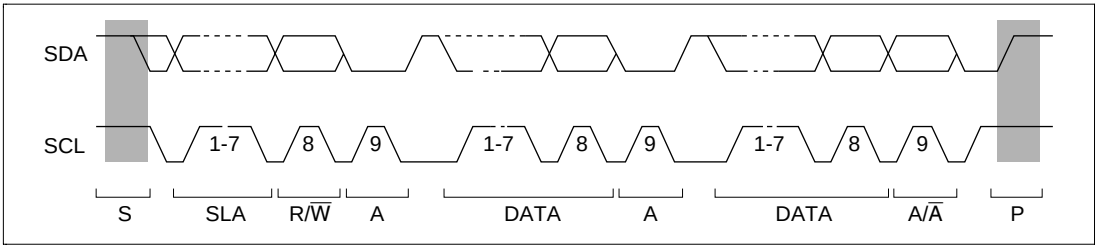


Figure 18-5 I²C Bus Timing

Table 18-4 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address, by which the master device selects a slave device
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device (the slave in master transmit mode, or the master in master receive mode) drives SDA low to acknowledge a transfer
DATA	Transferred data. The bit length is set by bits BC2 to BC0 in ICMR. The MSB-first or LSB-first format is selected by bit MLS in ICMR
P	Stop condition. The master device drives SDA from low to high while SCL is high

18.3.2 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The transmission procedure and operations are described below.

(1) Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in STCR, according to the operating mode.

(2) Read the BBSY flag in ICCR to confirm that the bus is free, then set bits MST and TRS to 1 in ICCR to select master transmit mode. Next, write 1 to BBSY and 0 to SCP. This changes SDA from high to low when SCL is high, and generates the start condition. The TDRE internal flag is then set to 1, and the IRIC and IRTR flags are also set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.

(3) With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction. Write the data (slave address + R/$\overline{W}$) to ICDR. The TDRE internal flag is then cleared to 0. The written address data is transferred to ICDRS, and the TDRE internal flag is set to 1 again. This is identified as indicating the end of the transfer, and so the IRIC flag is cleared to 0. The master device sequentially sends the transmit clock and the data written to

ICDR using the timing shown in figure 18-6. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

(4) When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, after one frame has been transmitted SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

(5) To continue transfer, write the next data to be transmitted into ICDR. After the data has been transferred to ICDRS and the TDRE internal flag has been set to 1, clear the IRIC flag to 0. Transmission of the next frame is performed in synchronization with the internal clock.

Data can be transmitted sequentially by repeating steps (4) and (5). To end transmission, after clearing the IRIC flag and transmitting the final data (with no more transmit data in ICDRT), write H'FF dummy data to ICDR, and then write 0 to BBSY and SCP in ICCR when the IRIC flag is set again. This changes SDA from low to high when SCL is high, and generates the stop condition.

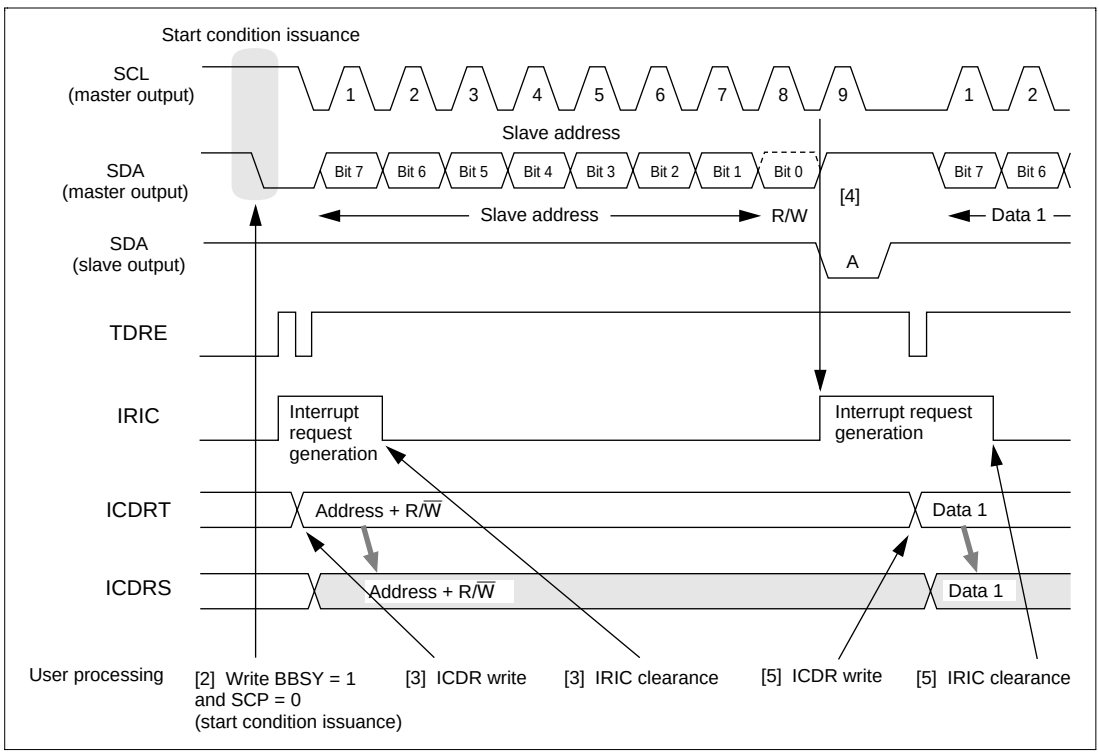


Figure 18-6 Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)

To transmit data continuously:

(6) Before the rise of the 9th transmit clock pulse for the data being transmitted, clear the IRIC flag to 0 and then write the next transmit data to ICDR.

(7) When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. At the same time, the next transmit data written into ICDR (ICDRT) is transferred to ICDRS, the TDRE internal flag is set to 1, and then the next frame is transmitted in synchronization with the internal clock.

Data can be transmitted continuously by repeating steps (6) and (7).

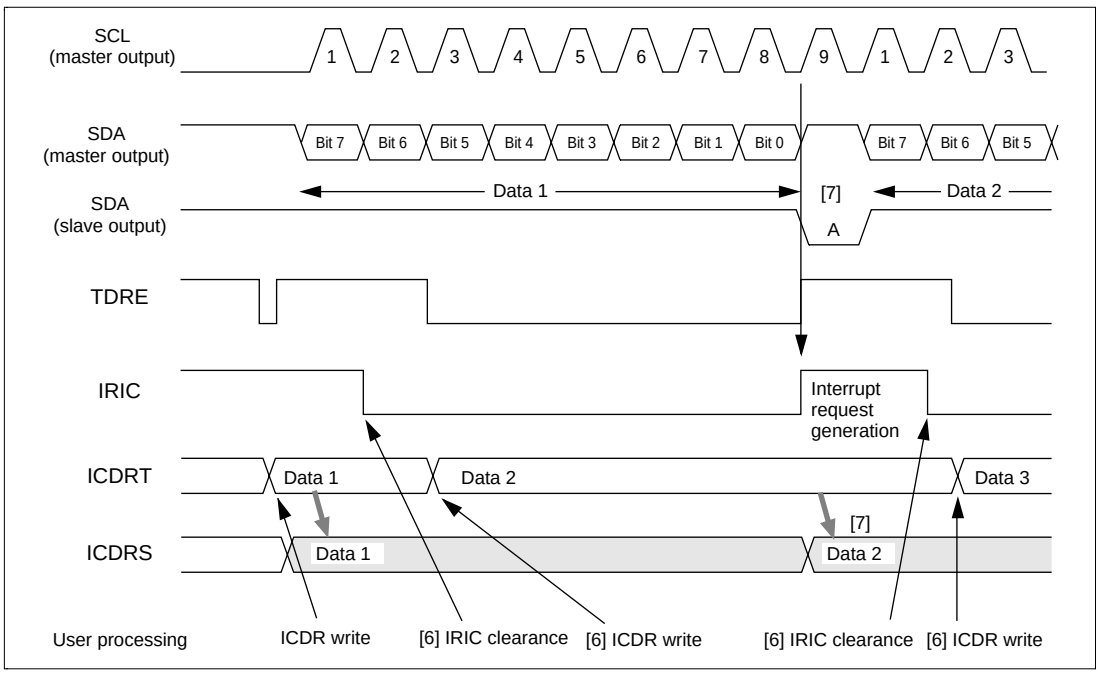


Figure 18-7 Example of Master Transmit Mode Continuous Transmit Operation Timing (MLS = WAIT = 0)

18.3.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data. The reception procedure and operations in master receive mode are described below.

(1) Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode. Also clear the ACKB bit in ICSR to 0 (acknowledge data setting).

(2) When ICDR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. In order to determine the end of reception, the IRIC flag in ICCR must be cleared beforehand.

(3) The master device drives SDA at the 9th receive clock pulse to return an acknowledge signal. When one frame of data has been received, the IRIC flag in ICCR is set to 1 at the rise of the 9th receive clock pulse. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If reception of the next frame ends before the ICDR read/IRIC flag clearing in (4) is performed, SCL is automatically fixed low in synchronization with the internal clock.

(4) Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Data can be received continuously by repeating steps (3) and (4). As the RDRF internal flag is cleared to 0 when reception is started after initially switching from master transmit mode to master receive mode, reception of the next frame of data is started automatically. To halt reception, the TRS bit must be set to 1 before the rise of the receive clock for the next frame.

To halt reception, set the TSR bit to 1, read ICDR, then write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

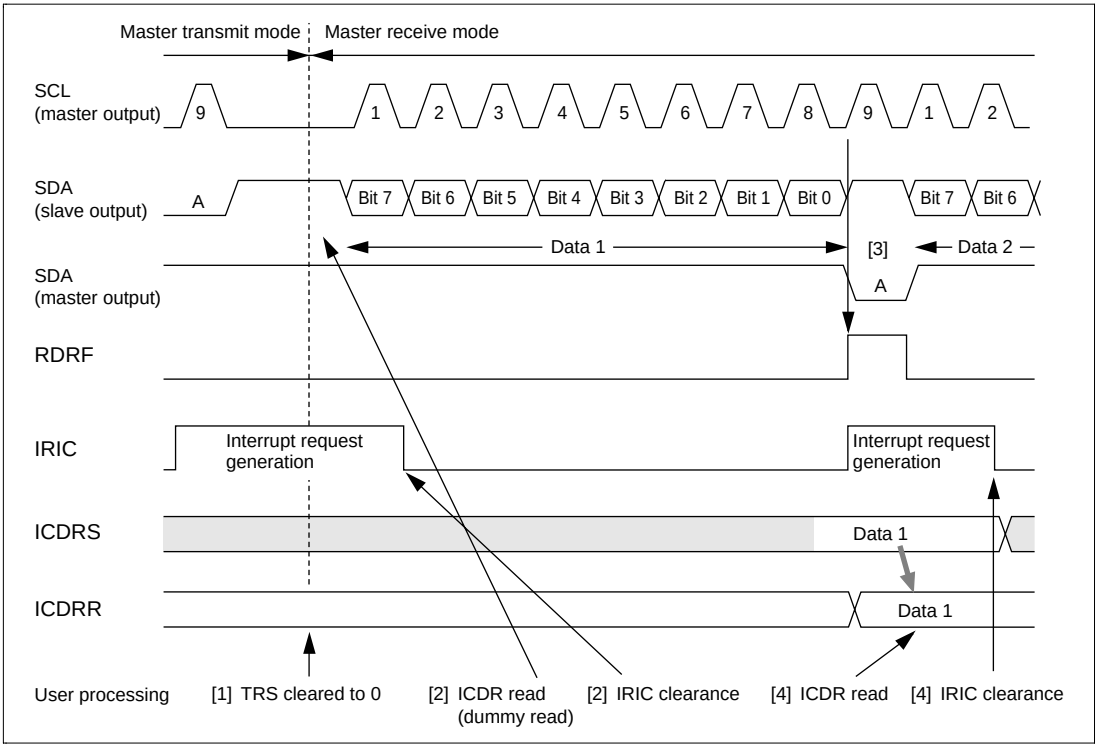


Figure 18-8 Example of Master Receive Mode Operation Timing (MLS = WAIT = ACKB = 0)

18.3.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

(1) Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

(2) When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.

(3) When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.

(4) At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag

has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.

(5) Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps (4) and (5). When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

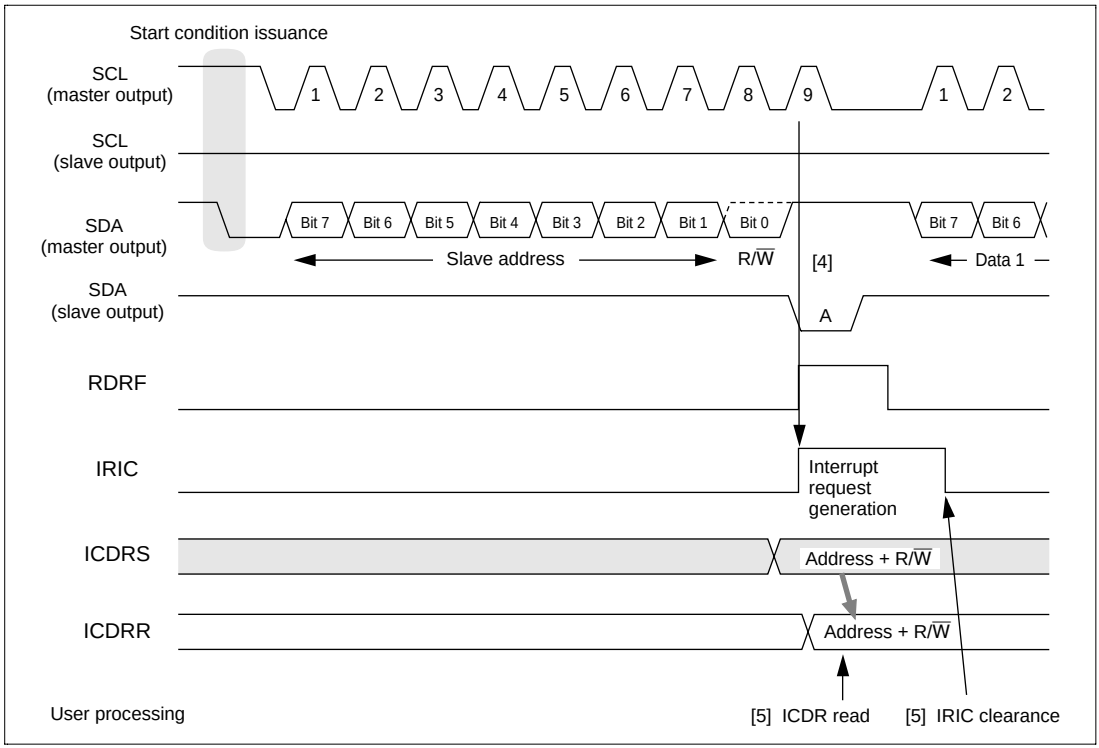


Figure 18-9 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0)

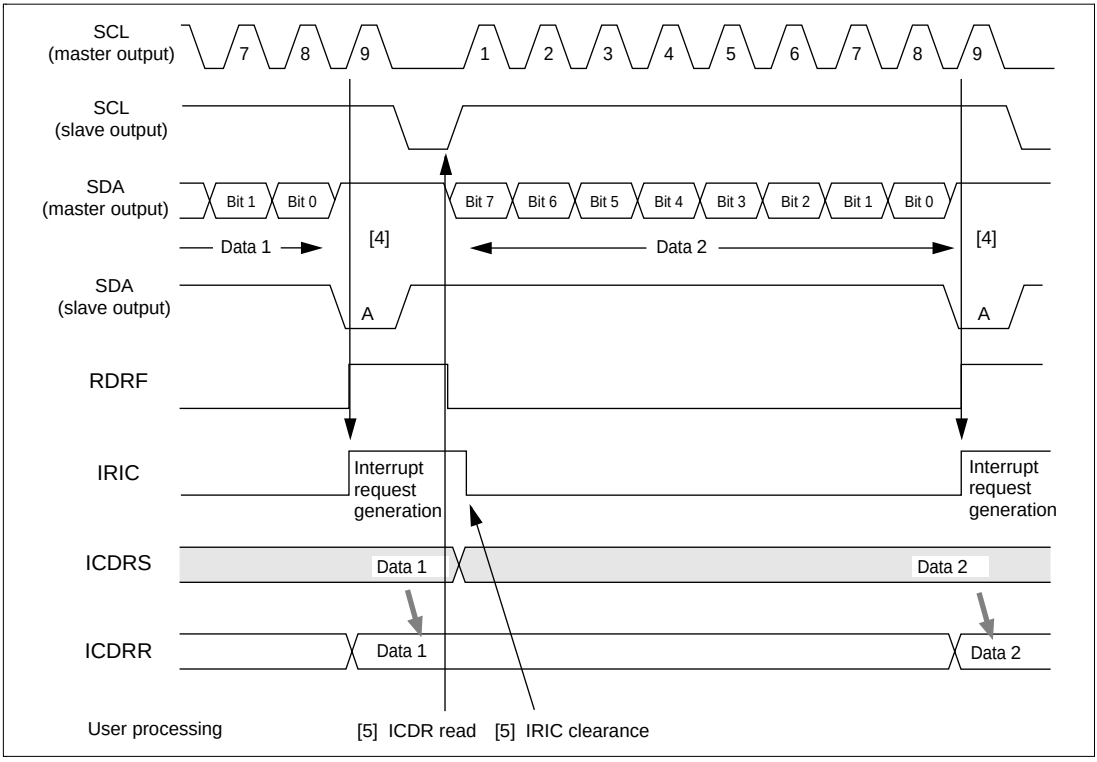


Figure 18-10 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0)

18.3.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

(1) Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

(2) When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. .If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRF flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

(3) After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The

slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 18-11.

(4) When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

(5) To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps (4) and (5). To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

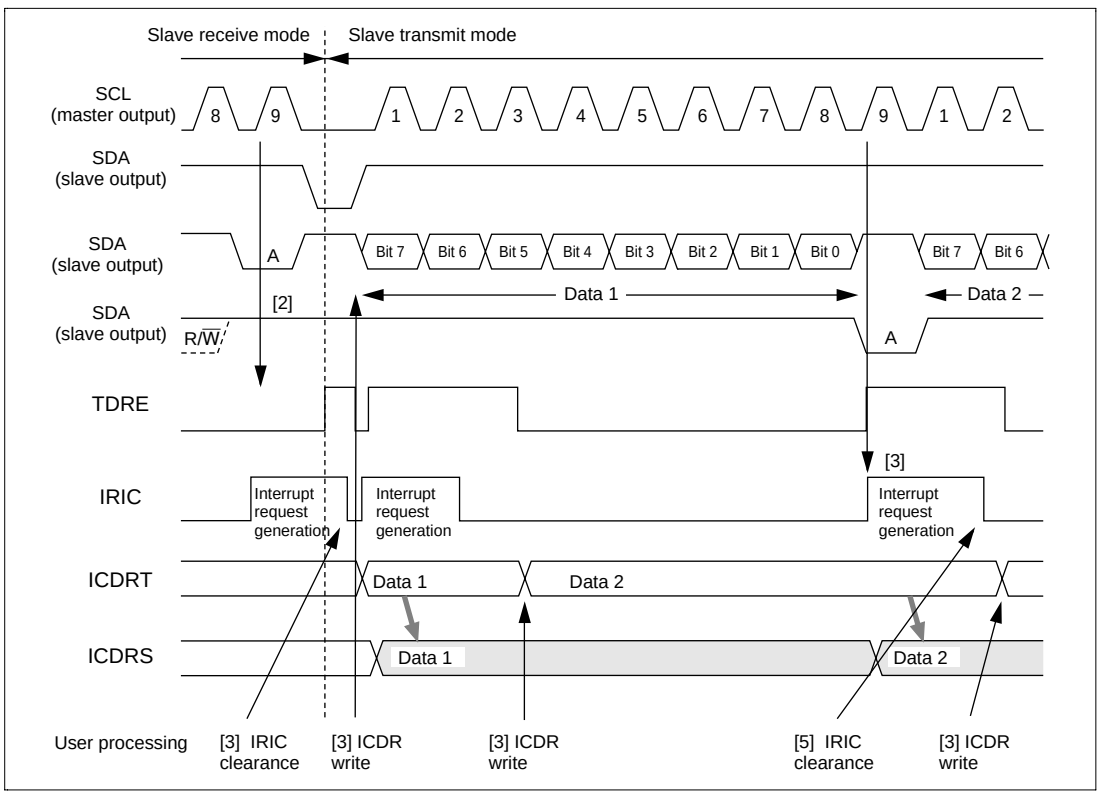


Figure 18-11 Example of Slave Transmit Mode Operation Timing (MLS = 0)

18.3.6 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 18-12 shows the IRIC set timing and SCL control.

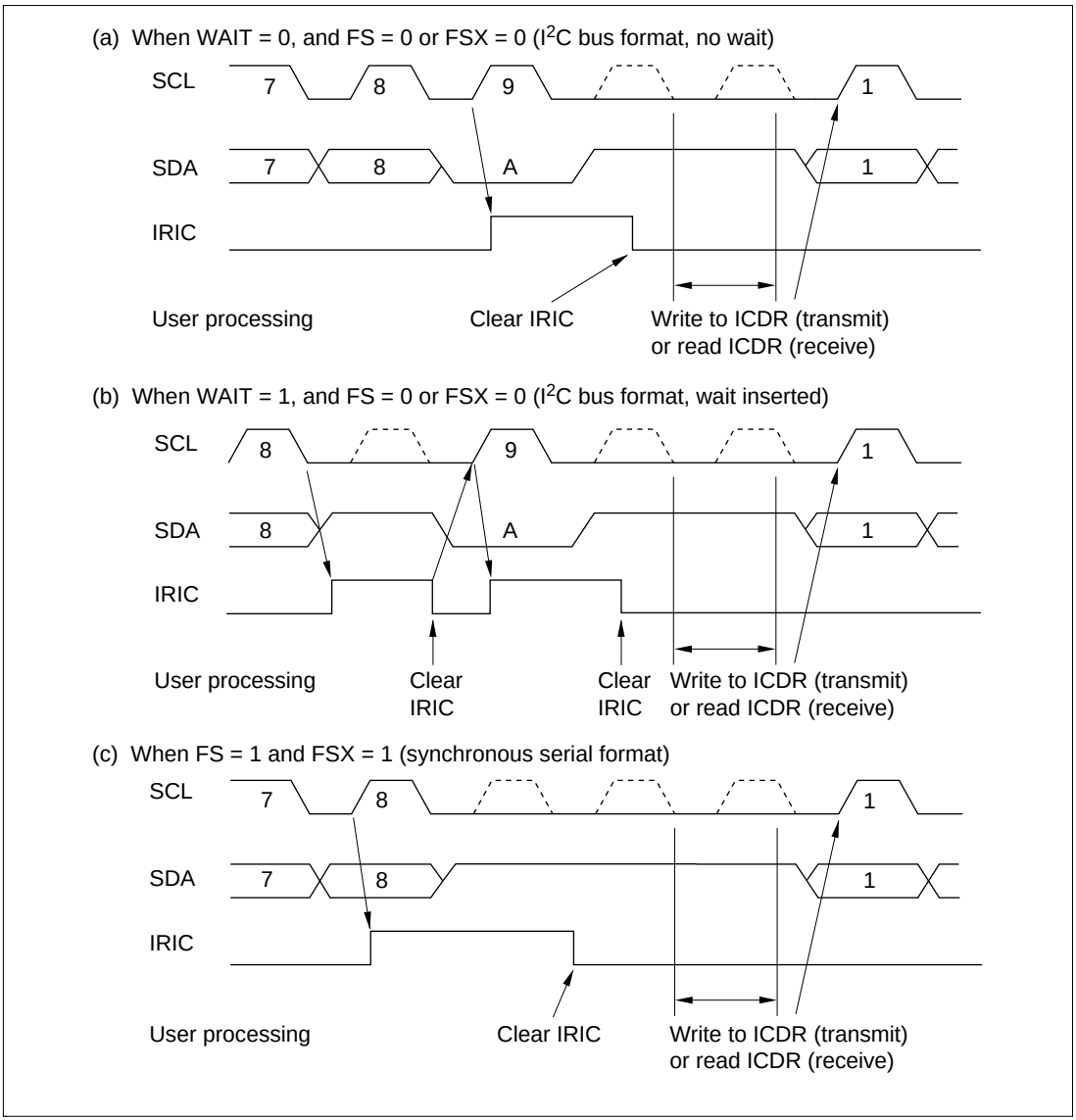


Figure 18-12 IRIC Setting Timing and SCL Control

18.3.7 Operation Using the DTC

The I^2C bus format provides for selection of the slave device and transfer direction by means of the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, indication of the last frame, and so on. Therefore, continuous data transfer using the DTC must be carried out in conjunction with CPU processing by means of interrupts.

Table 18-5 shows some examples of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

Table 18-5 Examples of Operation Using the DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception by CPU (ICDR read)
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: End condition issuance by CPU	Not necessary	Automatic clearing on detection of end condition during transmission of dummy data (H'FF)	Not necessary
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception: Actual data count

18.3.8 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 18-13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

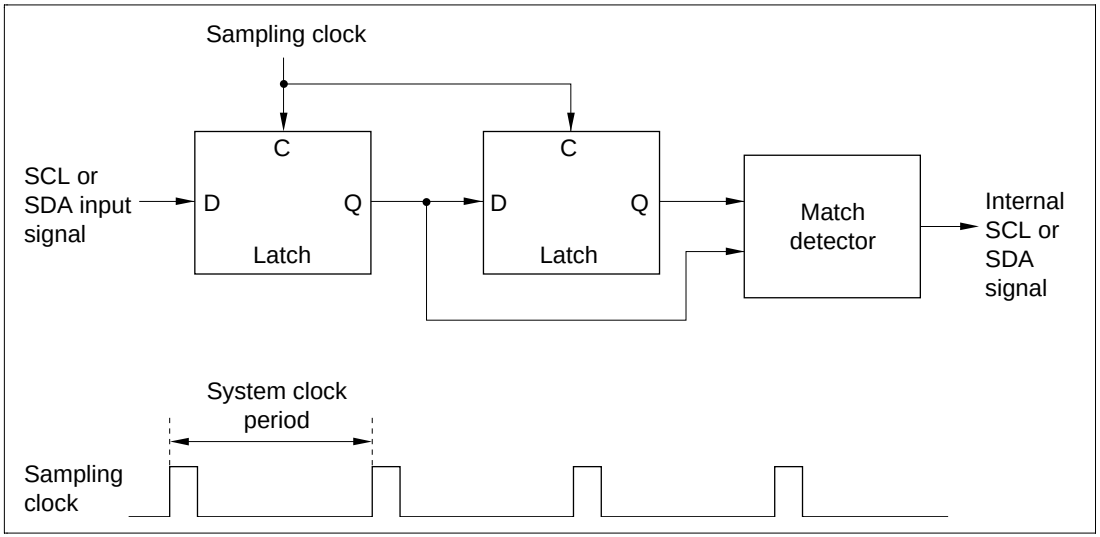


Figure 18-13 Block Diagram of Noise Canceler

18.3.9 Sample Flowcharts

Figures 18-14 to 18-17 show sample flowcharts for using the I^2C bus interface in each mode.

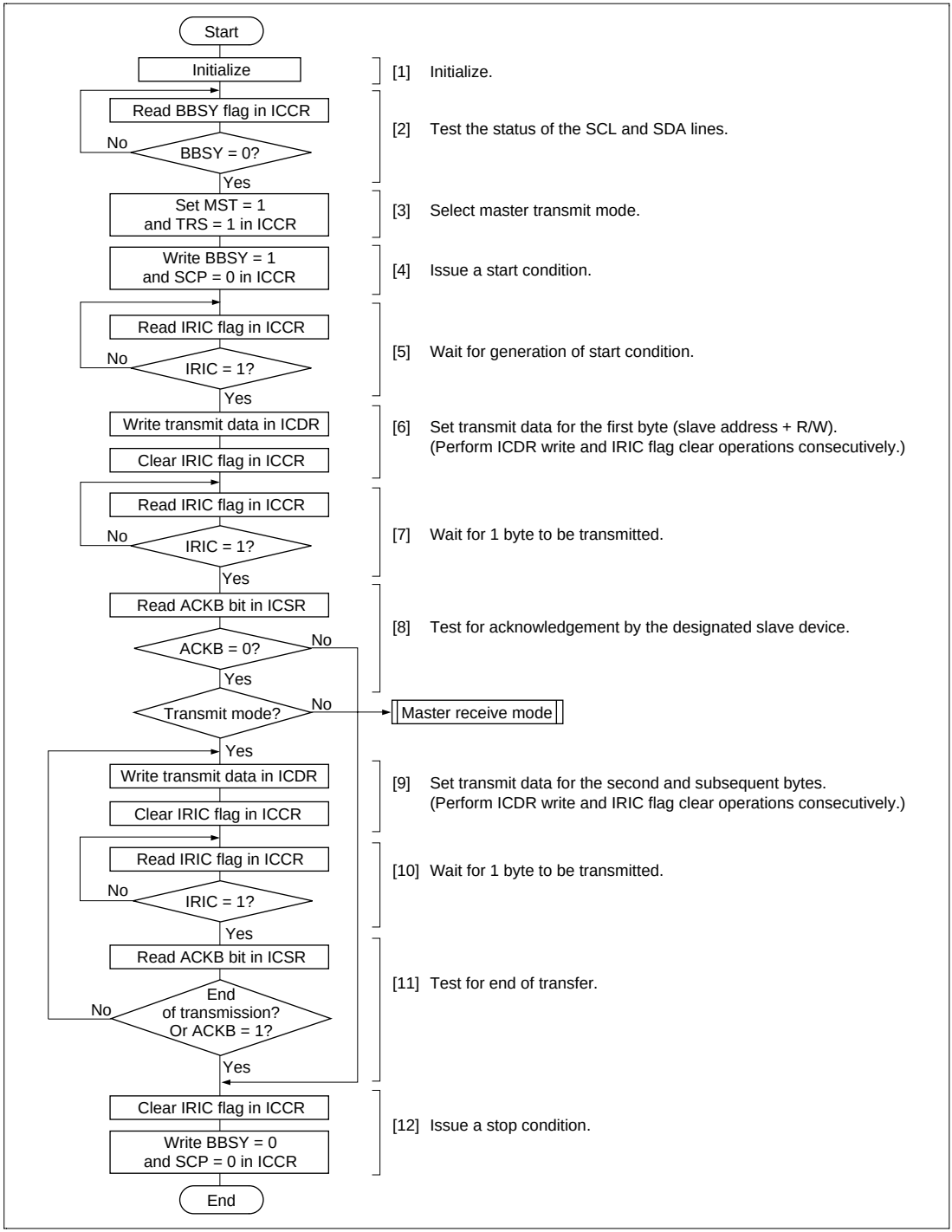


Figure 18-14 Flowchart for Master Transmit Mode (Example)

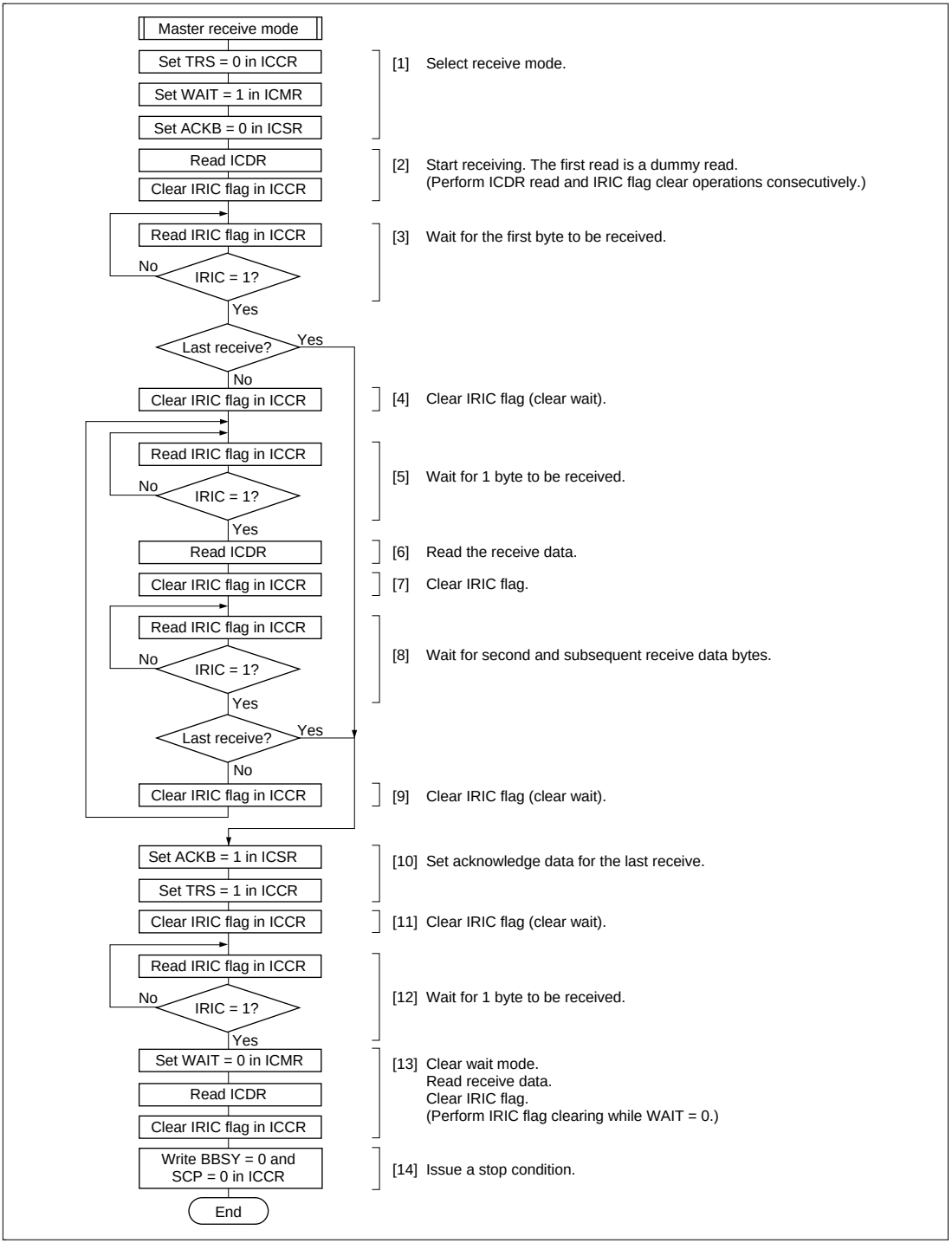


Figure 18-15 Flowchart for Master Receive Mode (Example)

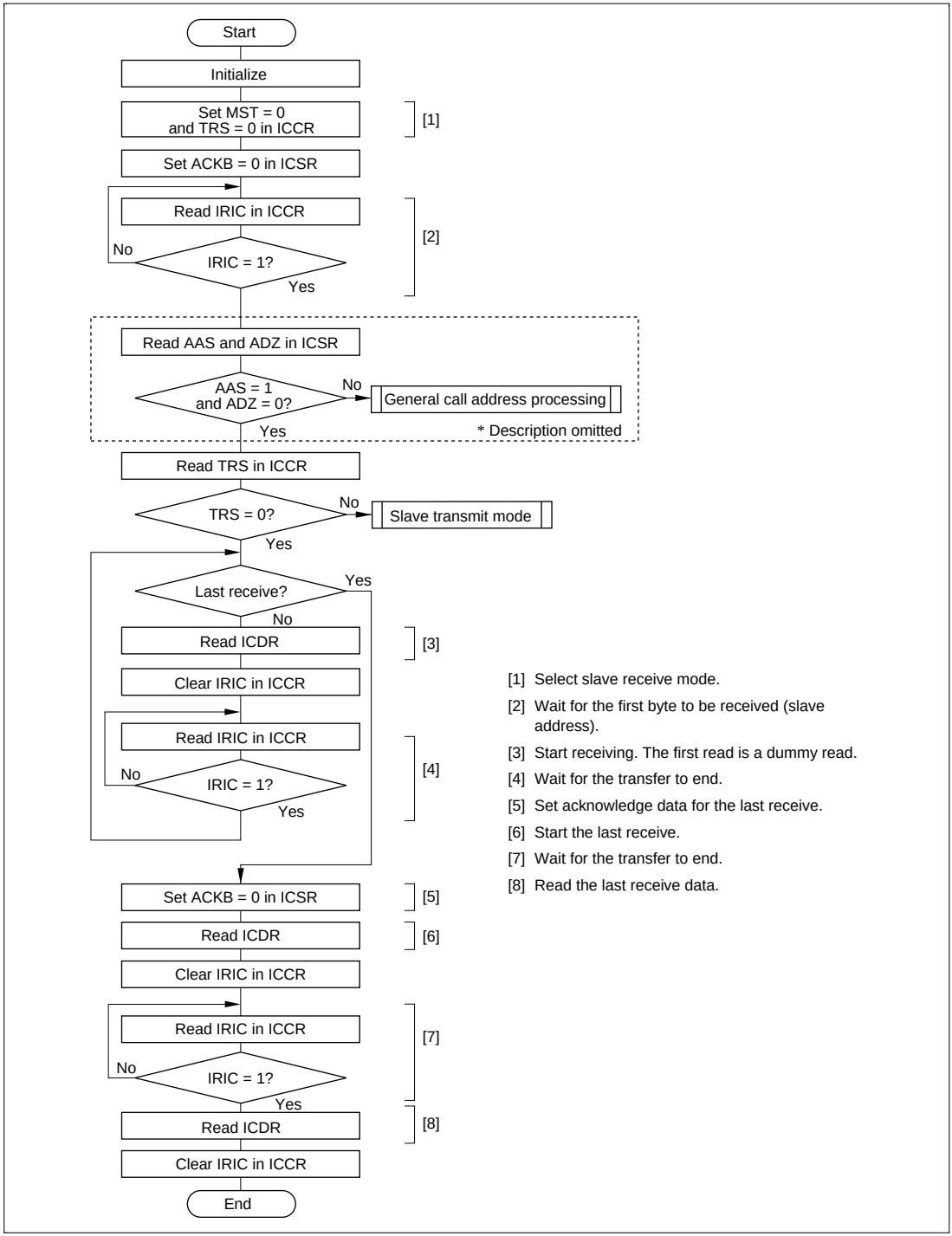


Figure 18-16 Flowchart for Slave Transmit Mode (Example)

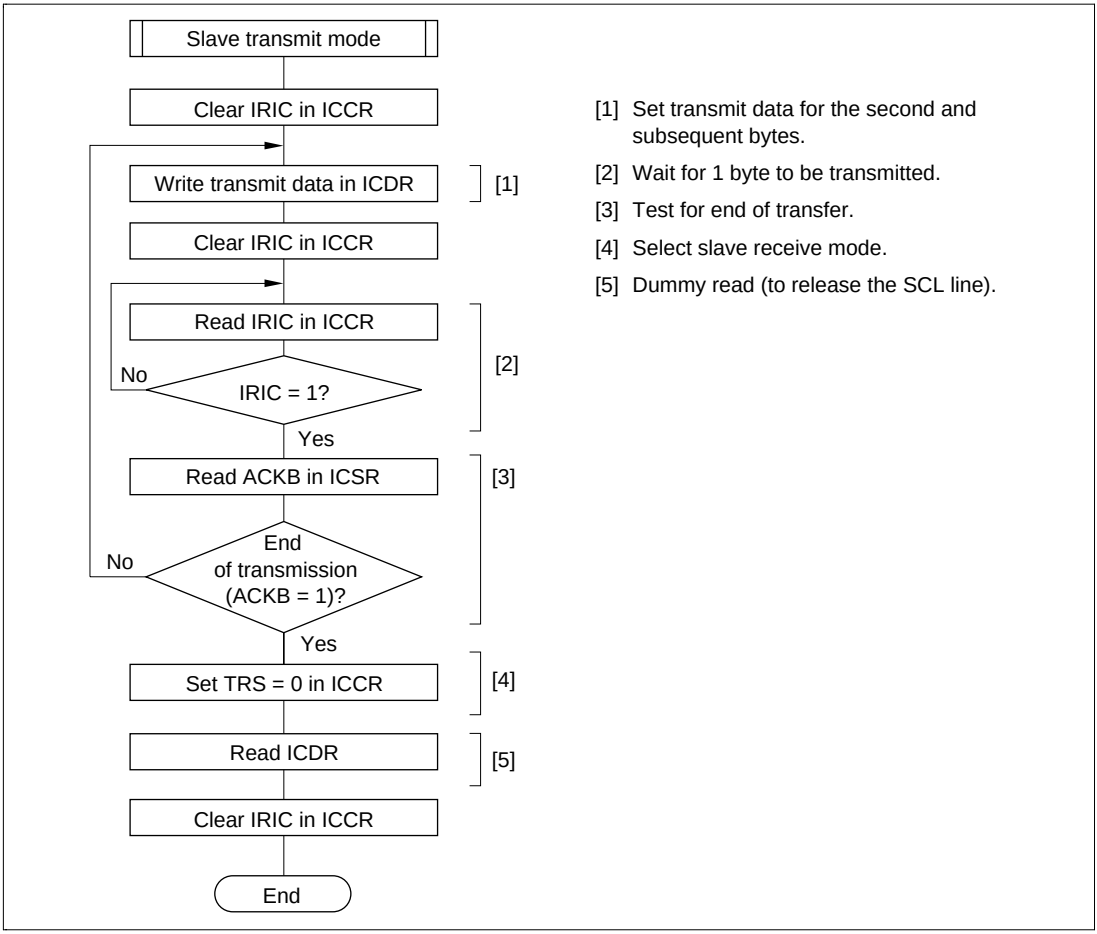


Figure 18-17 Flowchart for Slave Receive Mode (Example)

18.3.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed by (1) setting bits CLR3 to CLR0 in the DDCSWR register or (2) clearing the ICE bit. For details of settings for bits CLR3 to CLR0, see section 18.2.8, DDC Switch Register (DDCSWR).

Scope of Initialization: The initialization executed by this function covers the following items:

- TDRE and RDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, DDCSWR, STCR)
- Internal latches used to retain register read information for setting/clearing flags in the ICMR, ICCR, ICSR, and DDCSWR registers
- The value of the ICMR register bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

Notes on Initialization:

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is performed by means of the DDCSWR register, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBST bit to 0, and wait for two transfer rate clock cycles.

3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
4. Initialize (re-set) the IIC registers.

18.4 Usage Notes

- In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
  - — Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - — Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)

- Table 18-6 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 18-6 I^2C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	Figure 25-33 (reference)
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – 3t_{cyc}$		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

- SCL and SDA input is sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 25-11 in section 25, Electrical Characteristics. Note that the I^2C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.

- The I^2C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I^2C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I^2C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table below.

Table 18-7 Permissible SCL Rise Time (t_{Sr}) Values

			Time Indication						
IICX	t_{cyc} Indication		I^2C Bus Specification (Max.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz	ø = 25 MHz
0	7.5t_{cyc}	Standard mode	1000 ns	1000 ns	937 ns	750 ns	468 ns	375 ns	300 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns
1	17.5t_{cyc}	Standard mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns	875 ns	700 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns

- The I^2C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I^2C bus interface SCL and SDA output timing is prescribed by t_{Scyc} and t_{cyc}, as shown in table 18-6. However, because of the rise and fall times, the I^2C bus interface specifications may not be satisfied at the maximum transfer rate. Table 18-8 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

  t_{BUFO} fails to meet the I^2C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

  t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I^2C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be

investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

Table 18-8 I²C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]							
			t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz	ø = 25 MHz
t_{SCLHO}	$0.5t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*1	1000*1	1000*1	1000*1	1000*1	1000*1
t_{BUFO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4700	3800*1	3875*1	3900*1	3938*1	3950*1	3960*1
		High-speed mode	–300	1300	750*1	825*1	850*1	888*1	900*1	910*1
t_{STAHO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700	4710
		High-speed mode	–250	600	800	875	900	938	950	960
t_{STASO}	$1t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100	4080
		High-speed mode	–300	600	1350	1200	1150	1075	1050	1030
t_{SDASO} (master)	$1t_{SCLLO}$*2 – $3t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550	3580
		High-speed mode	–200	100	400	625	700	813	850	880
t_{SDASO} (slave)	$1t_{SCLL}$*2 – $3t_{cyc}$*2 ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550	3580
		High-speed mode	–300	100	400	625	700	813	850	880

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]							
			t_{Sr}/t_{Sf} Influence (Max.)	I^2C Bus Specification (Min.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz	ø = 25 MHz
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188	150	120
		High-speed mode	0	0	600	375	300	188	150	120

Notes: *1 Does not meet the I^2C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I^2C bus interface specifications are met must be determined in accordance with the actual setting conditions.

*2 Calculated using the I^2C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

- Note on ICDR Read at End of Master Reception

  To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data.

  If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, and the bus has been released, then read the ICDR register with TRS cleared to 0.

  Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

  Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 18.18 (after confirming that the BBSY bit has been cleared to 0 in the ICCR register).

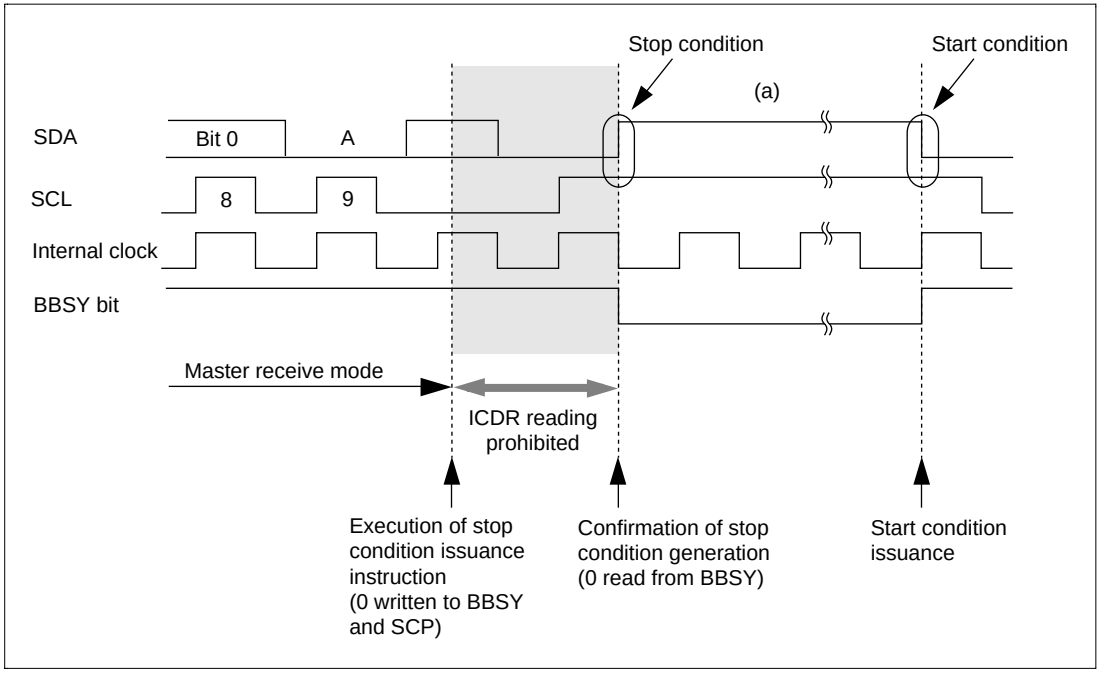
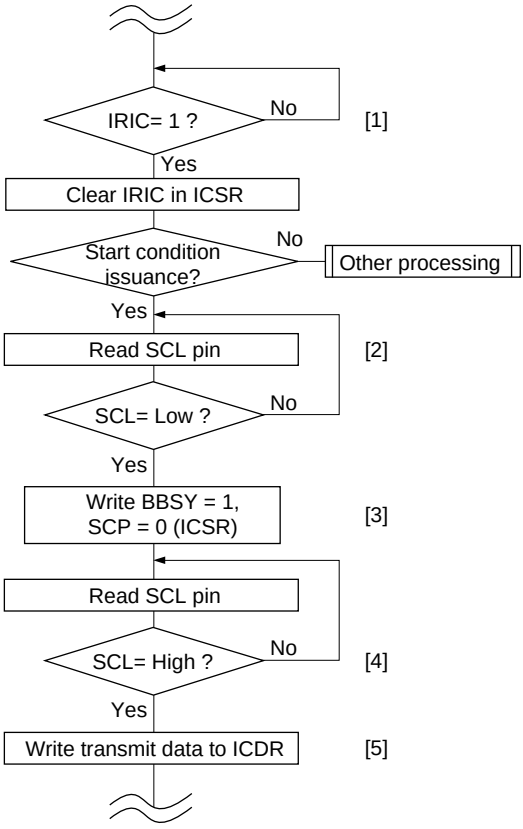


Figure 18-18 Points for Attention Concerning Reading of Master Receive Data

- Notes on Start Condition Issuance for Retransmission

  Figure 18-19 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.



[1] Wait for end of 1-byte transfer

[2] Determine whether SCL is low

[3] Issue restart condition instruction for retransmission

[4] Determine whether SCL is high

[5] Set transmit data (slave address + R/$\overline{W}$)

Note: Program so that processing from [3] to [5] is executed continuously.

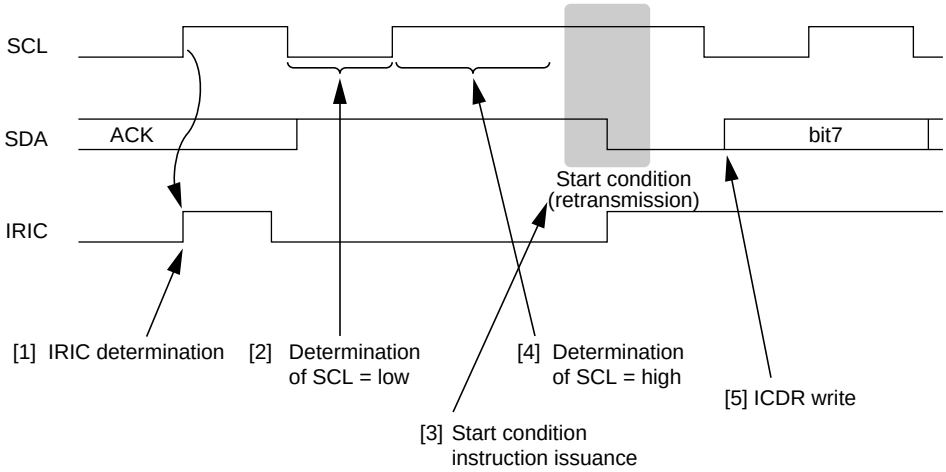


Figure 18-19 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

- Notes on I^2C Bus Interface Stop Condition Instruction Issuance

  If the rise time of the 9th SCL acknowledge exceeds the specification because the bus load capacitance is large, or if there is a slave device of the type that drives SCL low to effect a wait, issue the stop condition instruction after reading SCL and determining it to be low, as shown below.

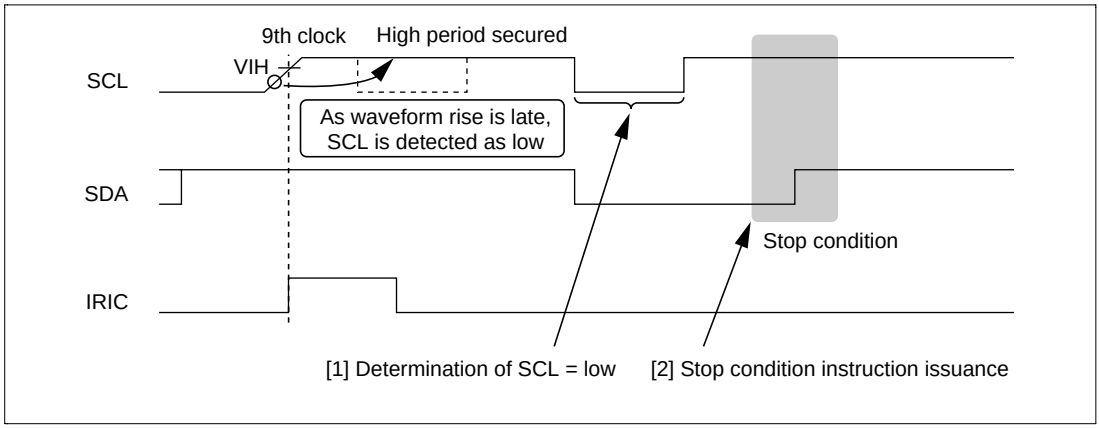


Figure 18-20 Timing of Stop Condition Issuance

Section 19 A/D Converter

19.1 Overview

The H8S/2633 Series incorporates a successive approximation type 10-bit A/D converter that allows up to sixteen analog input channels to be selected.

19.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Sixteen input channels
- Settable analog conversion voltage range
  - — Conversion of analog voltages with the reference voltage pin (Vref) as the analog reference voltage
- High-speed conversion
  - — Minimum conversion time: 10.64 µs per channel (at 25 MHz operation)
- Choice of single mode or scan mode
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - — Choice of software or timer conversion start trigger (TPU or 8-bit timer), or $\overline{\text{ADTRG}}$ pin
- A/D conversion end interrupt generation
  - — A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion
- Module stop mode can be set
  - — As the initial setting, A/D converter operation is halted. Register access is enabled by exiting module stop mode.

19.1.2 Block Diagram

Figure 19-1 shows a block diagram of the A/D converter.

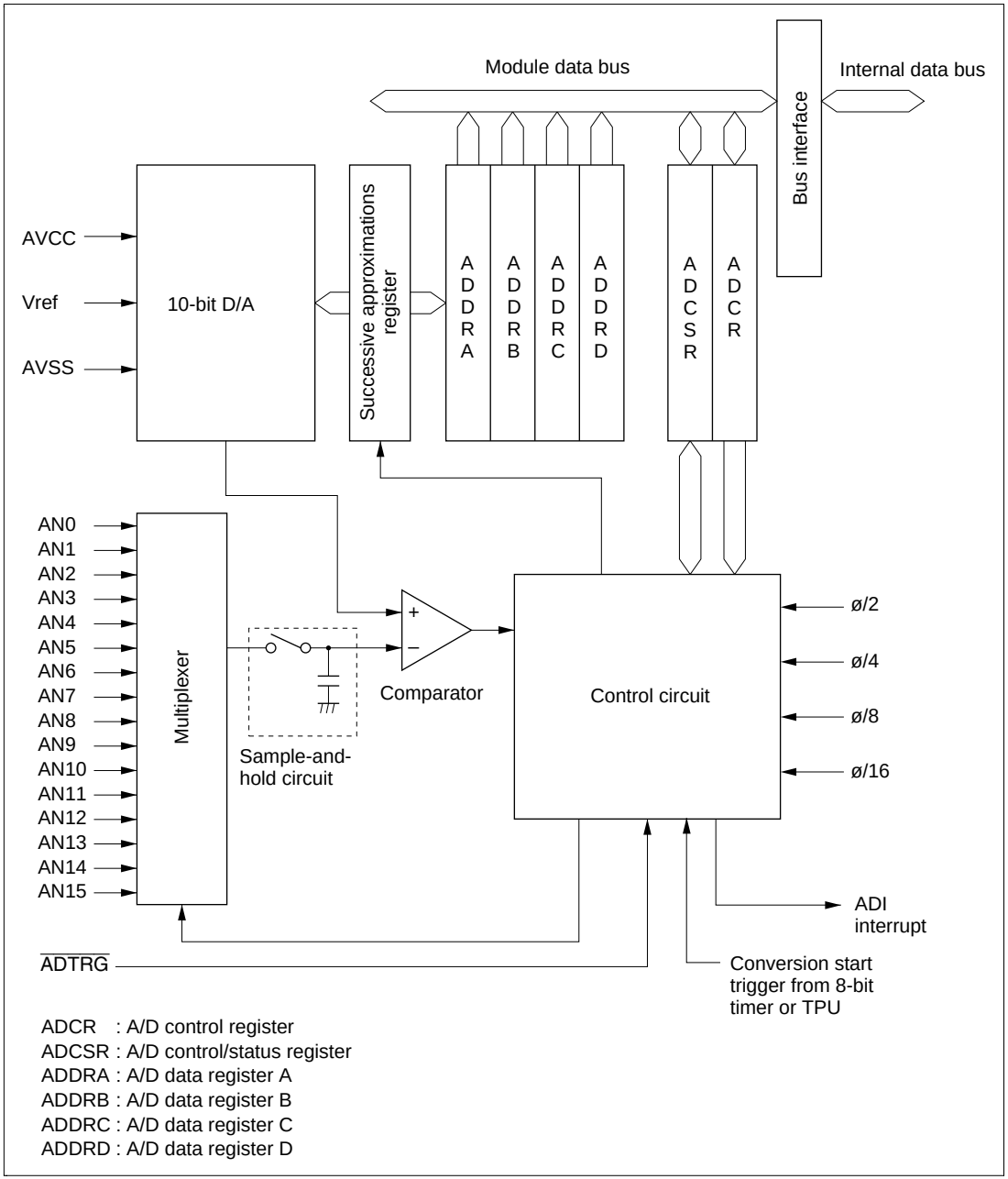


Figure 19-1 Block Diagram of A/D Converter

19.1.3 Pin Configuration

Table 19-1 summarizes the input pins used by the A/D converter.

The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the A/D conversion reference voltage pin.

The 16 analog input pins are divided into two channel sets and two groups, with analog input pins 0 to 7 (AN0 to AN7) comprising channel set 0, analog input pins 8 to 15 (AN8 to AN15) comprising channel set 1, analog input pins 0 to 3 and 8 to 11 (AN0 to AN3, AN8 to AN11) comprising group 0, and analog input pins 4 to 7 and 12 to 15 (AN4 to AN7, AN12 to AN15) comprising group 1.

Table 19-1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and reference voltage
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 0) group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Channel set 0 (CH3 = 0) group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 (CH3 = 1) group 0 analog inputs
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	Channel set 1 (CH3 = 1) group 1 analog inputs
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

19.1.4 Register Configuration

Table 19-2 summarizes the registers of the A/D converter.

Table 19-2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
A/D data register AH	ADDRAH	R	H'00	H'FF90
A/D data register AL	ADDRAL	R	H'00	H'FF91
A/D data register BH	ADDRBH	R	H'00	H'FF92
A/D data register BL	ADDRBL	R	H'00	H'FF93
A/D data register CH	ADDRCH	R	H'00	H'FF94
A/D data register CL	ADDRCL	R	H'00	H'FF95
A/D data register DH	ADDRDH	R	H'00	H'FF96
A/D data register DL	ADDRDL	R	H'00	H'FF97
A/D control/status register	ADCSR	R/(W)*[2]	H'00	H'FF98
A/D control register	ADCR	R/W	H'33	H'FF99
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.
*2 Bit 7 can only be written with 0 for flag clearing.

19.2 Register Descriptions

19.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the results of A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for the selected channel and stored there. The upper 8 bits of the converted data are transferred to the upper byte (bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 and 6) and stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown in table 19-3.

ADDR can always be read by the CPU. The upper byte can be read directly, but for the lower byte, data transfer is performed via a temporary register (TEMP). For details, see section 19.3, Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode or module stop mode.

Table 19-3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		
Group 0	Group 1	Group 0	Group 1	A/D Data Register
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

19.2.2 A/D Control/Status Register (ADCSR)

Bit	:	7	6	5	4	3	2	1	0
		ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to bit 7, to clear this flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCSR is initialized to H'00 by a reset, and in hardware standby mode or module stop mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description	
0	[Clearing conditions]	(Initial value)
	• When 0 is written to the ADF flag after reading ADF = 1	
	• When the DTC is activated by an ADI interrupt and ADDR is read	
1	[Setting conditions]	
	• Single mode: When A/D conversion ends	
	• Scan mode: When A/D conversion ends on all specified channels	

Bit 6—A/D Interrupt Enable (ADIE): Selects enabling or disabling of interrupt (ADI) requests at the end of A/D conversion.

Bit 6

ADIE	Description	
0	A/D conversion end interrupt (ADI) request disabled	(Initial value)
1	A/D conversion end interrupt (ADI) request enabled	

Bit 5—A/D Start (ADST): Selects starting or stopping on A/D conversion. Holds a value of 1 during A/D conversion.

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5

ADST	Description
0	• A/D conversion stopped (Initial value)
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion operating mode. See section 19.4, Operation, for single mode and scan mode operation. Only set the SCAN bit while conversion is stopped (ADST = 0).

Bit 4

SCAN	Description
0	Single mode (Initial value)
1	Scan mode

Bit 3—Channel Select 3 (CH3): Switches the analog input pins assigned to group 0 or group 1. Setting CH3 to 1 enables AN8 to AN15 to be used instead of AN0 to AN7.

Bit 3

CH3	Description
0	AN8 to AN11 are group 0 analog input pins, AN12 to AN15 are group 1 analog input pins
1	AN0 to AN3 are group 0 analog input pins, AN4 to AN7 are group 1 analog input pins (Initial value)

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, these bits select the analog input channels.

Only set the input channel while conversion is stopped (ADST = 0).

Channel Selection				Description	
CH3	**CH2**	**CH1**	**CH0**	**Single Mode (SCAN = 0)**	**Scan Mode (SCAN = 1)**
0	0	0	0	AN0 (Initial value)	AN0
			1	AN1	AN0, AN1
		1	0	AN2	AN0 to AN2
			1	AN3	AN0 to AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4 to AN6
			1	AN7	AN4 to AN7
1	0	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8 to AN10
			1	AN11	AN8 to AN11
	1	0	0	AN12	AN12
			1	AN13	AN12, AN13
		1	0	AN14	AN12 to AN14
			1	AN15	AN12 to AN15

19.2.3 A/D Control Register (ADCR)

Bit	:	7	6	5	4	3	2	1	0
		TRGS1	TRGS0	—	—	CKS1	CKS0	—	—
Initial value	:	0	0	1	1	0	0	1	1
R/W	:	R/W	R/W	—	—	R/W	R/W	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations and sets the A/D conversion time.

ADCR is initialized to H'33 by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): Select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped (ADST = 0).

Bit 7	Bit 6		
TRGS1	**TRGS0**	**Description**	
0	0	A/D conversion start by software is enabled	(Initial value)
	1	A/D conversion start by TPU conversion start trigger is enabled	
1	0	A/D conversion start by 8-bit timer conversion start trigger is enabled	
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled	

Bits 5, 4, 1, and 0—Reserved: They are always read as 1 and cannot be modified.

Bits 3 and 2—Clock Select 1 and 0 (CKS1, CKS0): These bits select the A/D conversion time. The conversion time should be changed only when ADST = 0.

Set bits CKS1 and CKS0 to give a conversion time of at least 10 µs when $AV_{CC} \geq 4.5$ V, and at least 16 µs when $AV_{CC} < 4.5$ V.

Bit 3	Bit 2		
CKS1	**CKS0**	**Description**	
0	0	Conversion time = 530 states (max.)	(Initial value)
	1	Conversion time = 266 states (max.)	
1	0	Conversion time = 134 states (max.)	
	1	Conversion time = 68 states (max.)	

19.2.4 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCR is a 8-bit readable/writable register that performs module stop mode control.

When the MSTPA1 bit in MSTPCR is set to 1, A/D converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 24.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 1—Module Stop (MSTPA1): Specifies the A/D converter module stop mode.

Bit 1

MSTPA1	Description	
0	A/D converter module stop mode cleared	
1	A/D converter module stop mode set	(Initial value)

19.3 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers, and the data bus to the bus master is 8 bits wide. Therefore, in accesses by the bus master, the upper byte is accessed directly, but the lower byte is accessed via a temporary register (TEMP).

A data read from ADDR is performed as follows. When the upper byte is read, the upper byte value is transferred to the CPU and the lower byte value is transferred to TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading ADDR. always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 19-2 shows the data flow for ADDR access.

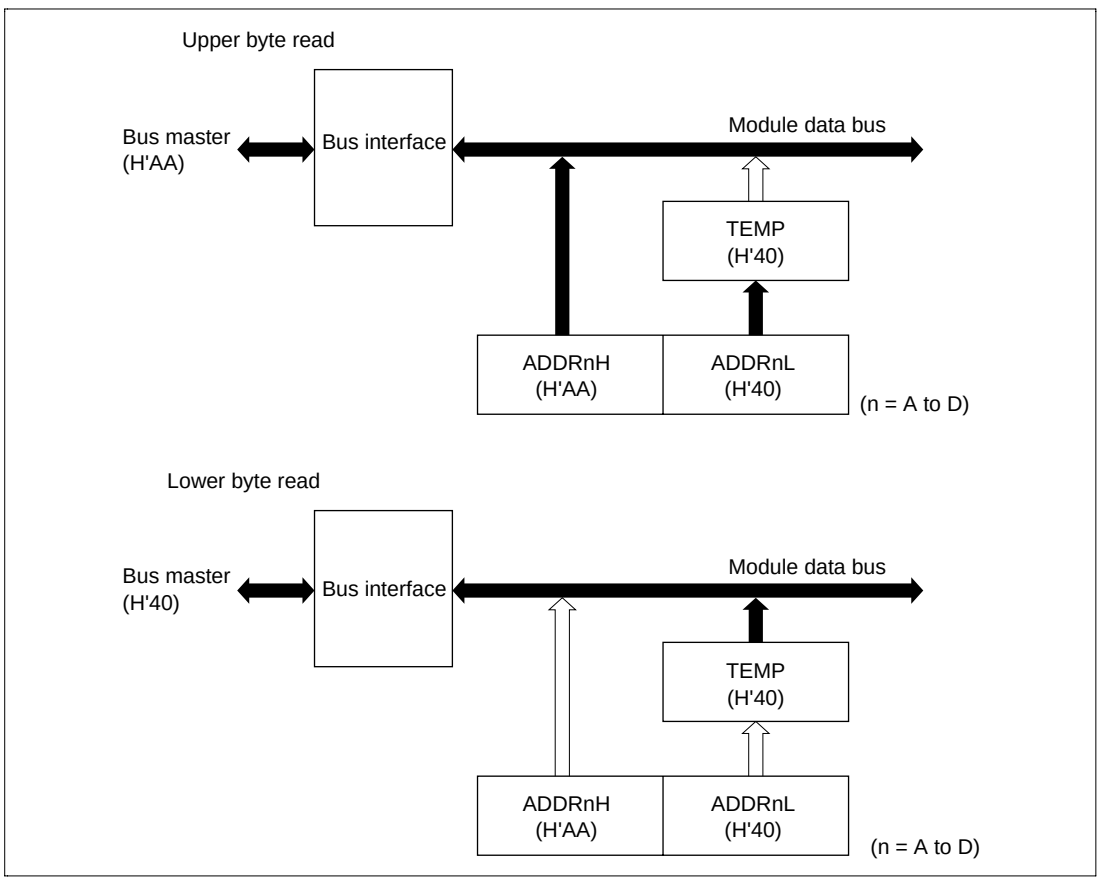


Figure 19-2 ADDR Access Operation (Reading H'AA40)

19.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode.

19.4.1 Single Mode (SCAN = 0)

Single mode is selected when A/D conversion is to be performed on a single channel only. A/D conversion is started when the ADST bit is set to 1, according to the software or external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading ADCSR.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next. Figure 19-3 shows a timing diagram for this example.

[1] Single mode is selected (SCAN = 0), input channel AN1 is selected (CH3 = 0, CH2 = 0, CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).

[2] When A/D conversion is completed, the result is transferred to ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.

[3] Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.

[4] The A/D interrupt handling routine starts.

[5] The routine reads ADCSR, then writes 0 to the ADF flag.

[6] The routine reads and processes the connection result (ADDRB).

[7] Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps [2] to [7] are repeated.

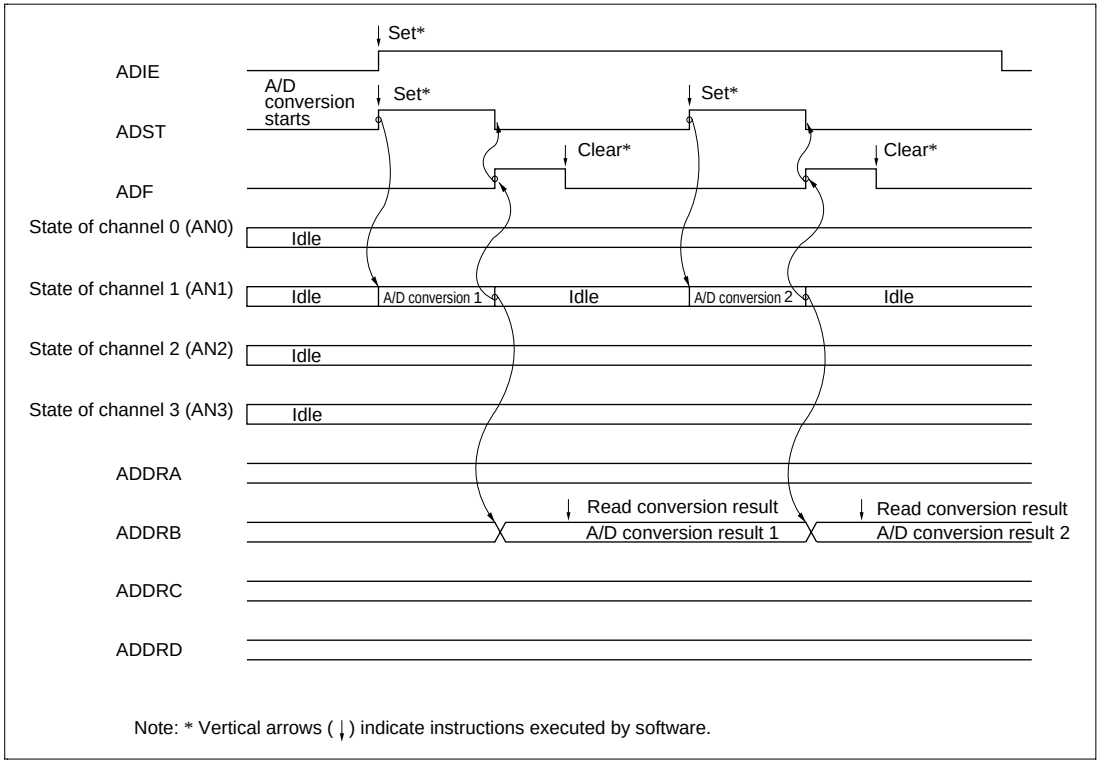


Figure 19-3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

19.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by a software, timer or external trigger input, A/D conversion starts on the first channel in the group (AN0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the ADDR registers corresponding to the channels.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again from the first channel (AN0). The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 19-4 shows a timing diagram for this example.

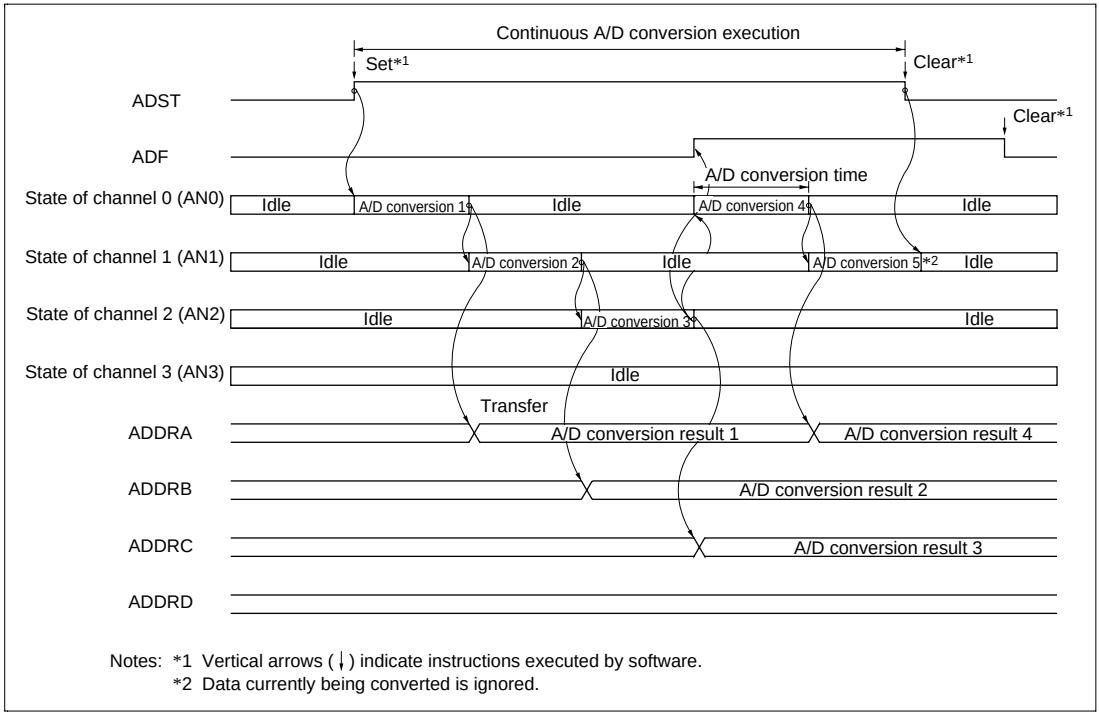
[1] Scan mode is selected (SCAN = 1), channel set 0 is selected (CH3 = 0), scan group 0 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1)

[2] When A/D conversion of the first channel (AN0) is completed, the result is transferred to ADDRA. Next, conversion of the second channel (AN1) starts automatically.

[3] Conversion proceeds in the same way through the third channel (AN2).

[4] When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends.

[5] Steps [2] to [4] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Notes: *1 Vertical arrows (↓) indicate instructions executed by software.
*2 Data currently being converted is ignored.

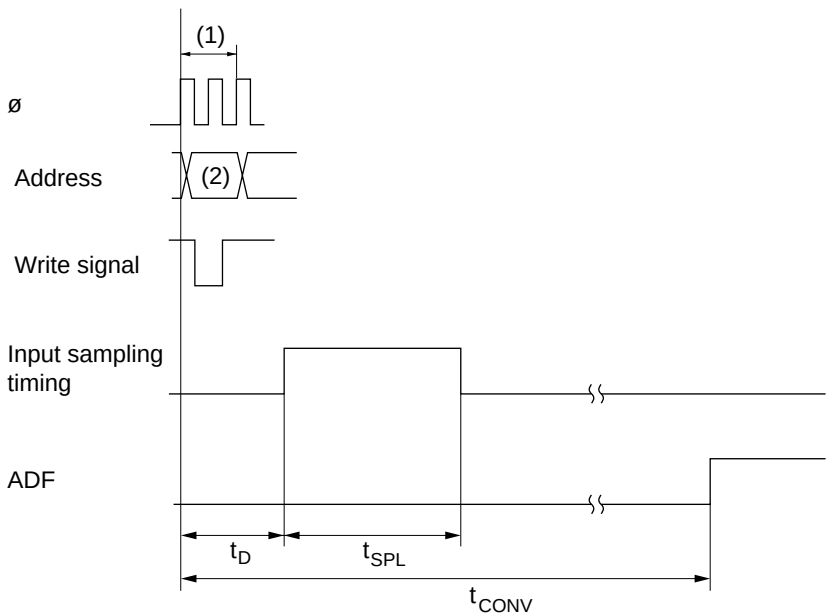
Figure 19-4 Example of A/D Converter Operation (Scan Mode, 3 Channels AN0 to AN2 Selected)

19.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 19-5 shows the A/D conversion timing. Table 19-4 indicates the A/D conversion time.

As indicated in figure 19-5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 19-4.

In scan mode, the values given in table 19-4 apply to the first conversion time. The values given in table 19-5 apply to the second and subsequent conversions. In both cases, set bits CKS1 and CKS0 in ADCR to give a conversion time of at least 10 µs when $AV_{CC} \geq 4.5$ V, and at least 16 µs when $AV_{CC} < 4.5$ V.



Legend
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 19-5 A/D Conversion Timing

Table 19-4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: Values in the table are the number of states.

Table 19-5 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

19.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit has been set to 1 by software. Figure 19-6 shows the timing.

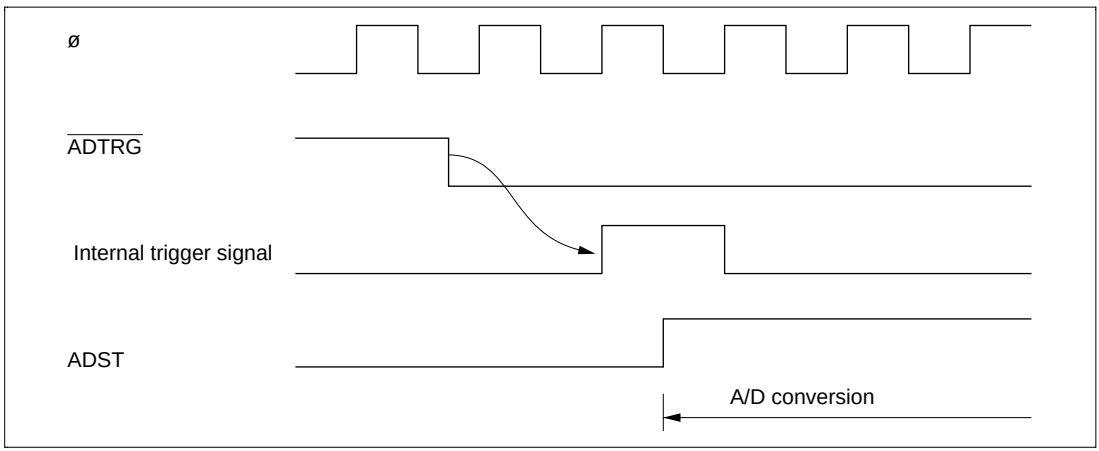


Figure 19-6 External Trigger Input Timing

19.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. ADI interrupt requests can be enabled or disabled by means of the ADIE bit in ADCSR.

The DTC and DMAC can be activated by an ADI interrupt. Having the converted data read by the DTC or DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

The A/D converter interrupt source is shown in table 19-6.

Table 19-6 A/D Converter Interrupt Source

Interrupt Source	Description	DTC, DMAC Activation
ADI	Interrupt due to end of conversion	Possible

19.6 Usage Notes

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins:

(1) Analog input voltage range

The voltage applied to analog input pin ANn during A/D conversion should be in the range AVSS ≤ ANn ≤ Vref.

(2) Relation between AVCC, AVSS and VCC, VSS

As the relationship between AVCC, AVSS and VCC, VSS, set AVSS = VSS. If the A/D converter is not used, the AVCC and AVSS pins must on no account be left open.

(3) Vref input range

The analog reference voltage input at the Vref pin set in the range Vref ≤ AVCC.

If conditions (1), (2), and (3) above are not met, the reliability of the device may be adversely affected.

Notes on Board Design: In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN15), analog reference power supply (Vref), and analog power supply (AVCC) by the analog ground (AVSS). Also, the analog ground (AVSS) should be connected at one point to a stable digital ground (VSS) on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN15) and analog reference power supply (Vref) should be connected between AVCC and AVSS as shown in figure 19-7.

Also, the bypass capacitors connected to AVCC and Vref and the filter capacitor connected to AN0 to AN15 must be connected to AVSS.

If a filter capacitor is connected as shown in figure 19-7, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

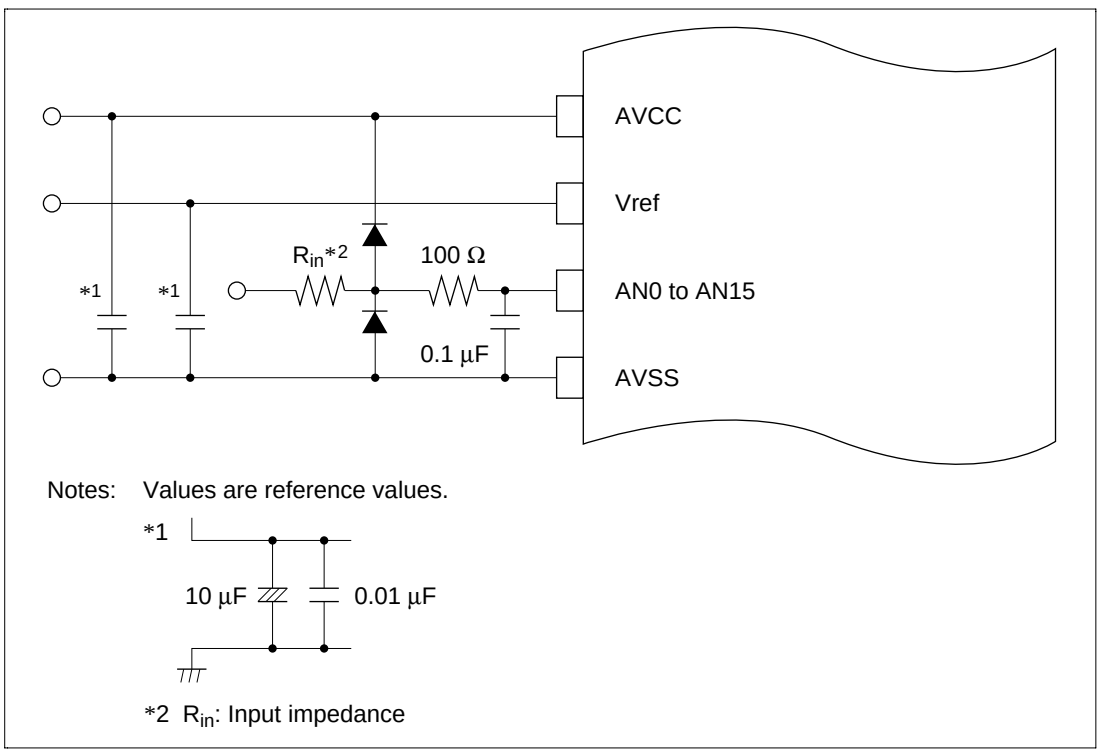
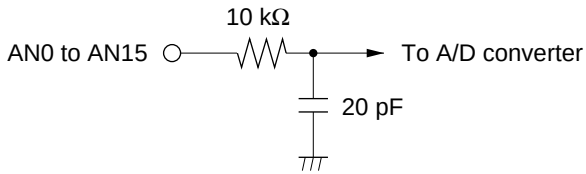


Figure 19-7 Example of Analog Input Protection Circuit

Table 19-7 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ



Note: Values are reference values.

Figure 19-8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: H8S/2633 Series A/D conversion precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'00) to B'0000000001 (H'01) (see figure 19-10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3E) to B'1111111111 (H'3F) (see figure 19-10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 19-9).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

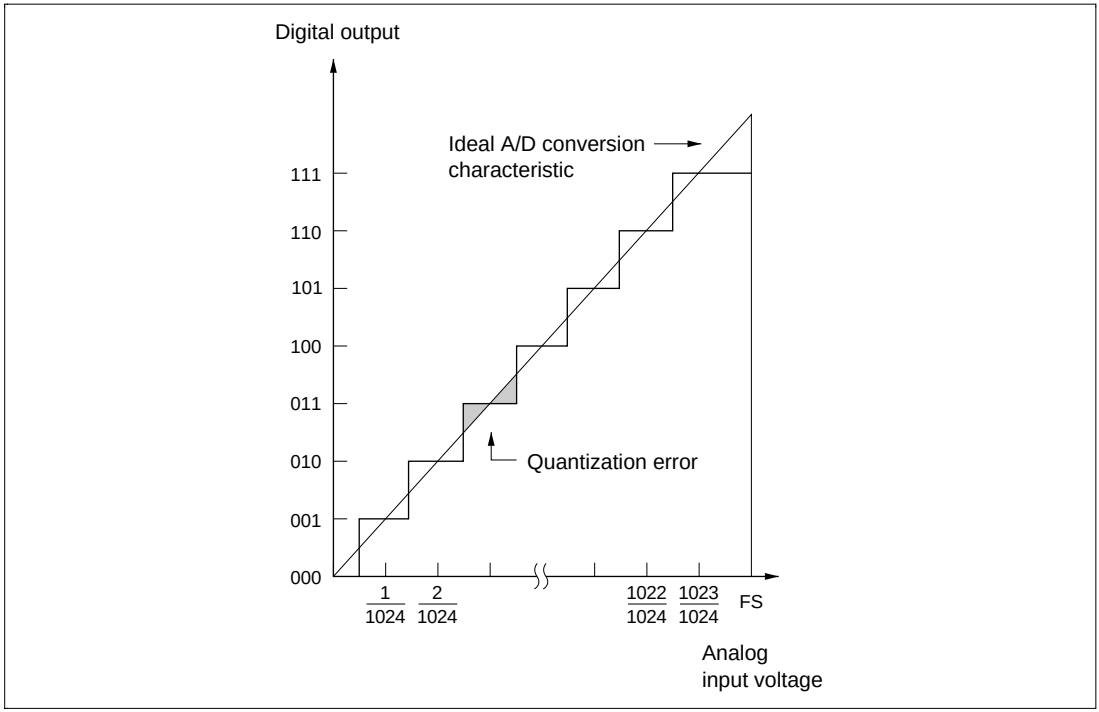


Figure 19-9 A/D Conversion Precision Definitions (1)

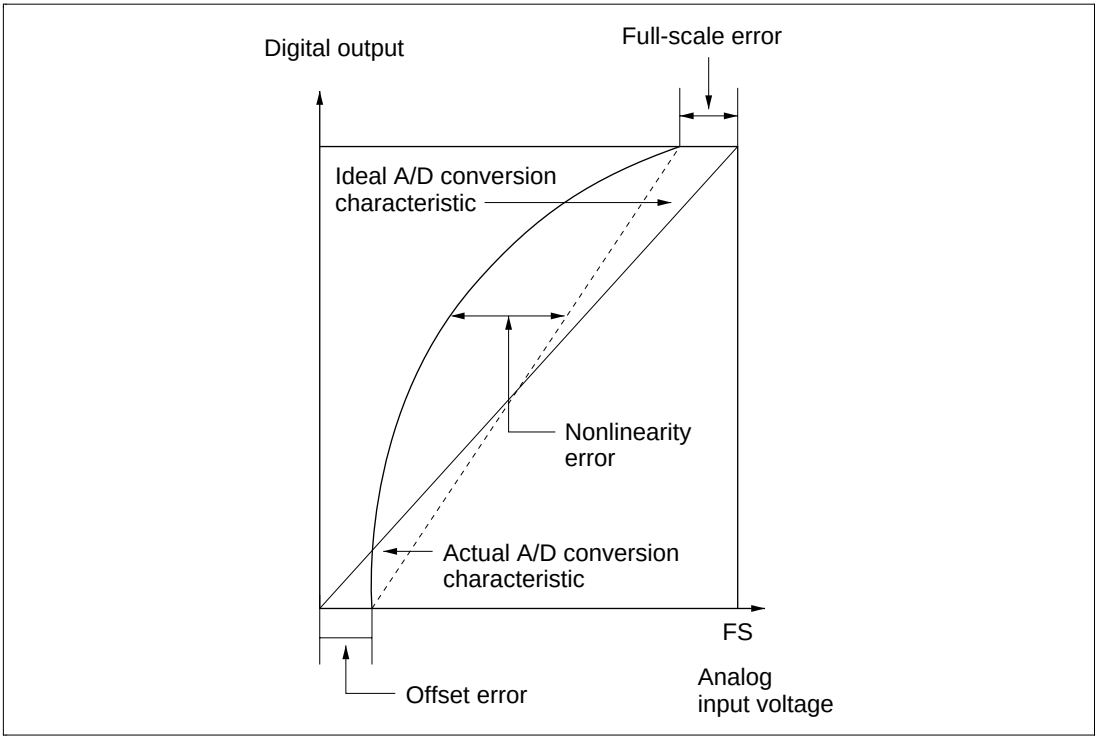


Figure 19-10 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance: H8S/2633 Series analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

However, if a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored.

However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

Influences on Absolute Precision: Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

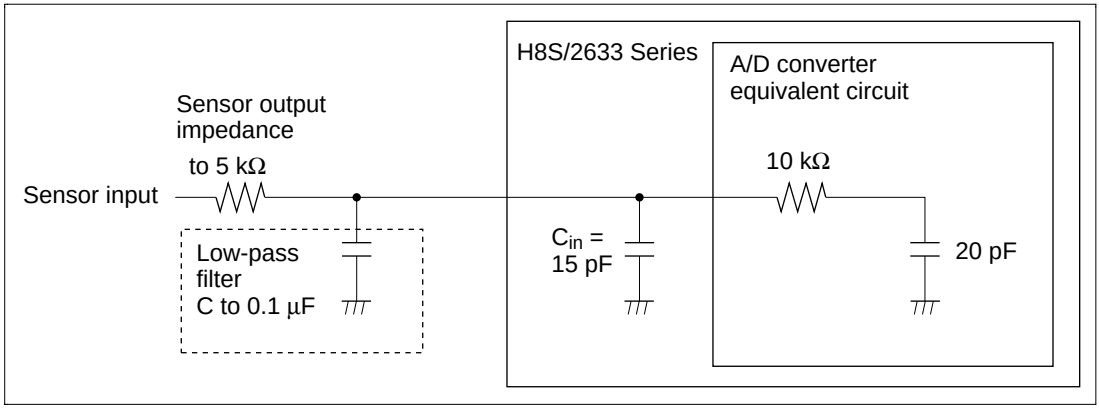


Figure 19-11 Example of Analog Input Circuit

Section 20 D/A Converter

20.1 Overview

The H8S/2633 Series has an on-chip D/A converter module with four channels.

20.1.1 Features

Features of the D/A converter module are listed below.

- Eight-bit resolution
- Four-channel output
- Maximum conversion time: 10 µs (with 20-pF load capacitance)
- Output voltage: 0 V to Vref
- D/A output retention in software standby mode
- Possible to set module stop mode

  Operation of D/A converter is disenabled by initial values. It is possible to access the register by canceling module stop mode.

20.1.2 Block Diagram

Figure 20-1 shows a block diagram of the D/A converter.

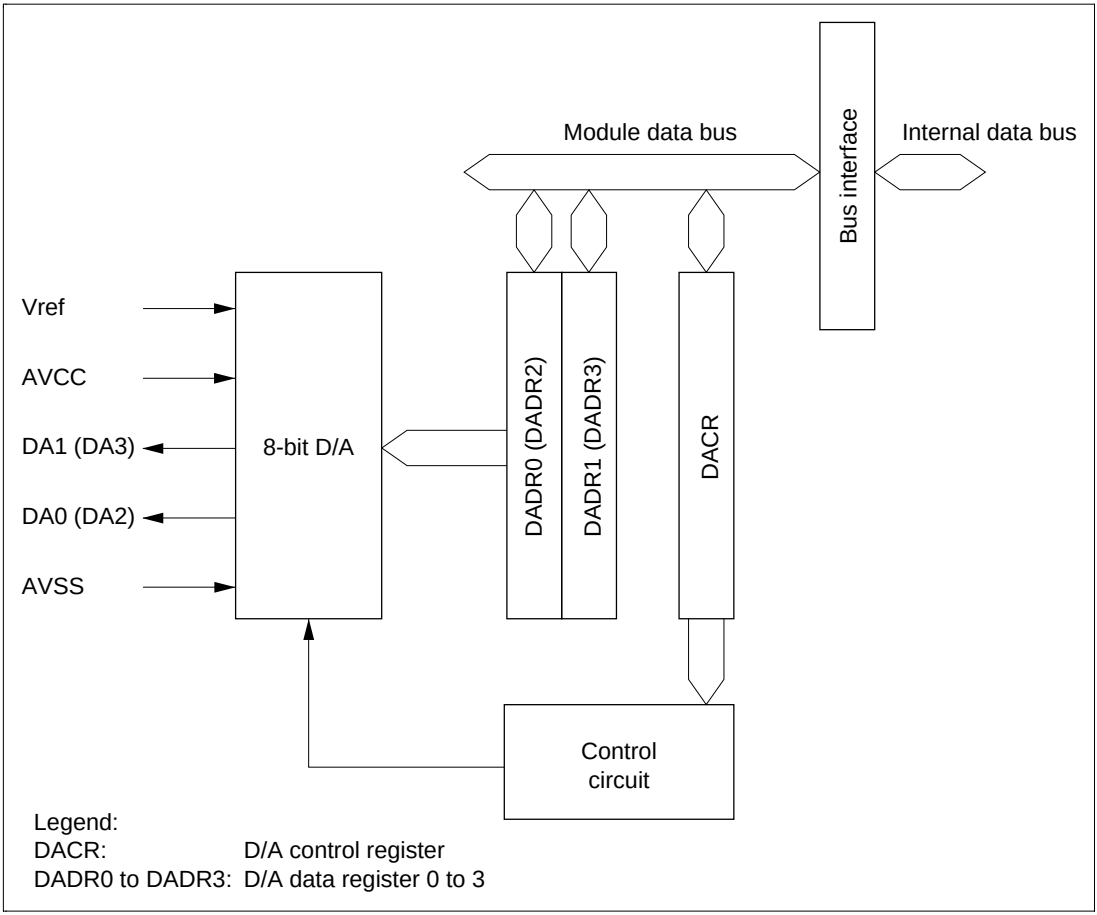


Figure 20-1 Block Diagram of D/A Converter

20.1.3 Input and Output Pins

Table 20-1 lists the input and output pins used by the D/A converter module.

Table 20-1 Input and Output Pins of D/A Converter Module

Name	Abbreviation	I/O	Function
Analog supply voltage	AVCC	Input	Power supply for analog circuits
Analog ground	AVSS	Input	Ground and reference voltage for analog circuits
Analog output 0	DA0	Output	Analog output channel 0
Analog output 1	DA1	Output	Analog output channel 1
Analog output 2	DA2	Output	Analog output channel 2
Analog output 3	DA3	Output	Analog output channel 3
Reference voltage	Vref	Input	Reference voltage of analog section

20.1.4 Register Configuration

Table 20-2 lists the registers of the D/A converter module.

Table 20-2 D/A Converter Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*
0, 1	D/A data register 0	DADR0	R/W	H'00	H'FFA4
	D/A data register 1	DADR1	R/W	H'00	H'FFA5
	D/A control register 01	DACR01	R/W	H'1F	H'FFA6
2, 3	D/A data register 2	DADR2	R/W	H'00	H'FDAC
	D/A data register 3	DADR3	R/W	H'00	H'FDAD
	D/A control register 23	DACR23	R/W	H'1F	H'FDAE
All	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDF8
	Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA

Note: * Lower 16 bits of the address.

20.2 Register Descriptions

20.2.1 D/A Data Registers 0 to 3 (DADR0 to DADR3)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A data registers 0 to 3 (DADR0 to DADR3) are 8-bit readable/writable registers that store data to be converted. When analog output is enabled, the value in the D/A data register is converted and output continuously at the analog output pin.

The D/A data registers are initialized to H'00 by a reset and in hardware standby mode.

20.2.2 D/A Control Register 01 and 23 (DACR01 and DACR23)

Bit	:	7	6	5	4	3	2	1	0
		DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—	—

DACR01 and DACR23 are an 8-bit readable/writable register that controls the operation of the D/A converter module.

DACR01 and DACR23 are initialized to H'1F by a reset and in hardware standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7 DAOE1	Description
0	Analog output DA1 (DA3) is disabled (Initial value)
1	D/A conversion is enabled on channel 1. Analog output DA1 (DA3) is enabled

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6 DAOE0	Description	
0	Analog output DA0 (DA2) is disabled	(Initial value)
1	D/A conversion is enabled on channel 0. Analog output DA0 (DA2) is enabled	

Bit 5—D/A Enable (DAE): Controls D/A conversion, in combination with bits DAOE0 and DAOE1. D/A conversion is controlled independently on channels 0 and 1 when DAE = 0. Channels 0 and 1 are controlled together when DAE = 1.

Output of the converted results is always controlled independently by DAOE0 and DAOE1.

Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	D/A conversion
0	0	*	Disabled on channels 0 and 1 (channels 2 and 3)
	1	0	Enabled on channel 0 (channel 2)
			Disabled on channel 1 (channel 3)
		1	Enabled on channels 0 and 1 (channels 2 and 3)
1	0	0	Disabled on channel 0 (channel 2)
			Enabled on channel 1 (channel 3)
		1	Enabled on channels 0 and 1 (channels 2 and 3)
	1	*	Enabled on channels 0 and 1 (channels 2 and 3)

*: Don't care

If the H8S/2633 Series chip enters software standby mode while D/A conversion is enabled, the D/A output is retained and the analog power supply current is the same as during D/A conversion. If it is necessary to reduce the analog power supply current in software standby mode, disable D/A output by clearing both the DAOE0 and DAOE1 bits to 0.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

20.2.3 Module Stop Control Register A and C (MSTPCRA and MSTPCRC)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	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

MSTPCRA and MSTPCRC are an 8-bit readable/writable registers that performs module stop mode control. When the MSTPA2 and MSTPC5 are set to 1, the D/A converter halts and enters module stop mode at the end of the bus cycle. Register read/write is disenabled in module stop mode. See section 24.5, Module Stop Mode, for details.

MSTPCRA is initialized to H'3F by a power-on reset and in hardware standby mode. MSTPCRC is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Module Stop Control Register A (MSTPCRA)

Bit 2—Module Stop (MSTPA2): Specifies D/A converter (channels 0 and 1) module stop mode.

Bit 2		
MSTPA2	Description	
0	D/A converter (channels 0 and 1) module stop mode is cleared	
1	D/A converter (channels 0 and 1) module stop mode is set	(Initial value)

Module Stop Control Register C (MSTPCRC)

Bit 5—Module Stop (MSTPC5): Specifies D/A converter (channels 2 and 3) module stop mode.

Bit 5		
MSTPC5	Description	
0	D/A converter (channels 2 and 3) module stop mode is cleared	
1	D/A converter (channels 2 and 3) module stop mode is set	(Initial value)

20.3 Operation

The D/A converter module has two built-in D/A converter circuits that can operate independently.

D/A conversion is performed continuously whenever enabled by the D/A control register (DACR). When a new value is written in DADR0 or DADR1, conversion of the new value begins immediately. The converted result is output by setting the DAOE0 or DAOE1 bit to 1.

An example of conversion on channel 0 is given next. Figure 20-2 shows the timing.

- Software writes the data to be converted in DADR0.
- D/A conversion begins when the DAOE0 bit in DACR is set to 1. After the elapse of the conversion time, analog output appears at the DA0 pin. The output value is Vref × (DADR0 value)/256.

  This output continues until a new value is written in DADR0 or the DAOE0 bit is cleared to 0.
- If a new value is written in DADR0, conversion begins immediately. Output of the converted result begins after the conversion time.
- When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

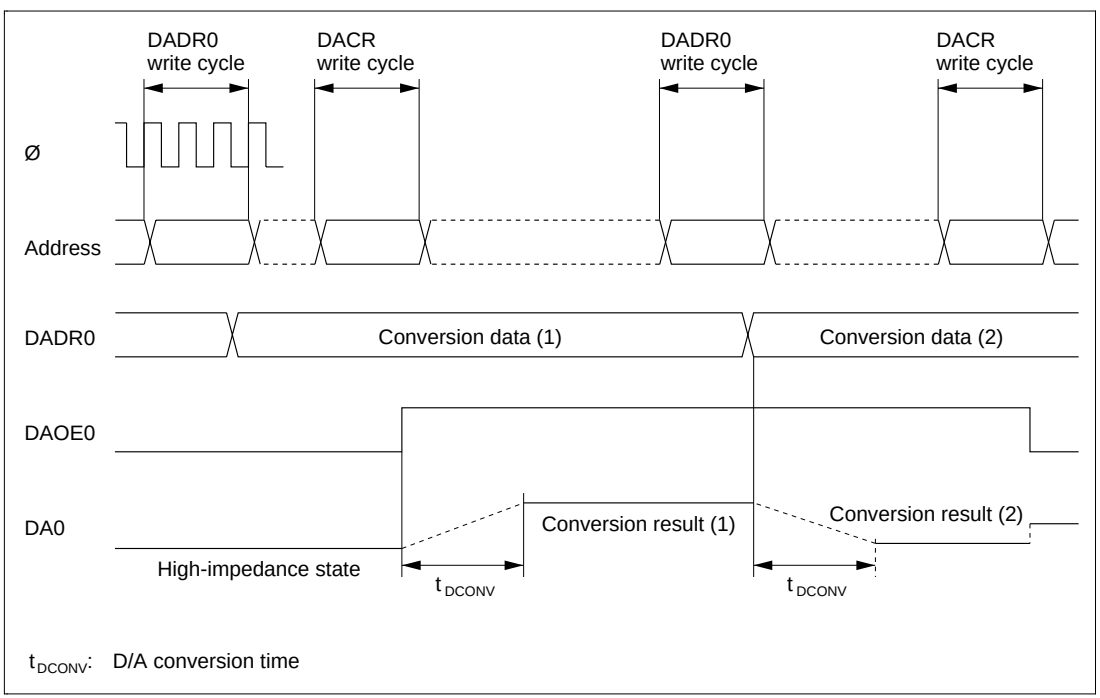


Figure 20-2 D/A Conversion (Example)

Section 21 RAM

21.1 Overview

The H8S/2633 has 16 kbytes of on-chip high-speed static RAM, the H8S/2632 has 12 kbytes, and the H8S/2631 has 8 kbytes. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR).

21.1.1 Block Diagram

Figure 21-1 shows a block diagram of the on-chip RAM.

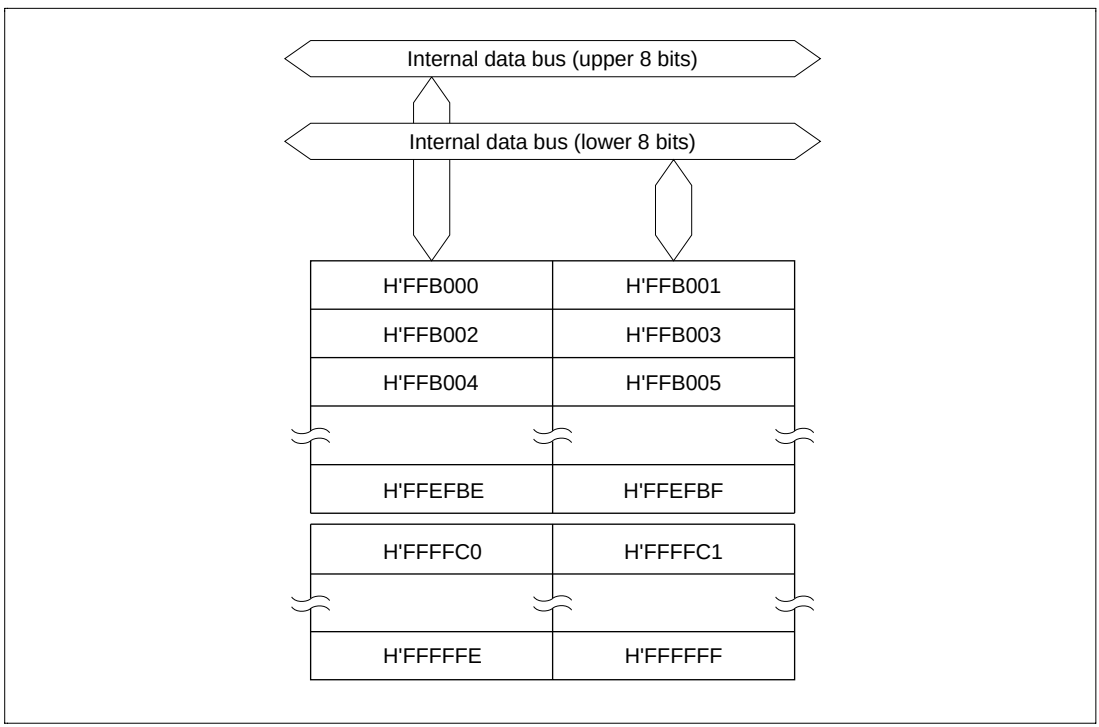


Figure 21-1 Block Diagram of RAM (H8S/2633 Series)

21.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 21-1 shows the address and initial value of SYSCR.

Table 21-1 RAM Register

Name	Abbreviation	R/W	Initial Value	Address*
System control register	SYSCR	R/W	H'01	H'FDE5

Note: * Lower 16 bits of the address.

21.2 Register Descriptions

21.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of other bits in SYSCR, see section 3.2.2, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset state is released. It is not initialized in software standby mode.

Note: When the DTC is used, the RAME bit must not be cleared to 0.

Bit 0

RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

21.3 Operation

When the RAME bit is set to 1, accesses to addresses H'FFB000 to H'FFEFBF and H'FFFFC0 to H'FFFFFF in the H8S/2633, to addresses H'FFC000 to H'FFEFBF and H'FFFFC0 to H'FFFFFF in the H8S/2632, and to addresses H'FFD000 to H'FFEFBF and H'FFFFC0 to H'FFFFFF in the H8S/2631, are directed to the on-chip RAM. When the RAME bit is cleared to 0, the off-chip address space is accessed.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written to and read in byte or word units. Each type of access can be performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

21.4 Usage Notes

When Using the DTC: DTC register information can be located in addresses H'FFEBC0 to H'FFEFBF. When the DTC is used, the RAME bit must not be cleared to 0.

Reserved Areas: Addresses H'FFB000 to H'FFBFFF in the H8S/2632, and H'FFB000 to H'FFCFFF in the H8S/2631 are reserved areas that cannot be read or written to. When the RAME bit is cleared to 0, the off-chip address space is accessed.

22.1 Overview

The H8/2633 Series has 256 kbytes of on-chip flash memory, or 256 , 192 , or 128 kbytes of on-chip mask ROM. The ROM is connected to the bus master via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching is thus speeded up, and processing speed increased.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2 to MD0).

The flash memory version can be erased and programmed on-board, as well as with a special-purpose PROM programmer.

22.1.1 Block Diagram

Figure 22-1 shows a block diagram of 256-kbyte ROM.

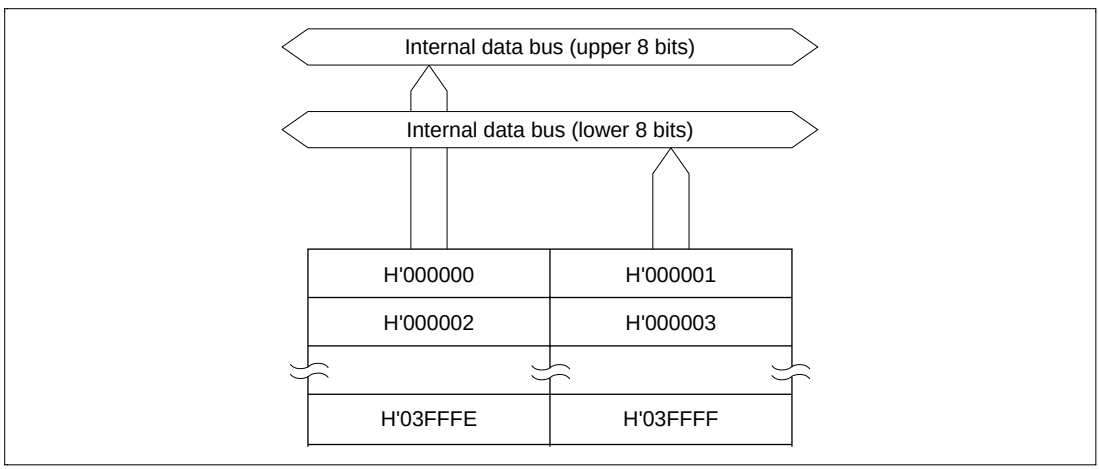


Figure 22-1 Block Diagram of ROM (256 kbytes)

22.1.2 Register Configuration

The H8/2633 Series operating mode is controlled by the mode pins and the BCRL register. The register configuration is shown in table 22-1.

Table 22-1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined	H'FDE7

Note: * Lower 16 bits of the address.

22.2 Register Descriptions

22.2.1 Mode Control Register (MDCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value:	1	0	0	0	0	—*	—*	—*
R/W:	R/W	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register used to monitor the current H8/2633 Series operating mode.

Bit 7—Reserved: Only 1 should be written to this bit.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0. MDS2 to MDS0 are read-only bits, and cannot be modified. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a power-on reset, but are retained in a manual reset.

22.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data can be accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2, MD1, and MD0). These settings are shown in table 22-2.

Table 22-2 Operating Modes and ROM (F-ZTAT Version)

		Mode Pins				
	Operating Mode	**FWE**	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	0	—
Mode 1					1	
Mode 2				1	0	
Mode 3					1	
Mode 4	Advanced expanded mode with on-chip ROM disabled		1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled				1	
Mode 6	Advanced expanded mode with on-chip ROM enabled			1	0	Enabled (256 kbytes)
Mode 7	Advanced single-chip mode				1	Enabled (256 kbytes)
Mode 8	—	1	0	0	0	—
Mode 9					1	
Mode 10	Boot mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (256 kbytes)
Mode 11	Boot mode (advanced single-chip mode)*2				1	Enabled (256 kbytes)
Mode 12	—		1	0	0	—
Mode 13					1	
Mode 14	User program mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (256 kbytes)
Mode 15	User program mode (advanced single-chip mode)*2				1	Enabled (256 kbytes)

Notes: *1 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced expanded mode with on-chip ROM enabled.

*2 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced single-chip mode.

Table 22-3 Operating Modes and ROM (Mask ROM Version)

		Mode Pins			
	Operating Mode	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	—
Mode 1				1	
Mode 2			1	0	
Mode 3				1	
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1	
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	Enabled (256 kbytes)*
Mode 7	Advanced single-chip mode			1	Enabled (256 kbytes)*

Note: * In the case of the H8S/2633. 192 kbytes are enabled in the H8S/2632, and 128 kbytes in the H8S/2631.

22.4 Flash Memory Overview

22.4.1 Features

The H8S/2633 Series has 256 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - — Program mode
  - — Erase mode
  - — Program-verify mode
  - — Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block units) can be performed. To erase the entire flash memory, each block must be erased in turn. Block erasing can be performed as required on 4 kbytes, 32 kbytes, and 64 kbytes blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte programming, equivalent to 78 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - — Boot mode
  - — User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

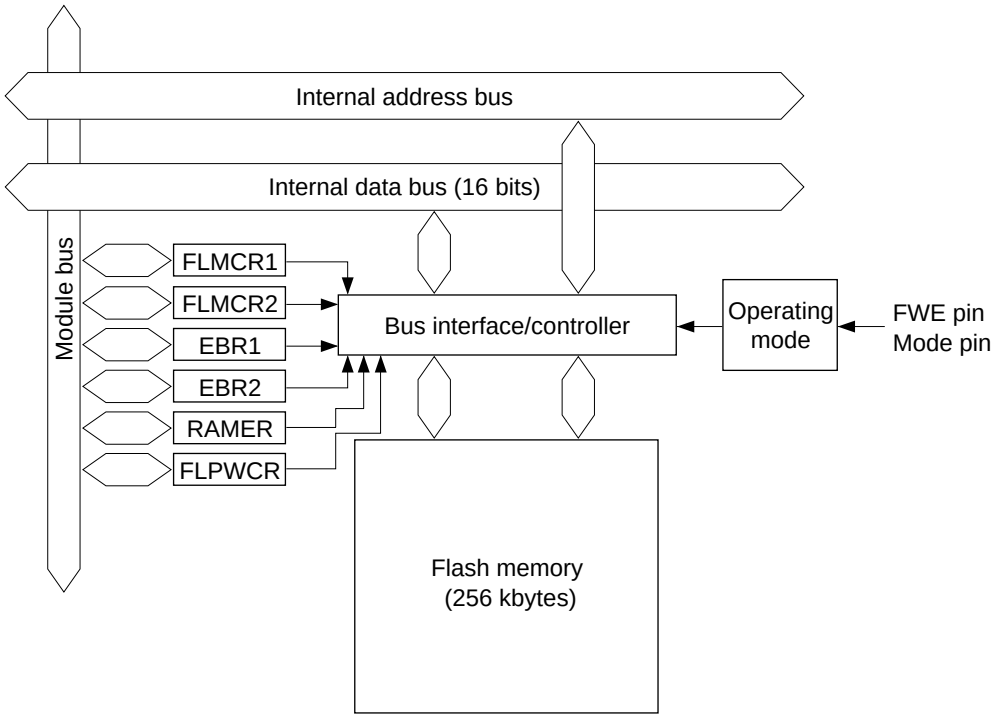
  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are three protect modes, hardware, software, and error protection, which allow protected status to be designated for flash memory program/erase/verify operations.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

22.4.2 Overview

Block Diagram



Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

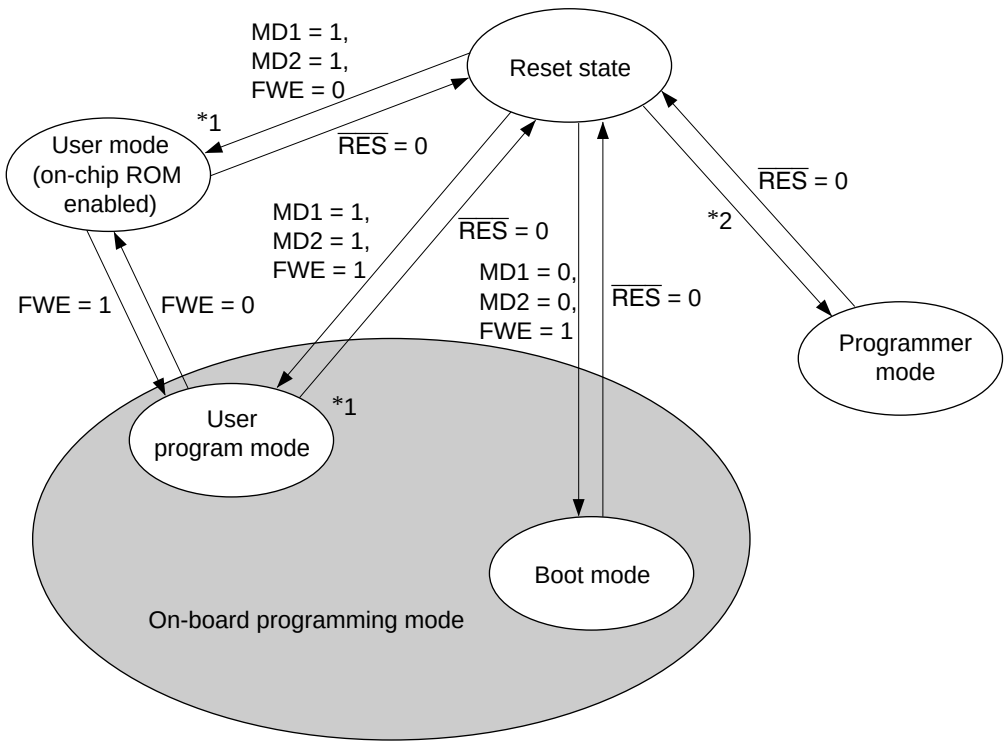
Figure 22-2 Block Diagram of Flash Memory

22.4.3 Flash Memory Operating Modes

Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the microcomputer enters an operating mode as shown in figure 22-3. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

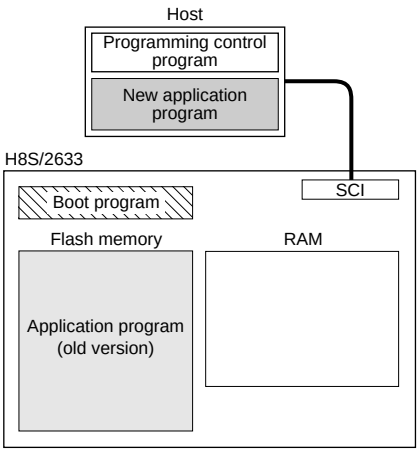
*1 RAM emulation possible
*2 MD0 = 0, MD1 = 0, MD2 = 0, P14 = 0, P16 = 0, PF0 = 1

Figure 22-3 Flash Memory State Transitions

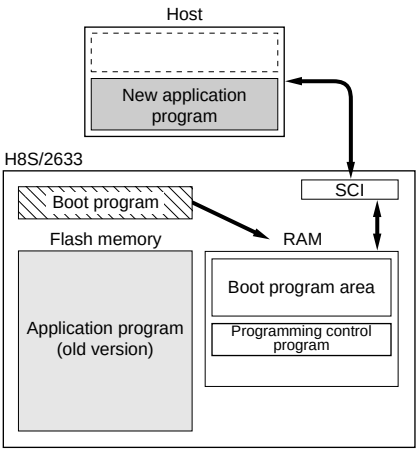
22.4.4 On-Board Programming Modes

Boot Mode

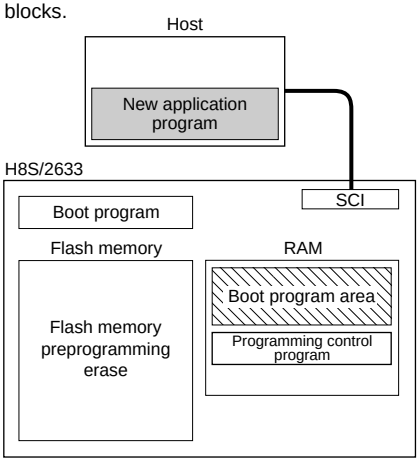
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



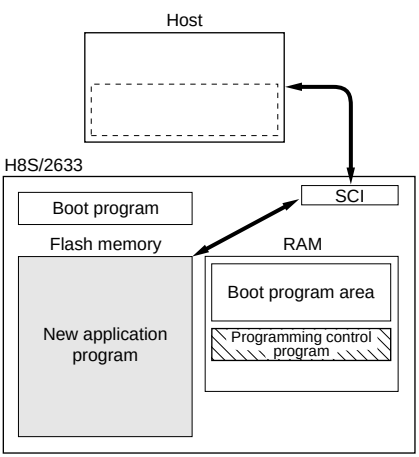
2. Programming control program transfer
   When boot mode is entered, the boot program in the H8S/2633 (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

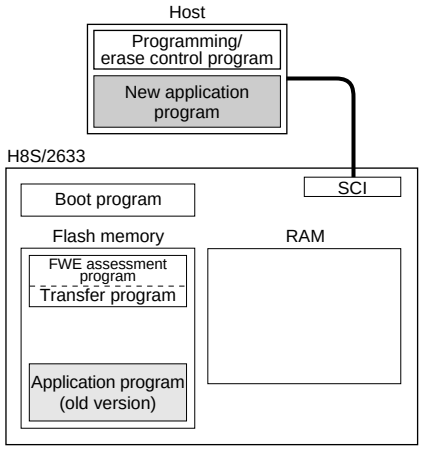


4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

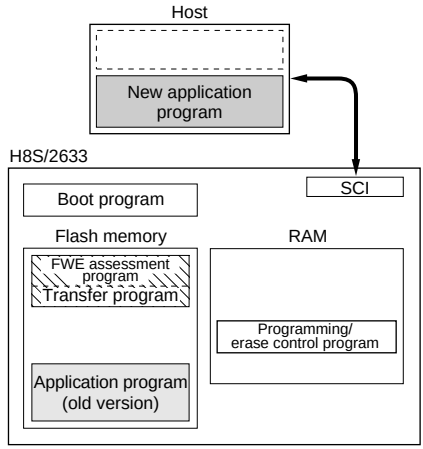


Program execution state

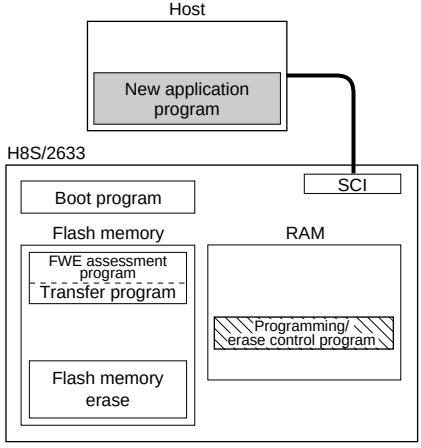
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



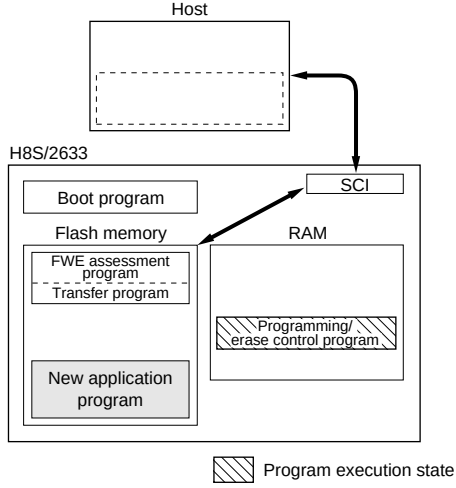
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



22.4.5 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

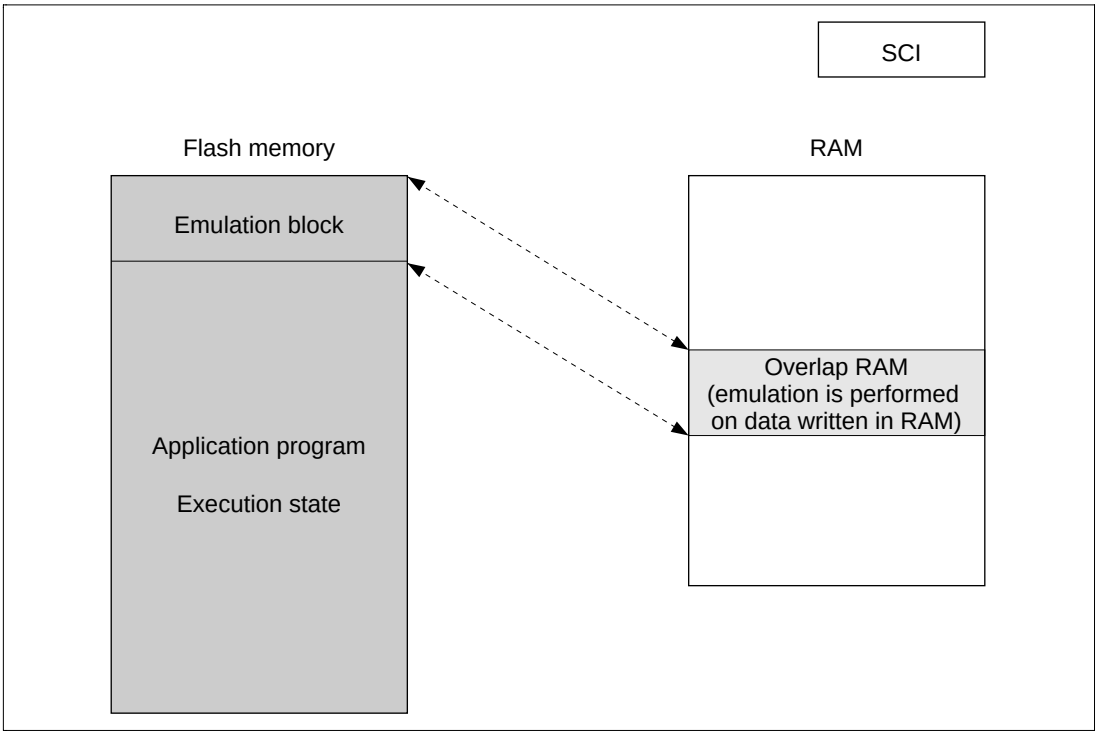


Figure 22-4 Reading Overlap RAM Data in User Mode or User Program Mode

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

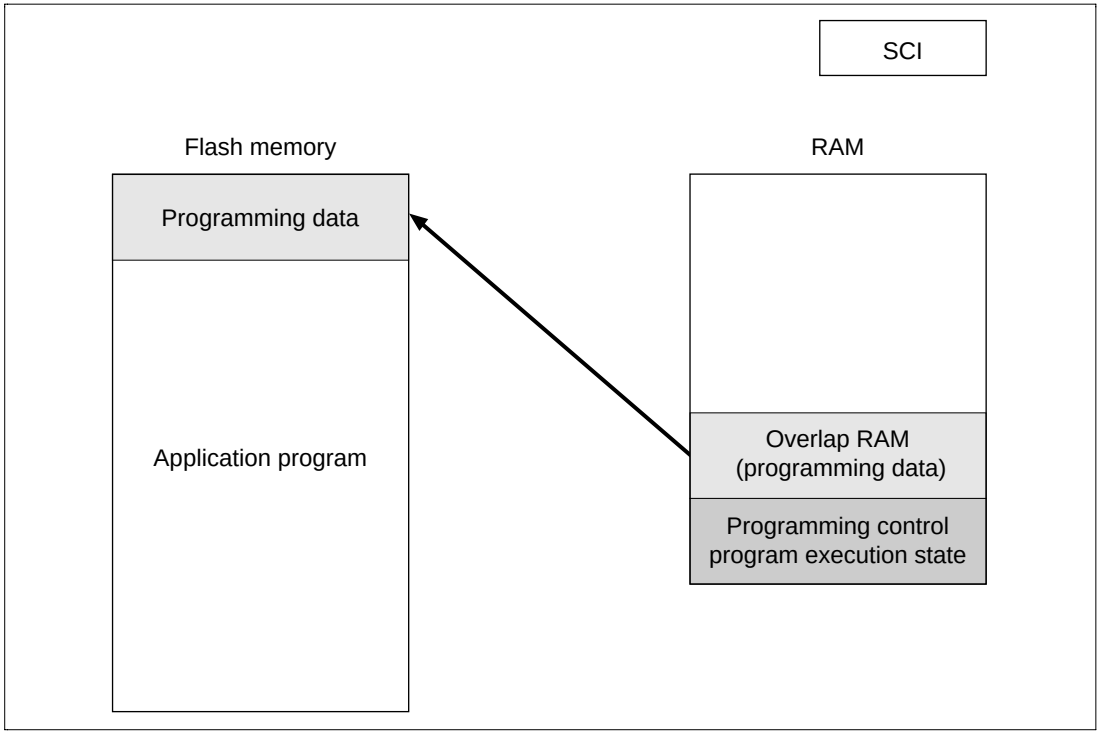


Figure 22-5 Writing Overlap RAM Data in User Program Mode

22.4.6 Differences between Boot Mode and User Program Mode

Table 22-4 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify Program/program-verify Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

22.4.7 Block Configuration

The flash memory is divided into three 64 kbytes blocks, one 32 kbytes block, and eight 4 kbytes blocks.

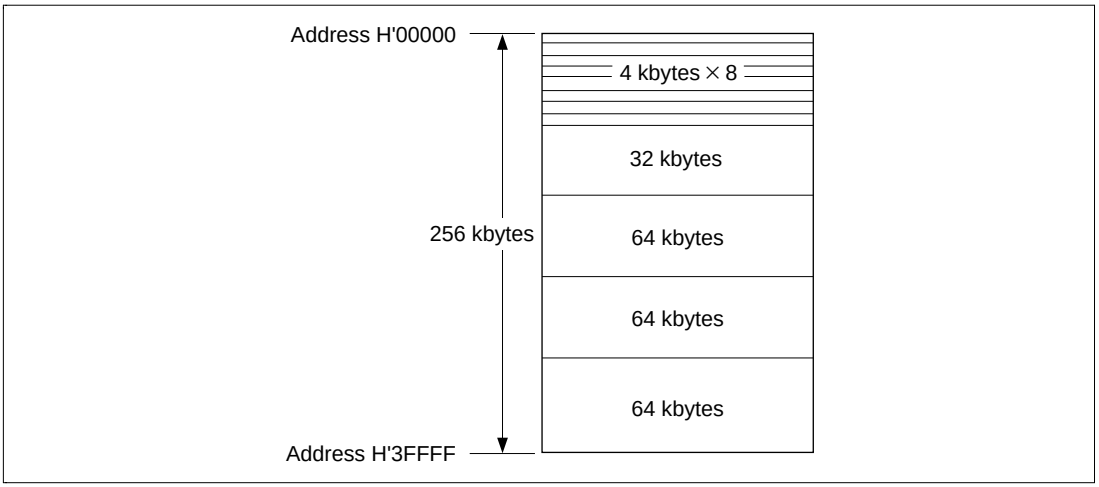


Figure 22-6 Flash Memory Block Configuration

22.4.8 Pin Configuration

The flash memory is controlled by means of the pins shown in table 22-5.

Table 22-5 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash memory program/erase protection by hardware
Mode 2	MD2	Input	Sets MCU operating mode
Mode 1	MD1	Input	Sets MCU operating mode
Mode 0	MD0	Input	Sets MCU operating mode
Port F0	PF0	Input	Sets MCU operating mode in programmer mode
Port 16	P16	Input	Sets MCU operating mode in programmer mode
Port 14	P14	Input	Sets MCU operating mode in programmer mode
Transmit data	TxD2	Output	Serial transmit data output
Receive data	RxD2	Input	Serial receive data input

22.4.9 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 22-6. In order to access these registers, the FLSHE bit in SCRX must be set to 1 (except for RAMER, SCRX).

Table 22-6 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*5	R/W*2	H'00*3	H'FFA8
Flash memory control register 2	FLMCR2*5	R*2	H'00	H'FFA9
Erase block register 1	EBR1*5	R/W*2	H'00*4	H'FFAA
Erase block register 2	EBR2*5	R/W*2	H'00*4	H'FFAB
RAM emulation register	RAMER*5	R/W	H'00	H'FEDB
Flash memory power control register	FLPWCR*5	R/W*2	H'00*4	H'FFAC
Serial control register X	SCRX	R/W	H'00	H'FDB4

Notes: *1 Lower 16 bits of the address.

*2 To access these registers, set the FLSHE bit to 1 in serial control register X. Even if FLSHE is set to 1, if the chip is in a mode in which the on-chip flash memory is disabled, a read will return H'00 and writes are invalid. Writes are also invalid when the FWE bit in FLMCR1 is not set to 1.

*3 When a high level is input to the FWE pin, the initial value is H'80.

*4 When a low level is input to the FWE pin, or if a high level is input and the SWE1 bit in FLMCR1 is not set, these registers are initialized to H'00.

*5 FLMCR1, FLMCR2, EBR1, and EBR2, RAMER, and FLPWCR are 8-bit registers. Use byte access on these registers.

22.5 Register Descriptions

22.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000 to H'3FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the PV1 or EV1 bit. Program mode for addresses H'00000 to H'3FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the PSU1 bit, and finally setting the P1 bit. Erase mode for addresses H'00000 to H'3FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the ESU1 bit, and finally setting the E1 bit. FLMCR1 is initialized by a power-on reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes are enabled only in the following cases: Writes to bit SWE1 of FLMCR1 enabled when FWE = 1, to bits ESU1, PSU1, EV1, and PV1 when FWE = 1 and SWE1 = 1, to bit E1 when FWE = 1, SWE1 = 1 and ESU1 = 1, and to bit P1 when FWE = 1, SWE1 = 1, and PSU1 = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1
Initial value:	—*	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7

FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit 1 (SWE1): This bit selects write and erase valid/invalid of the flash memory. Set it when setting bits 5 to 0, bits 7 to 0 of EBR1, and bits 3 to 0 of EBR2.

Bit 6

SWE1	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit 1 (ESU1): Prepares for a transition to erase mode. Set this bit to 1 before setting the E1 bit in FLMCR1 to 1. Do not set the SWE1, PSU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 5

ESU1	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 4—Program Setup Bit 1 (PSU1): Prepares for a transition to program mode. Set this bit to 1 before setting the P1 bit in FLMCR1 to 1. Do not set the SWE1, ESU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 4		
PSU1	**Description**	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or clearing. Do not set the SWE1, ESU1, PSU1, PV1, E1, or P1 bit at the same time.

Bit 3		
EV1	**Description**	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 2—Program-Verify 1 (PV1): Selects program-verify mode transition or clearing. Do not set the SWE1, ESU1, PSU1, EV1, E1, or P1 bit at the same time.

Bit 2		
PV1	**Description**	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 1—Erase 1 (E1): Selects erase mode transition or clearing. Do not set the SWE1, ESU1, PSU1, EV1, PV1, or P1 bit at the same time.

Bit 1

E1	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE1 = 1, and ESU1 = 1	

Bit 0—Program 1 (P1): Selects program mode transition or clearing. Do not set the SWE1, PSU1, ESU1, EV1, PV1, or E1 bit at the same time.

Bit 0

P1	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE1 = 1, and PSU1 = 1	

22.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. When on-chip flash memory is disabled, a read will return H'00.

Bit:	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	—	—	—

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7

FLER	Description
0	Flash memory is operating normally (Initial value) Flash memory program/erase protection (error protection) is disabled [Clearing condition] Power-on reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See 22.8.3 Error Protection

Bits 6 to 0—Reserved: These bits always read 0.

22.5.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE1 bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory erase block configuration is shown in table 22-7.

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.5.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin. Bit 0 will be initialized to 0 if bit SWE1 of FLMCR1 is not set, even though a high level is input to pin FWE. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. Bits 7 to 4 are reserved and must only be written with 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory erase block configuration is shown in table 22-7.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EB11	EB10	EB9	EB8
Initial value:	0	0	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 22-7 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (4 kbytes)	H'000000–H'000FFF
EB1 (4 kbytes)	H'001000–H'001FFF
EB2 (4 kbytes)	H'002000–H'002FFF
EB3 (4 kbytes)	H'003000–H'003FFF
EB4 (4 kbytes)	H'004000–H'004FFF
EB5 (4 kbytes)	H'005000–H'005FFF
EB6 (4 kbytes)	H'006000–H'006FFF
EB7 (4 kbytes)	H'007000–H'007FFF
EB8 (32 kbytes)	H'008000–H'00FFFF
EB9 (64 kbytes)	H'010000–H'01FFFF
EB10 (64 kbytes)	H'020000–H'02FFFF
EB11 (64 kbytes)	H'030000–H'03FFFF

22.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 22-8. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits always read 0.

Bits 5 and 4—Reserved: Only 0 may be written to these bits.

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 3

RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 2 to 0—Flash Memory Area Selection: These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 22-8.)

Table 22-8 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM1	RAM1	RAM0
H'FFD000–H'FFDFFF	RAM area 4 kbytes	0	*	*	*
H'000000–H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000–H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000–H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000–H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000–H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000–H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000–H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000–H'007FFF	EB7 (4 kbytes)	1	1	1	1

*: Don't care

22.5.6 Flash Memory Power Control Register (FLPWCR)

Bit:	7	6	5	4	3	2	1	0
	PDWND	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode.

Bit 7—Power-Down Disable (PDWND): Enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode.

Bit 7

PDWND	Description	
0	Transition to flash memory power-down mode enabled	(Initial value)
1	Transition to flash memory power-down mode disabled	

Bits 6 to 0—Reserved: These bits always read 0.

22.5.7 Serial Control Register X (SCRX)

Bit:	7	6	5	4	3	2	1	0
	—	IICX1	IICX0	IICE	FLSHE	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCRX is an 8-bit readable/writable register that controls on-chip flash memory.

SCRX is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Reserved: This bit should always be written with 0.

Bits 6 and 5—I^2C Transfer Rate Select (IICX1, IICX0): These bits, together with bits CKS2 to CKS0 in ICMR, select the transfer rate in master mode. For details of the transfer rate, see section 18.2.4, I^2C Bus Mode Register (ICMR).

Bit 4—I^2C Master Enable (IICE): Controls access to the I^2C bus interface data registers and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR). For details of the control, see section 18.2.7, Serial Control Register X (SCRX).

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). Setting the FLSHE bit to 1 enables read/write access to the flash memory control registers. If FLSHE is cleared to 0, the flash memory control registers are deselected. In this case, the flash memory control register contents are retained.

Bit 3	
FLSHE	**Description**
0	Flash control registers deselected in area H'FFFFA8 to H'FFFFAC (Initial value)
1	Flash control registers selected in area H'FFFFA8 to H'FFFFAC

Bits 2 to 0—Reserved: Should always be written with 0.

22.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 22-9. For a diagram of the transitions to the various flash memory modes, see figure 22-11.

Table 22-9 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1	MD0
Boot mode	Expanded mode	1	0	1	0
	Single-chip mode		0	1	1
User program mode	Expanded mode	1	1	1	0
	Single-chip mode		1	1	1

22.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2633 Series' pins have been set to boot mode, the boot program built into the H8S/2633 Series is started and the programming control program prepared in the host is serially transmitted to the H8S/2633 Series via the SCI. In the H8S/2633 Series, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 22-7, and the boot mode execution procedure in figure 22-8.

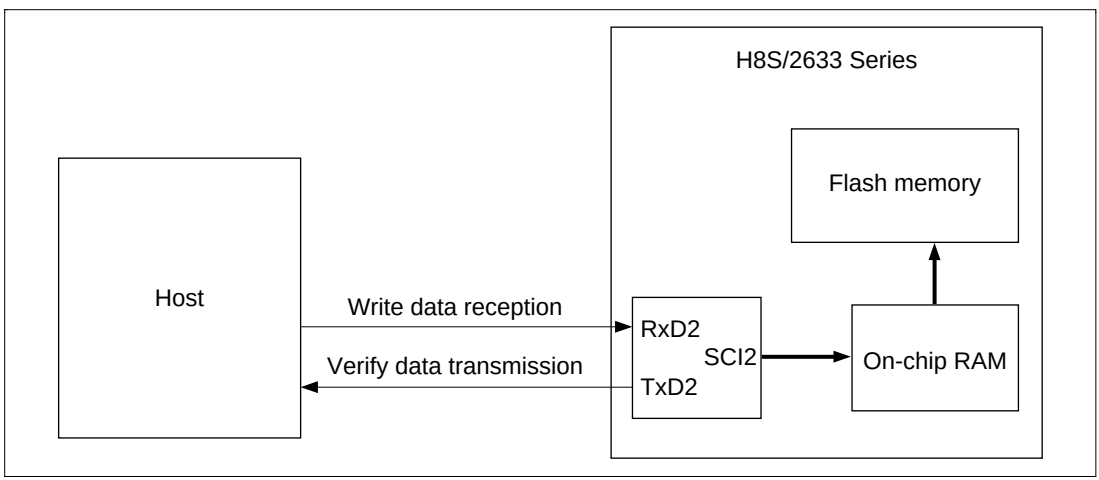
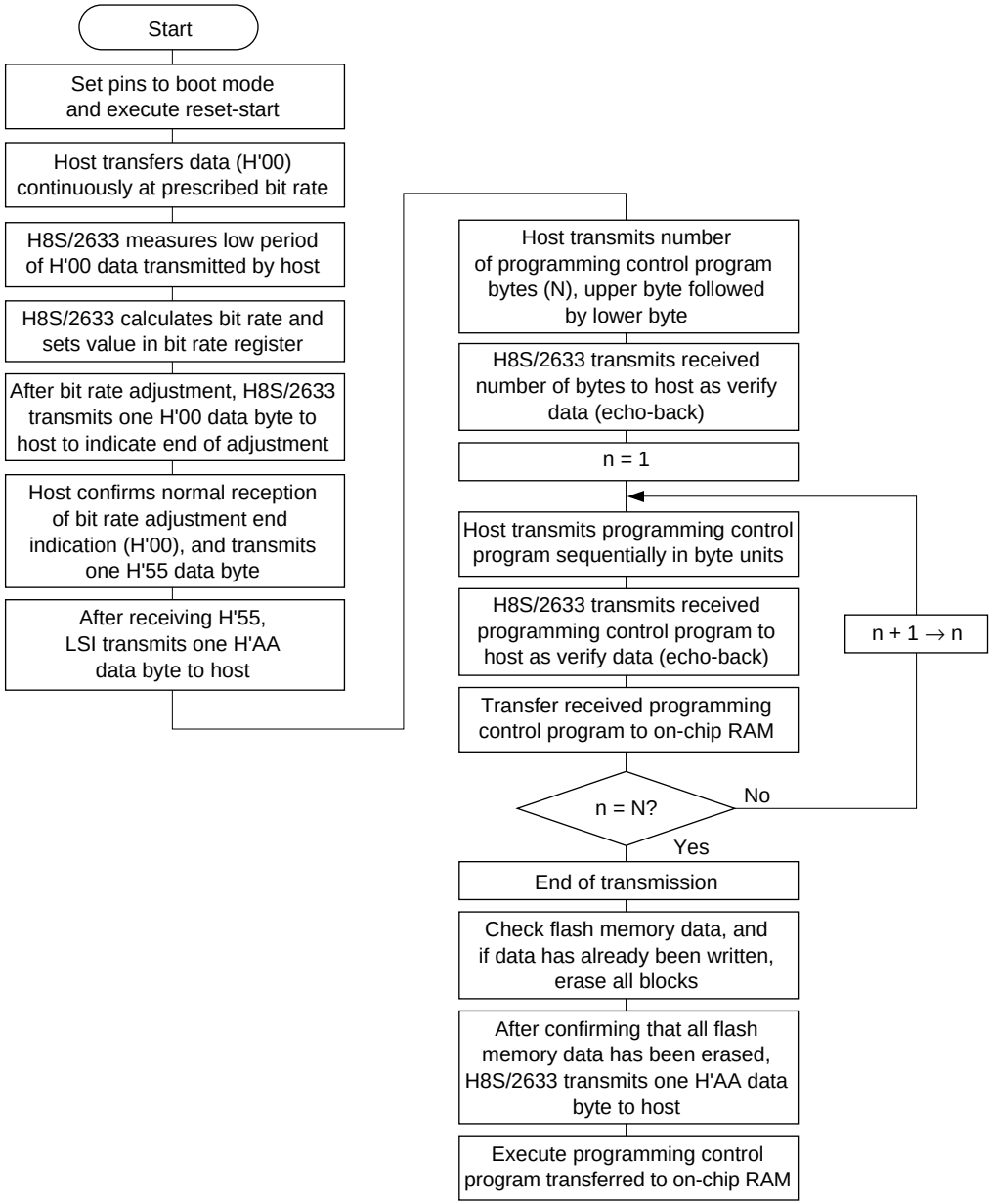


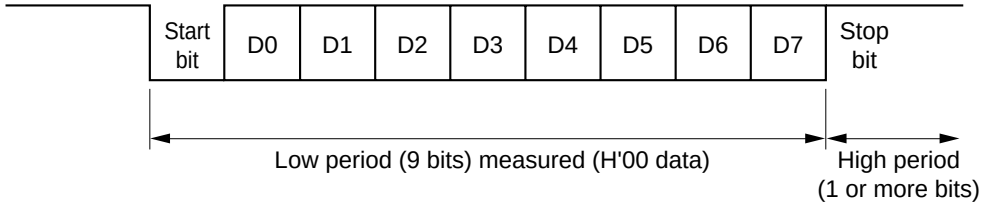
Figure 22-7 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 22-8 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment



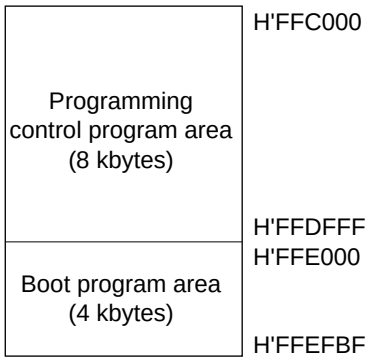
When boot mode is initiated, the H8S/2633 Series measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The H8S/2633 Series calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the H8S/2633 Series. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the H8S/2633 Series' system clock frequency, there will be a discrepancy between the bit rates of the host and the H8S/2633 Series. Set the host transfer bit rate at 2,400, 4,800, 9,600 or 19,200 bps to operate the SCI properly.

Table 22-10 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the H8S/2633 Series bit rate is possible. The boot program should be executed within this system clock range.

Table 22-10 System Clock Frequencies for which Automatic Adjustment of H8S/2633 Series Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of H8S/2633 Series Bit Rate is Possible
2,400 bps	2 to 8 MHz
4,800 bps	4 to 16 MHz
9,600 bps	8 to 25 MHz
19,200 bps	16 to 25 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 22-9. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.



Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

Figure 22-9 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the input at the SCI's RxD2 pin. The reset should end with RxD2 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD2 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD2 and TxD2 pins should be pulled up on the board.
- Before branching to the programming control program (RAM area H'FFC000), the chip terminates transmit and receive operations by the on-chip SCI (channel 2) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD2, goes to the high-level output state (PA1DDR = 1, PA1DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

The initial values of other on-chip registers are not changed.

- Boot mode can be entered by making the pin settings shown in table 22-9 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*[1]. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low while the boot program is being executed or while flash memory is being programmed or erased*[2].

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*[3].

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: *1 Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 4 states) with respect to the reset release timing.

*2 For further information on FWE application and disconnection, see section 22.13, Flash Memory Programming and Erasing Precautions.

*3 See appendix D, Pin States.

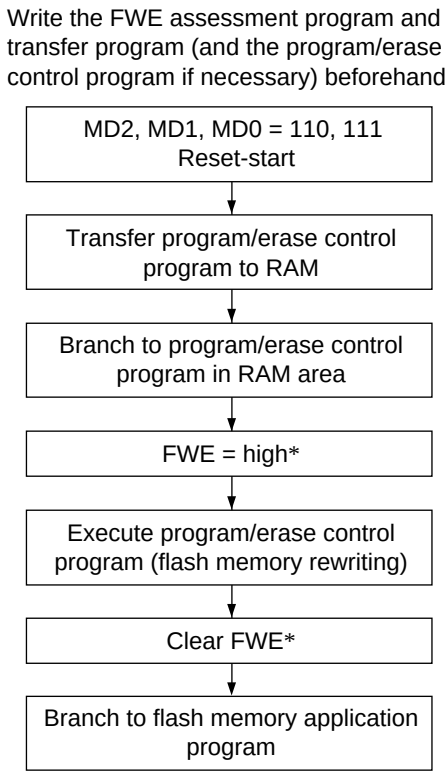
22.6.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE1 bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM.

Figure 22-10 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Note: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin only when the flash memory is programmed or erased. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

* For further information on FWE application and disconnection, see section 22.13, Flash Memory Programming and Erasing Precautions.

Figure 22-10 User Program Mode Execution Procedure

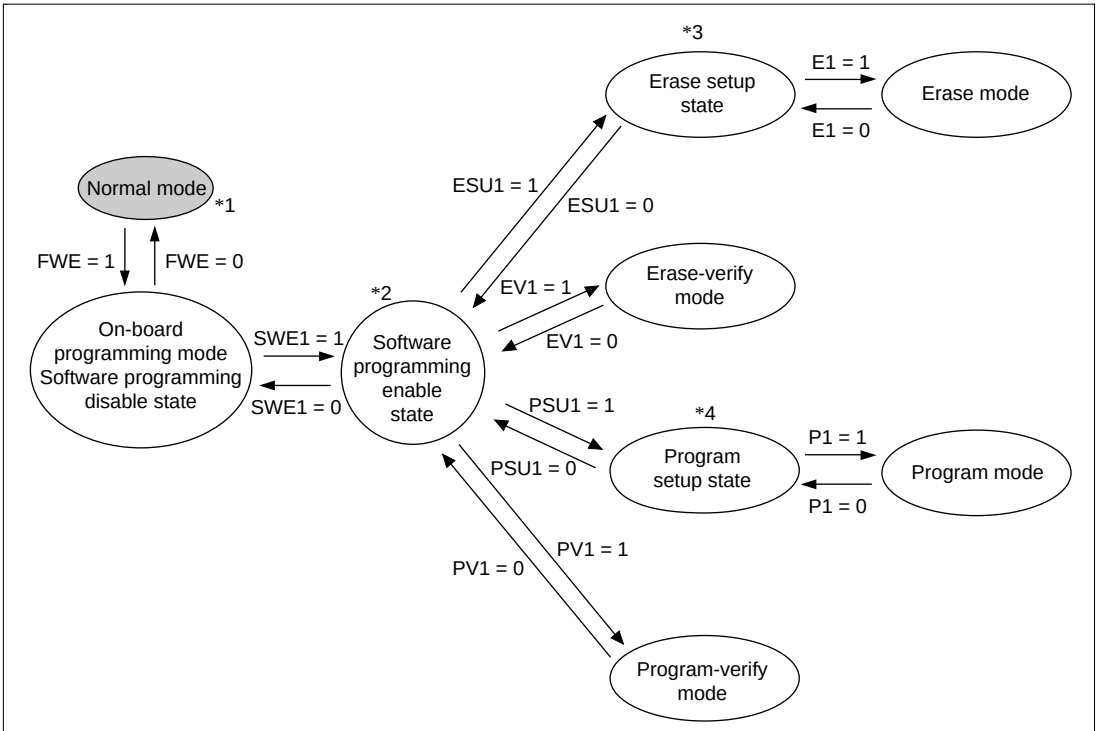
22.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU1, ESU1, P1, E1, PV1, and EV1 bits in FLMCR1 for addresses H'000000 to H'03FFFF.

The flash memory cannot be read while it is being written or erased. The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM. Also ensure that the DTC and DMAC is not activated before or after execution of the flash memory write instruction.

In the following operation descriptions, wait times after setting or clearing individual bits in FLMCR1 are given as parameters; for details of the wait times, see section 25.6, Flash Memory Characteristics.

Notes:
1. Operation is not guaranteed if bits SWE1, ESU1, PSU1, EV1, PV1, E1, and P1 of FLMCR1 are set/reset by a program in flash memory in the corresponding address areas.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.



Notes: In order to perform a normal read of flash memory, SWE must be cleared to 0. Also note that verify-reads can be performed during the programming/erasing process.

*1 : Normal mode : On-board programming mode
*2 Do not make a state transition by setting or clearing multiple bits simultaneously.
*3 After a transition from erase mode to the erase setup state, do not enter erase mode without passing through the software programming enable state.
*4 After a transition from program mode to the program setup state, do not enter program mode without passing through the software programming enable state.

Figure 22-11 FLMCR1 Bit Settings and State Transitions

22.7.1 Program Mode

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 22-12 should be followed. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

The wait times after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N1 + N2) are shown in table 25-14 in section 25.6, Flash Memory Characteristics.

Following the elapse of (x0) µs or more after the SWE1 bit is set to 1 in FLMCR1, 128-byte program data is stored in the program data area and reprogram data area, and the 128-byte data in the program data area in RAM is written consecutively to the program address (the lower 8 bits of the first address written to must be H'00 or H'80). 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a value greater than (y + z2 + α + β) ms as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSU1 bit in FLMCR1, and after the elapse of (y) µs or more, the operating mode is switched to program mode by setting the P1 bit in FLMCR1. The time during which the P1 bit is set is the flash memory programming time. Refer to the table in figure 22-12 for the programming time.

22.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of the given programming time, clear the P1 bit in FLMCR1, then wait for at least (α) µs before clearing the PSU1 bit to exit program mode. After the elapse of at least (β) µs, the watchdog timer is cleared and the operating mode is switched to program-verify mode by setting the PV1 bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 22-12) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait for at least (η) µs, then clear the SWE1 bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. The maximum number of repetitions of the program/program-verify sequence is indicated by the maximum programming count (N1 + N2). However, ensure that the program/program-verify sequence is not repeated more than (N1 + N2) times on the same bits.

Notes on Program/Program-Verify Procedure

1. In order to perform 128-byte-unit programming, the lower 8 bits of the write start address must be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit transfer should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data. Write H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P1 bit in FLMCR1 is set. In the H8S/2633, write pulses should be applied as follows in the program/program-verify procedure to prevent voltage stress on the device and loss of write data reliability.
   a. After write pulse application, perform a verify-read in program-verify mode and apply a write pulse again for any bits read as 1 (reprogramming processing). When all the 0-write bits in the 128-byte write data are read as 0 in the verify-read operation, the program/program-verify procedure is completed. In the H8S/2633, the number of loops in reprogramming processing is guaranteed not to exceed the maximum value of the maximum programming count (N).
   b. After write pulse application, a verify-read is performed in program-verify mode, and programming is judged to have been completed for bits read as 0.

   c. If programming of other bits is incomplete in the 128 bytes, reprogramming processing should be executed. If a bit for which programming has been judged to be completed is read as 1 in a subsequent verify-read, a write pulse should again be applied to that bit.

5. The period for which the P1 bit in FLMCR1 is set (the write pulse width) should be changed according to the degree of progress through the program/program-verify procedure. For detailed wait time specifications, see section 25.6, Flash Memory Characteristics.

6. The program/program-verify flowchart for the H8S/2633 is shown in figure 22-12.

   To cover the points noted above, bits on which reprogramming processing is to be executed, and bits on which additional programming is to be executed, must be determined as shown below.

   Since reprogram data and additional-programming data vary according to the progress of the programming procedure, it is recommended that the following data storage areas (128 bytes each) be provided in RAM.

Reprogram Data Computation Table

(D)	Result of Verify-Read after Write Pulse Application (V)	(X) Result of Operation	Comments
0	0	1	Programming completed: reprogramming processing not to be executed
0	1	0	Programming incomplete: reprogramming processing to be executed
1	0	1	—
1	1	1	Still in erased state: no action

Legend

(D): Source data of bits on which programming is executed

(X): Source data of bits on which reprogramming is executed

Additional-Programming Data Computation Table

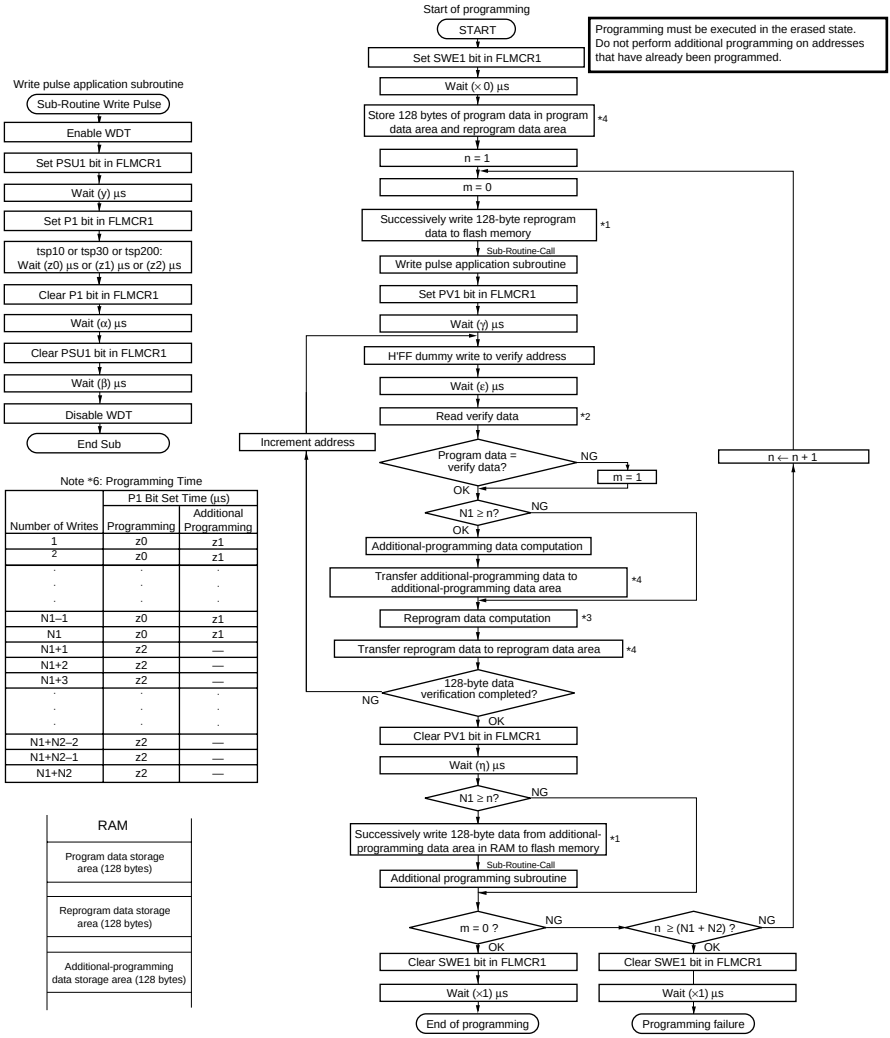
(X')	Result of Verify-Read after Write Pulse Application (V)	(Y) Result of Operation	Comments
0	0	0	Programming by write pulse application judged to be completed: additional programming processing to be executed
0	1	1	Programming by write pulse application incomplete: additional programming processing not to be executed
1	0	1	Programming already completed: additional programming processing not to be executed
1	1	1	Still in erased state: no action

Legend

(Y): Data of bits on which additional programming is executed

(X'): Data of bits on which reprogramming is executed in a certain reprogramming loop

7. It is necessary to execute additional programming processing during the course of the H8S/2633 program/program-verify procedure. However, once 128-byte-unit programming is finished, additional programming should not be carried out on the same address area. When executing reprogramming, an erase must be executed first. Note that normal operation of reads, etc., is not guaranteed if additional programming is performed on addresses for which a program/program-verify operation has finished.



Note *6: Programming Time

Number of Writes	P1 Bit Set Time (µs) Programming	Additional Programming
1	z0	z1
2	z0	z1
. . .	. . .	. . .
N1–1	z0	z1
N1	z0	z1
N1+1	z2	—
N1+2	z2	—
N1+3	z2	—
. . .	. . .	. . .
N1+N2–2	z2	—
N1+N2–1	z2	—
N1+N2	z2	—

Notes: *1 Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
*2 Verify data is read in 16-bit (word) units.
*3 Even bits for which programming has been completed in the 128-byte programming loop will be subject to programming again if they fail the subsequent verify operation.
*4 A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional-programming data must be provided in RAM. The reprogram and additional-programming data contents are modified as programming proceeds.
*5 A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note 6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming complete
0	1	0	Programming is incomplete: reprogramming should be performed
1	0	1	
1	1	1	Left in the erased state

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (X)	Comments
0	0	0	Additional programming should be performed
0	1	1	Additional programming should not be performed
1	0	1	Additional programming should not be performed
1	1	1	Additional programming should not be performed

Figure 22-12 Program/Program-Verify Flowchart

22.7.3 Erase Mode

When erasing flash memory, the single-block erase/erase-verify flowchart shown in figure 22-13 should be followed.

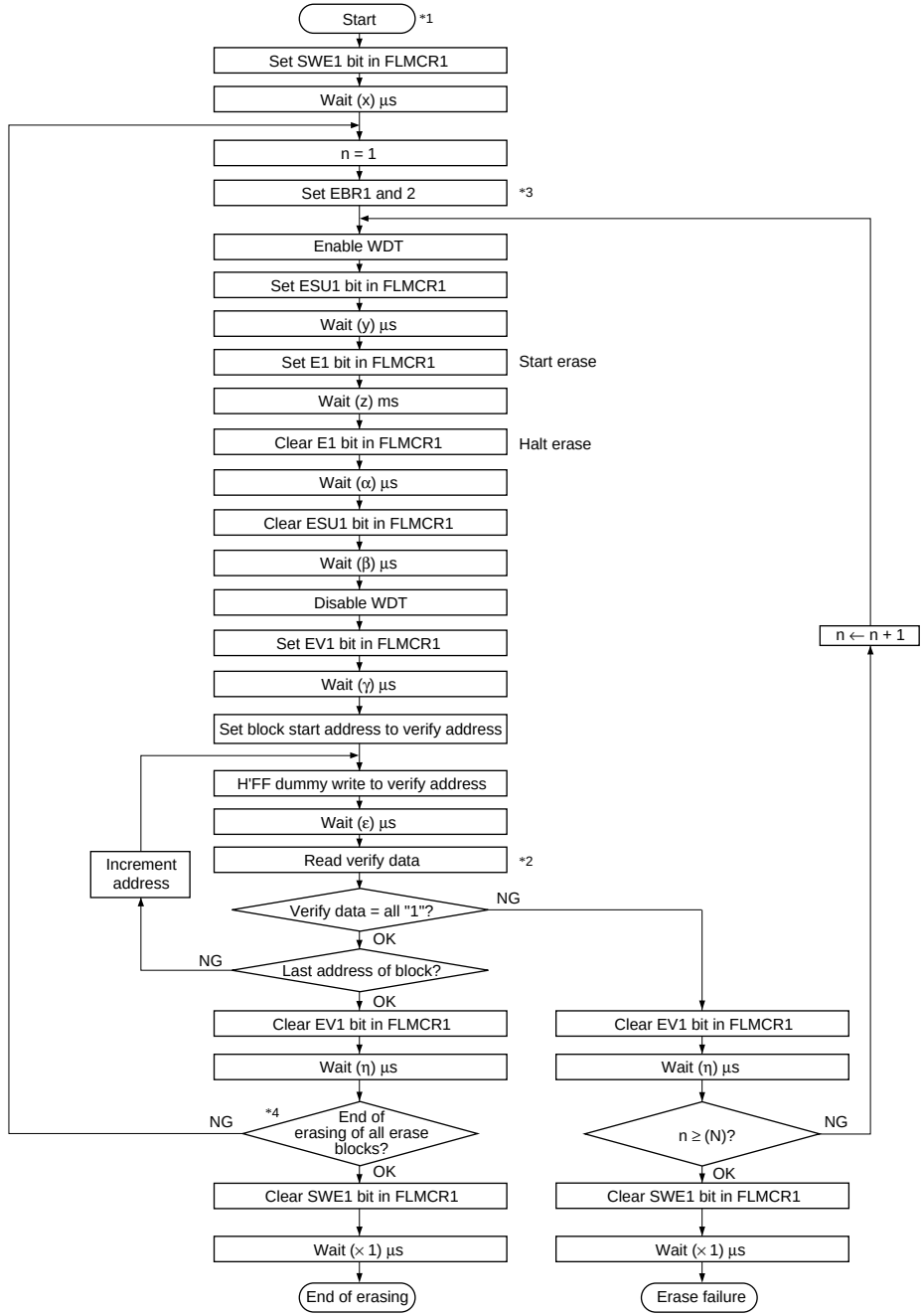
To erase flash memory contents, make a 1-bit setting for the flash memory area to be erased in erase block register 1 and 2 (EBR1, EBR2) at least (x) µs after setting the SWE1 bit to 1 in FLMCR1. Next, the watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than (y + z + α + β) ms as the WDT overflow period. Preparation for entering erase mode (erase setup) is performed next by setting the ESU1 bit in FLMCR1. The operating mode is then switched to erase mode by setting the E1 bit in FLMCR1 after the elapse of at least (y) µs. The time during which the E1 bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms.

Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all 0) is not necessary before starting the erase procedure.

22.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the fixed erase time, clear the E1 bit in FLMCR1, then wait for at least (α) µs before clearing the ESU1 bit to exit erase mode. After exiting erase mode, the watchdog timer is cleared after the elapse of (β) µs or more. The operating mode is then switched to erase-verify mode by setting the EV1 bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again and repeat the erase/erase-verify sequence in the same way. The maximum number of reoperations of the erase/erase-verify sequence is indicated by the maximum erase count (N). However, ensure that the erase/erase-verify sequence is not repeated more than (N) times. When verification is completed, exit erase-verify mode, and wait for at least (η) µs. If erasure has been completed on all the erase blocks, clear the SWE1 bit in FLMCR1. If there are any unerased blocks, make a 1 bit setting for the flash memory area to be erased, and repeat the erase/erase-verify sequence as before.



Notes: *1 Preprogramming (setting erase block data to all "0") is not necessary.
*2 Verify data is read in 16-bit (W) units.
*3 Set only one bit in EBR1 and 2. More than 2 bits cannot be set.
*4 Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.

Figure 22-13 Erase/Erase-Verify Flowchart

22.8 Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

22.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 22-11.)

Table 22-11 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, (except bit FLER) EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a power-on reset (including a WDT power-on reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

22.8.2 Software Protection

Software protection can be implemented by setting the SWE1 bit in FLMCR1, erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), does not cause a transition to program mode or erase mode. (See table 22-12.)

Table 22-12 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Setting bit SWE1 in FLMCR1 to 0 will place area H'000000 to H'03FFFF in the program/erase-protected state. (Execute the program in the on-chip RAM, external memory)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

22.8.3 Error Protection

In error protection, an error is detected when H8S/2633 Series runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the H8S/2633 Series malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P1 or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the CPU releases the bus to the DTC

Error protection is released only by a power-on reset and in hardware standby mode.

Figure 22-14 shows the flash memory state transition diagram.

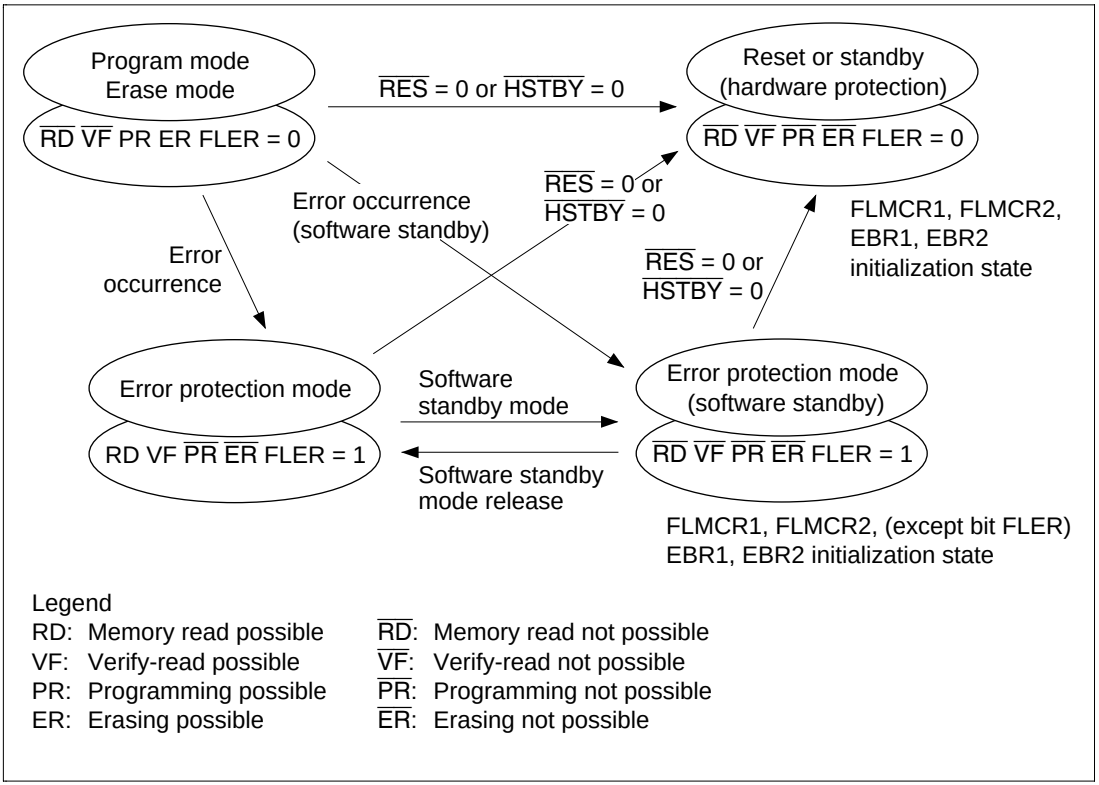


Figure 22-14 Flash Memory State Transitions

22.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses cannot be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 22-15 shows an example of emulation of real-time flash memory programming.

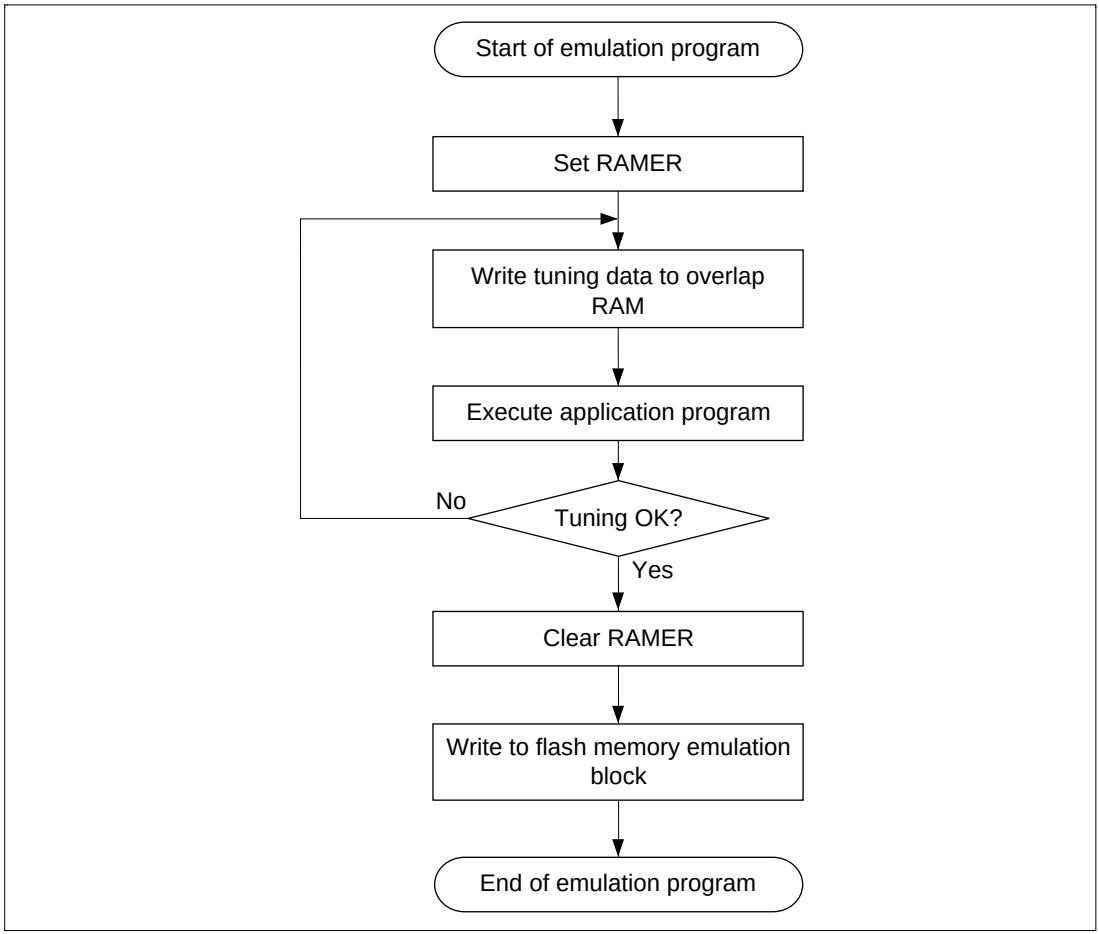


Figure 22-15 Flowchart for Flash Memory Emulation in RAM

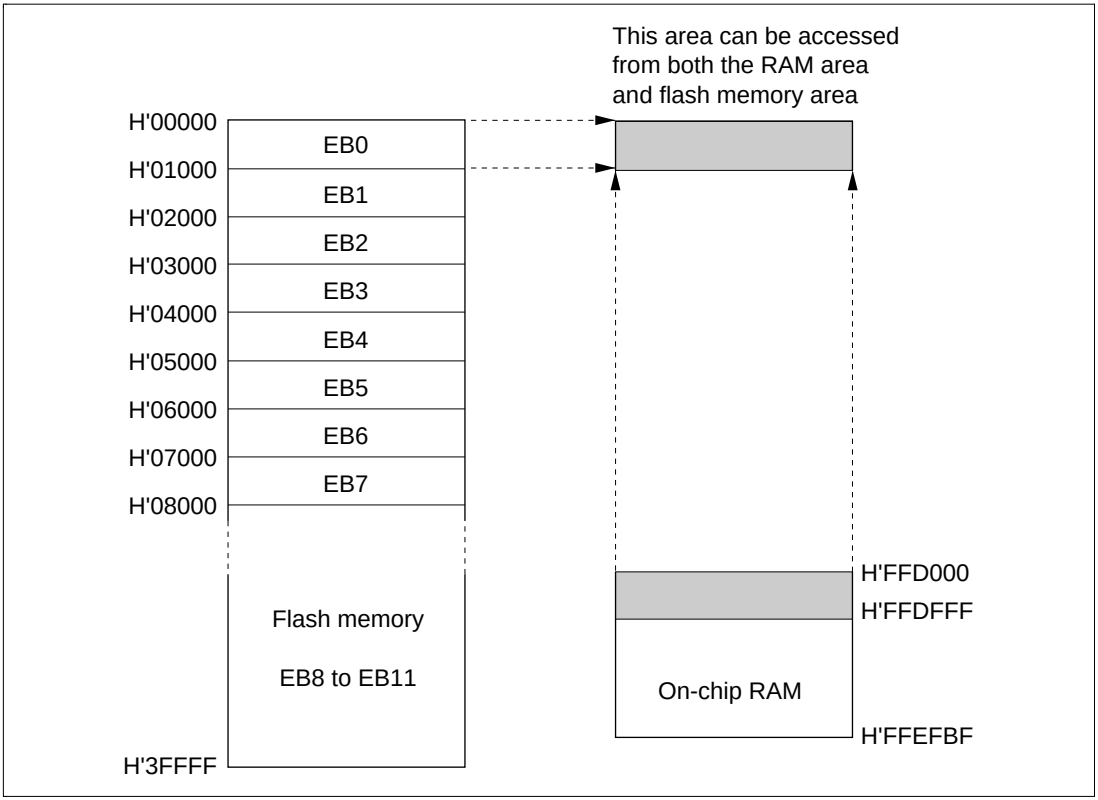


Figure 22-16 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB0 is Overlapped

1. Set bits RAMS, RAM2 to RAM0 in RAMER to 1, 0, 0, 0, to overlap part of RAM onto the area (EB0) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

22.10 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P1 or E1 bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupt, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All requests, including NMI interrupt, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. NMI interrupt is also disabled in the error-protection state while the P1 or E1 bit remains set in FLMCR1.

Notes: *1 Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.

*2 The vector may not be read correctly in this case for the following two reasons:

- If flash memory is read while being programmed or erased (while the P1 or E1 bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
- If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

22.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 22-13) and input a 12 MHz input clock.

Table 22-13 shows the pin settings for programmer mode. For the pin names in programmer mode, see figure 22-18, Socket Adapter Pin Correspondence Diagram.

Table 22-13 Programmer Mode Pin Settings

Pin Names	Settings
Mode pins: MD2, MD1, MD0	Low level input to MD2, MD1, and MD0.
Mode setting pins: PF0, P16, P14	High level input to PF0, low level input to P16 and P14
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL, PLLVCC, PLLCAP, PLLVSS pins	Oscillator circuit

22.11.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 22-18. This will enable conversion to a 40-pin arrangement. The on-chip ROM memory map is shown in figure 22-17, and the socket adapter pin correspondence diagram in figure 22-18.

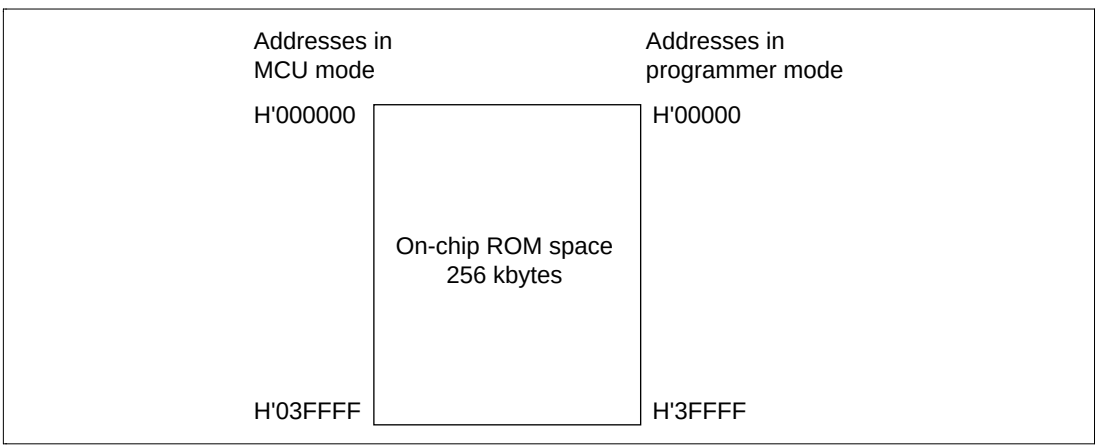


Figure 22-17 On-Chip ROM Memory Map

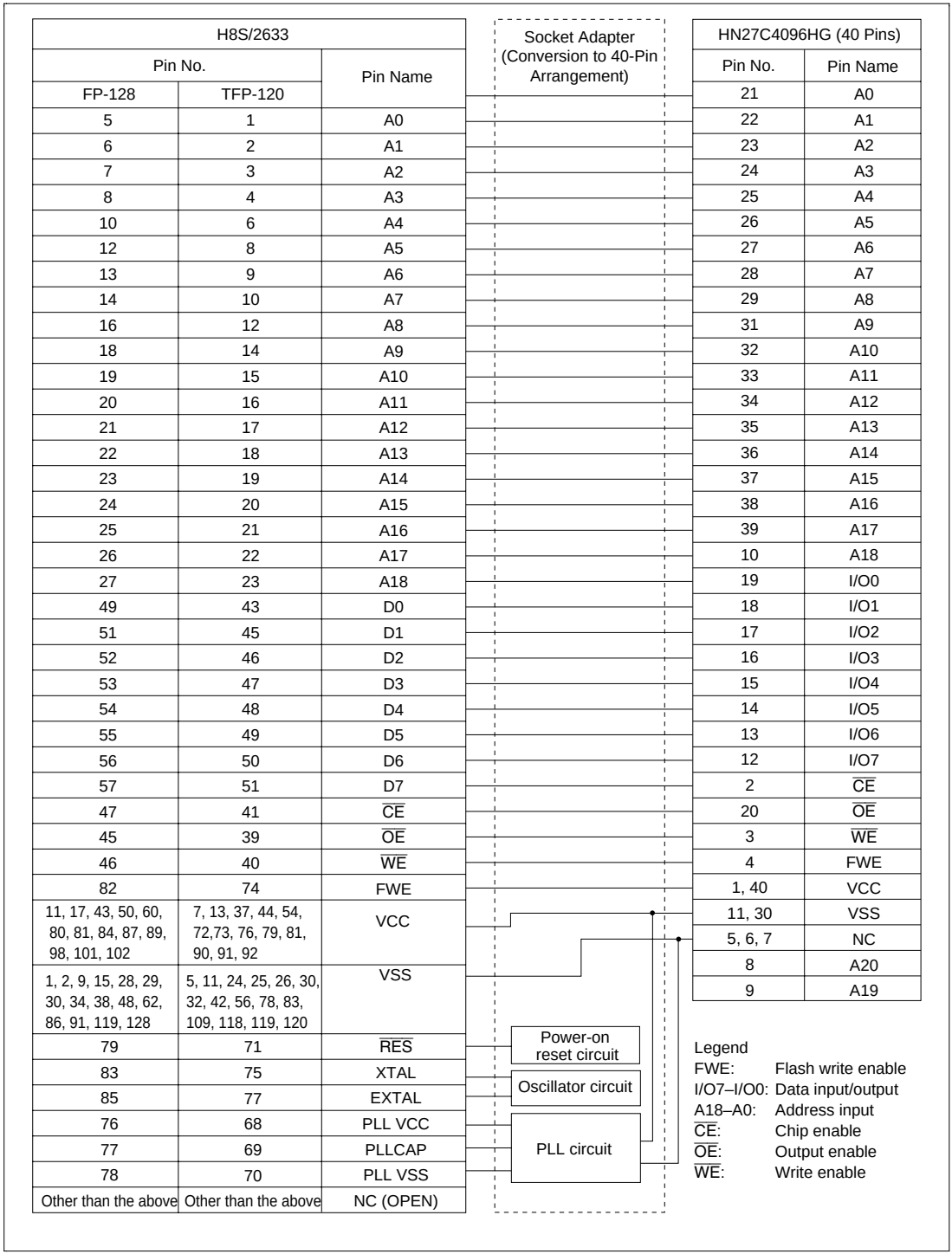


Figure 22-18 Socket Adapter Pin Correspondence Diagram

22.11.2 Programmer Mode Operation

Table 22-14 shows how the different operating modes are set when using programmer mode, and table 22-15 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 22-14 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A18–A0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L*3	L	H	L	Data input	Ain*2
Chip disable*1	H or L	H	X	X	Hi-z	X

Notes: *1 Chip disable is not a standby state; internally, it is an operation state.

*2 Ain indicates that there is also address input in auto-program mode.

*3 For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 22-15 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

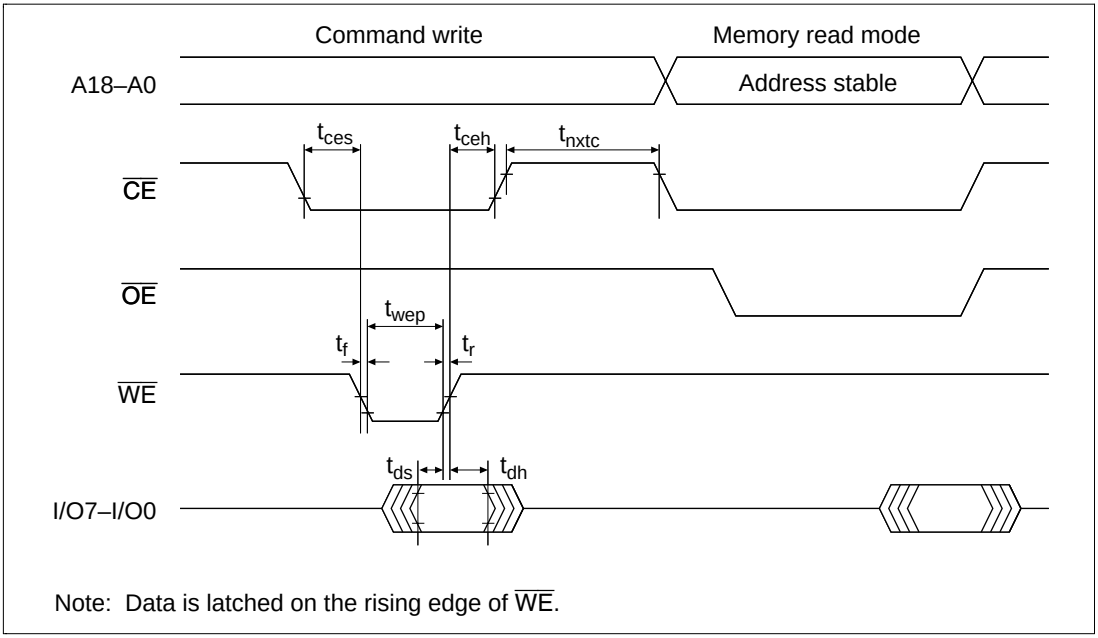
Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

22.11.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 22-16 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 22-19 Timing Waveforms for Memory Read after Memory Write

Table 22-17 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

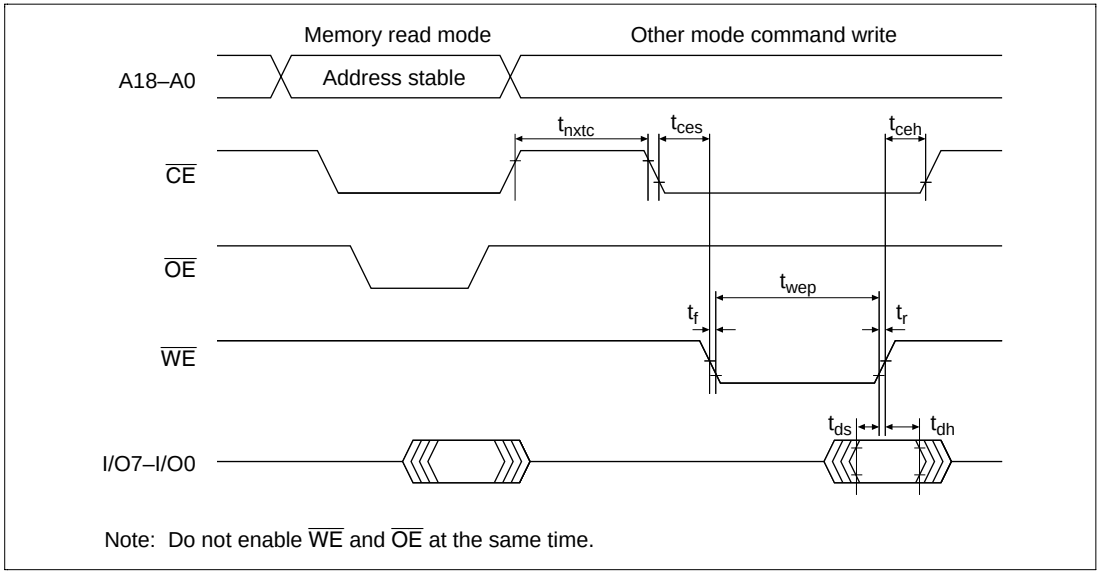


Figure 22-20 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 22-18 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Access time	t_{acc}	—	20	µs
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

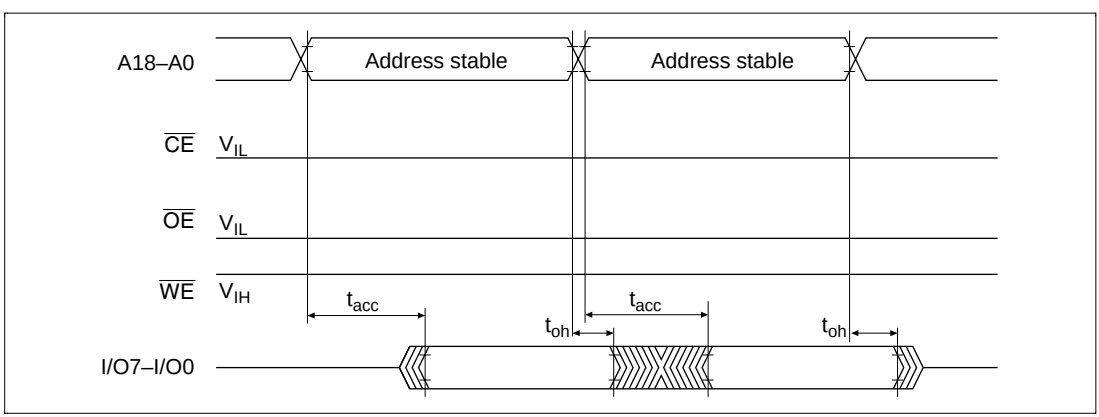


Figure 22-21 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

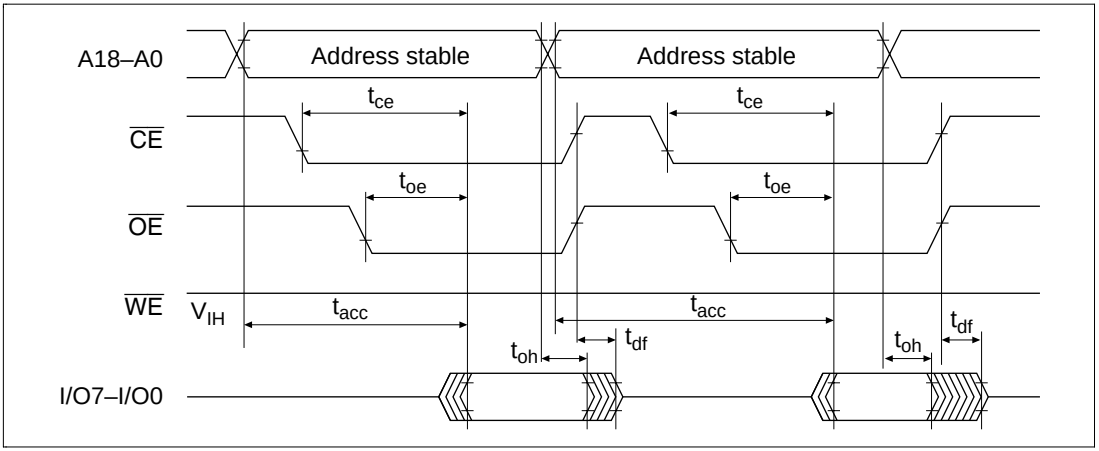


Figure 22-22 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

22.11.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 22-23). Do not perform transfer after the third cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end decision pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 22-19 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory write time	t_{write}	1	3000	ms
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

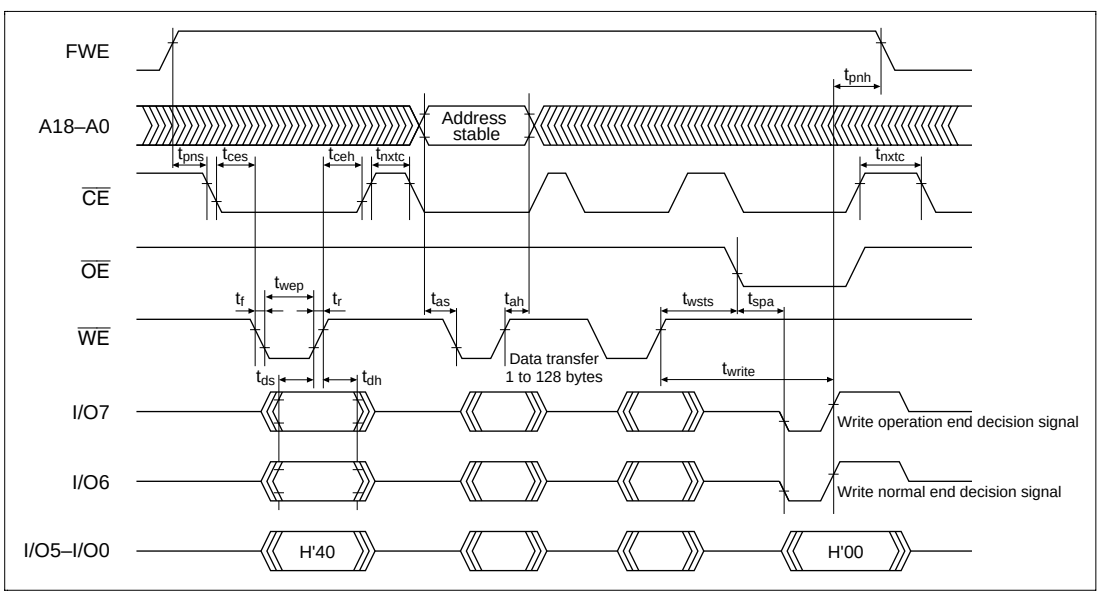


Figure 22-23 Auto-Program Mode Timing Waveforms

22.11.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-erase operation end decision pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 22-20 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
Erase setup time	t_{ens}	100	—	ns
Erase end setup time	t_{enh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

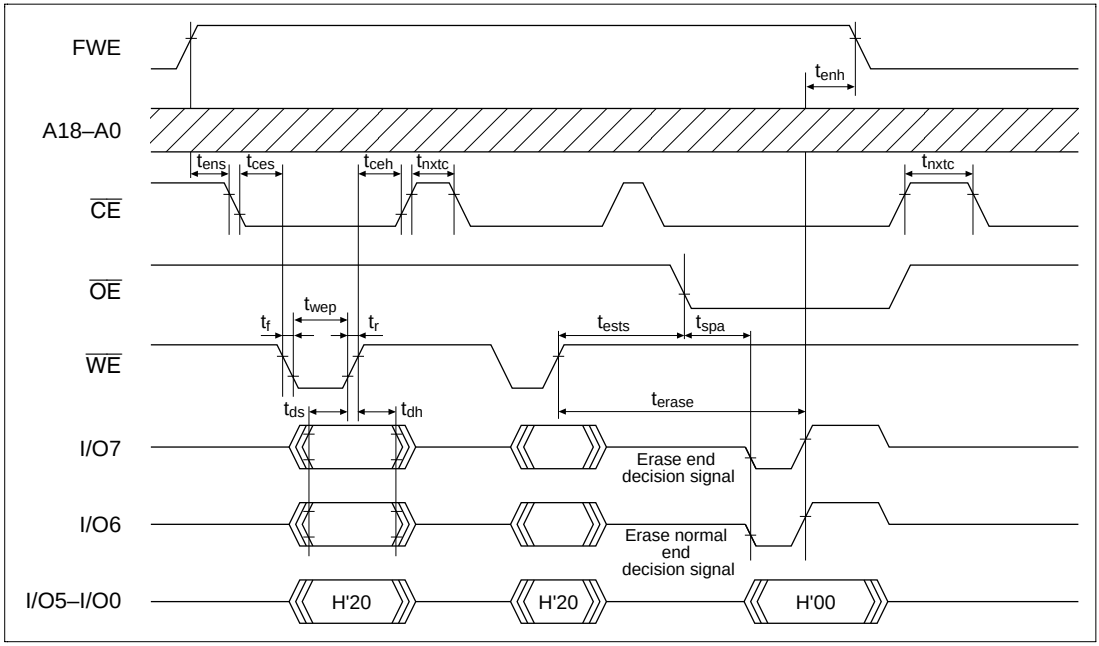


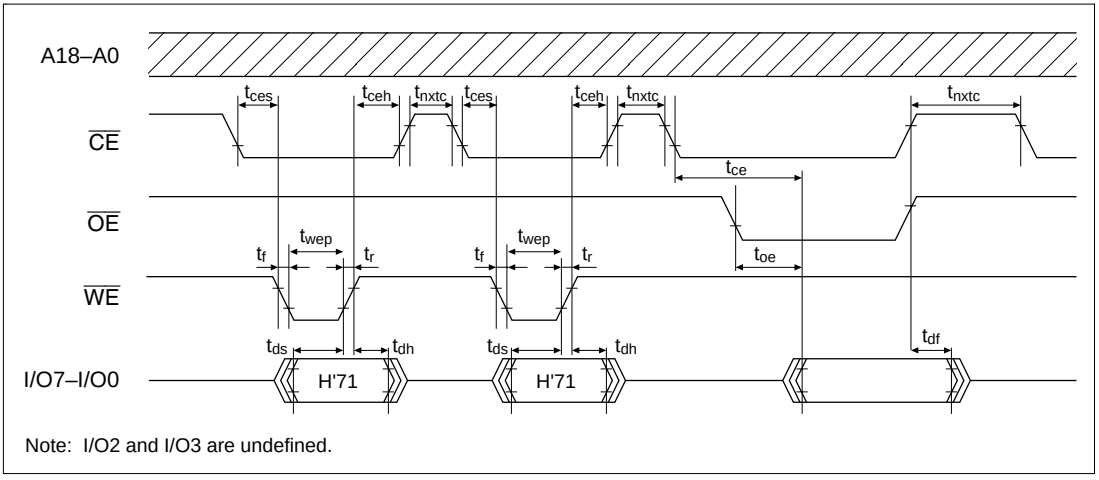
Figure 22-24 Auto-Erase Mode Timing Waveforms

22.11.6 Status Read Mode

1. Status read mode is provided to identify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 22-21 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Read time after command write	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Disable delay time	t_{df}	—	100	ns
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: I/O2 and I/O3 are undefined.

Figure 22-25 Status Read Mode Timing Waveforms

Table 22-22 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end decision	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined.

22.11.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-erase mode.
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 22-23 Status Polling Output Truth Table

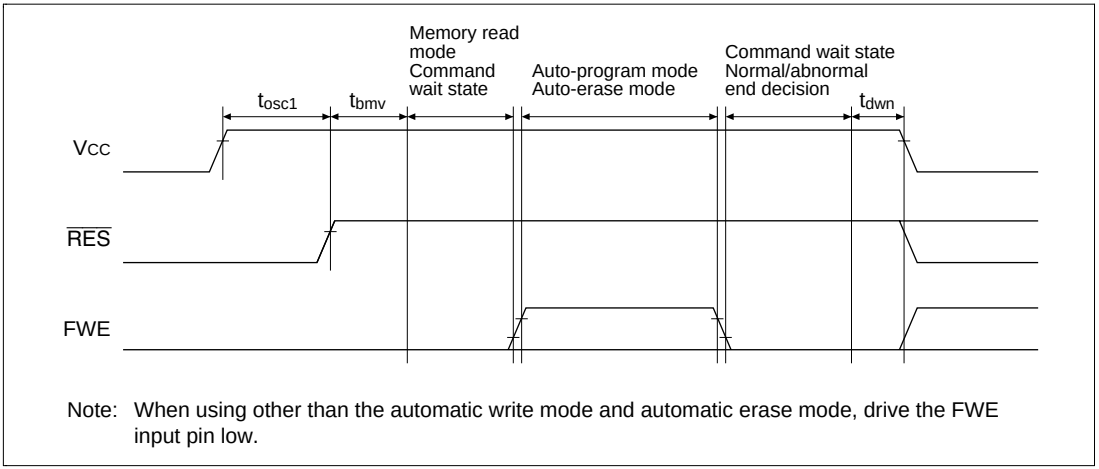
Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

22.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 22-24 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
Programmer mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms



Note: When using other than the automatic write mode and automatic erase mode, drive the FWE input pin low.

Figure 22-26 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

22.11.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

22.12 Flash Memory and Power-Down States

In addition to its normal operating state, the flash memory has power-down states in which power consumption is reduced by halting part or all of the internal power supply circuitry.

There are three flash memory operating states:

(1) Normal operating mode: The flash memory can be read and written to.
(2) Power-down mode: Part of the power supply circuitry is halted, and the flash memory can be read when the H8S/2633 is operating on the subclock.
(3) Standby mode: All flash memory circuits are halted, and the flash memory cannot be read or written to.

States (2) and (3) are flash memory power-down states. Table 22-25 shows the correspondence between the operating states of the H8S/2633 and the flash memory.

Table 22-25 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode	Normal mode (read/write)
Medium-speed mode	
Sleep mode	
Subactive mode	When PDWND = 0: Power-down mode (read-only)
Subsleep mode	When PDWND = 1: Normal mode (read-only)
Watch mode	Standby mode
Software standby mode	
Hardware standby mode	

22.12.1 Note on Power-Down States

When the flash memory is in a power-down state, part or all of the internal power supply circuitry is halted. Therefore, a power supply circuit stabilization period must be provided when returning to normal operation. When the flash memory returns to its normal operating state from a power-down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 20 µs (power supply stabilization time), even if an oscillation stabilization period is not necessary.

22.13 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Hitachi microcomputer device type with 256-kbyte on-chip flash memory (FZTAT256V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off (see figures 22-27 to 22-29): Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

FWE application/disconnection (see figures 22-27 to 22-29): FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
  Apply FWE when oscillation has stabilized (after the elapse of the oscillation stabilization time).
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 bits in FLMCR1 are cleared.
  Make sure that the SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin: Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P1 or E1 bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE1 bit during execution of a program in flash memory: Wait for at least 100 µs after clearing the SWE1 bit before executing a program or reading data in flash memory. When the SWE1 bit is set, data in flash memory can be rewritten, but when SWE1 = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE1 bit during programming, erasing, or verifying.

Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE1 bit must be cleared before executing a program or reading data in flash memory.

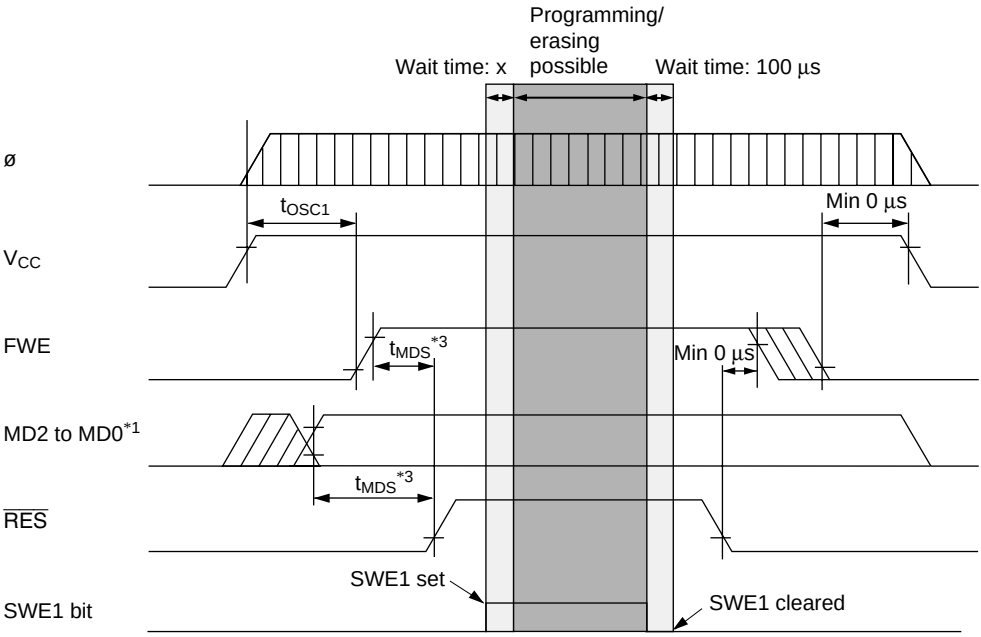
However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE1 bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased: All interrupt requests, including NMI, should be disabled during FWE application to give priority to program/erase operations.

Do not perform overwriting. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Before programming, check that the chip is correctly mounted in the PROM programmer: Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

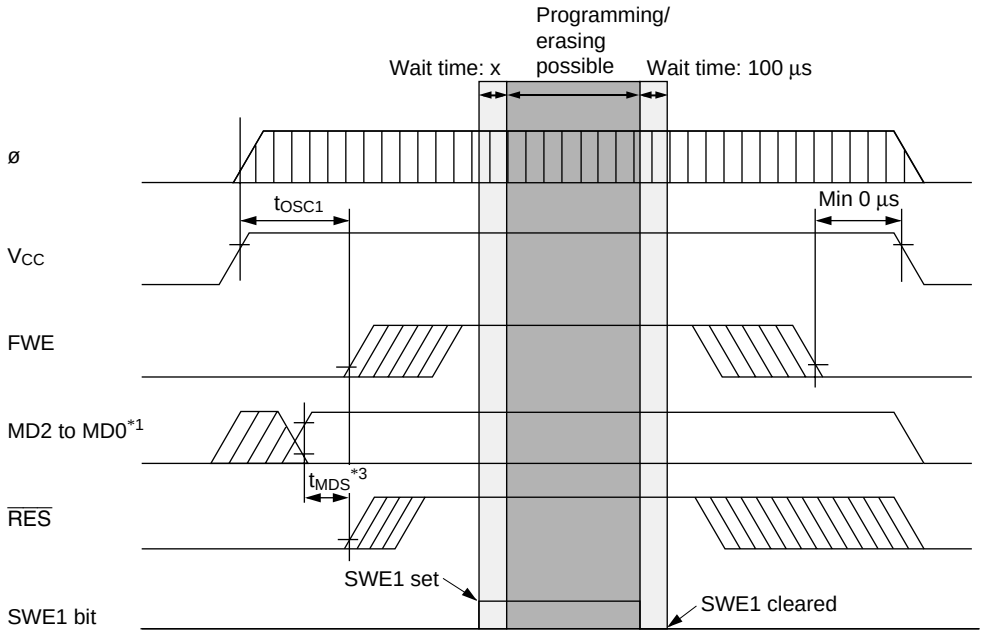


Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 25.6, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 22-27 Power-On/Off Timing (Boot Mode)

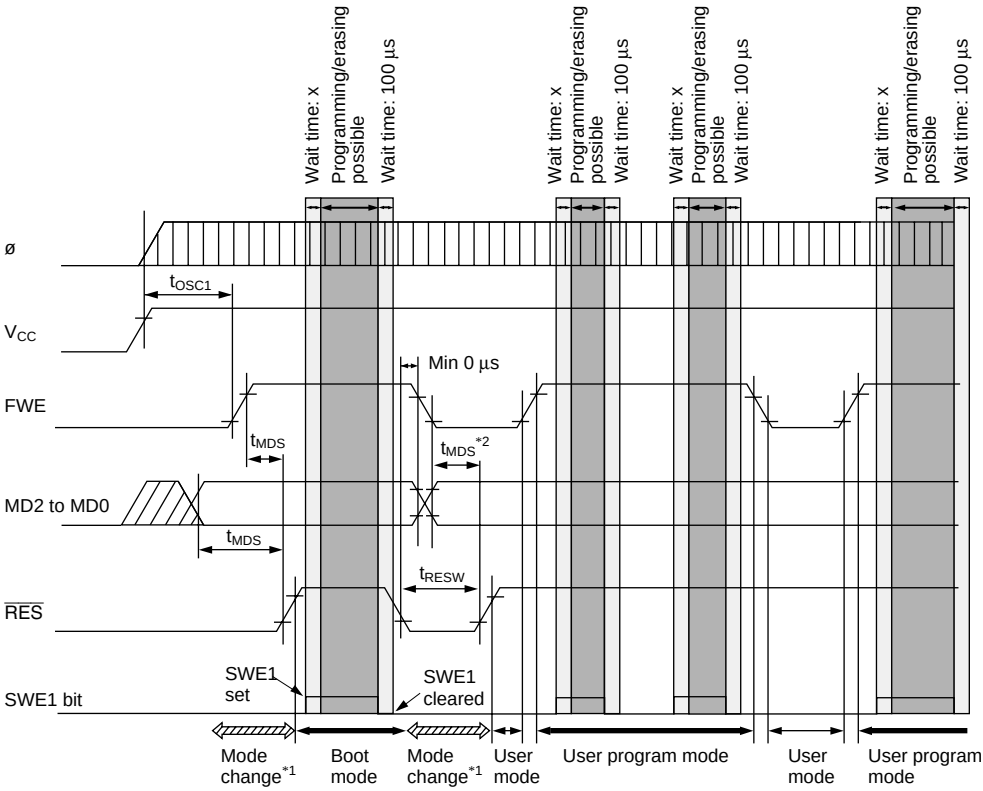


Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 25.6, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 22-28 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.

*2 When making a transition from boot mode to another mode, a mode programming setup time t_{MDS} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.

*3 See section 25.6, Flash Memory Characteristics.

Figure 22-29 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

22.14 Note on Switching from F-ZTAT Version to Mask ROM Version

The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 22-26 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 22-26 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 22-26 have no effect.

Table 22-26 Registers Present in F-ZTAT Version but Absent in Mask ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFC8
Flash memory control register 2	FLMCR2	H'FFC9
Erase block register 1	EBR1	H'FFCA
Erase block register 2	EBR2	H'FFCB
RAM emulation register	RAMER	H'FEDB

23.1 Overview

The H8S/2633 Series has a built-in clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator, PLL (phase-locked loop) circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit, subclock oscillator, and waveform shaping circuit. The frequency can be changed by means of the PLL circuit in the CPG. Frequency changes are performed by software by means of settings in the system clock control register (SCKCR) and low-power control register (LPWRCR).

23.1.1 Block Diagram

Figure 23-1 shows a block diagram of the clock pulse generator.

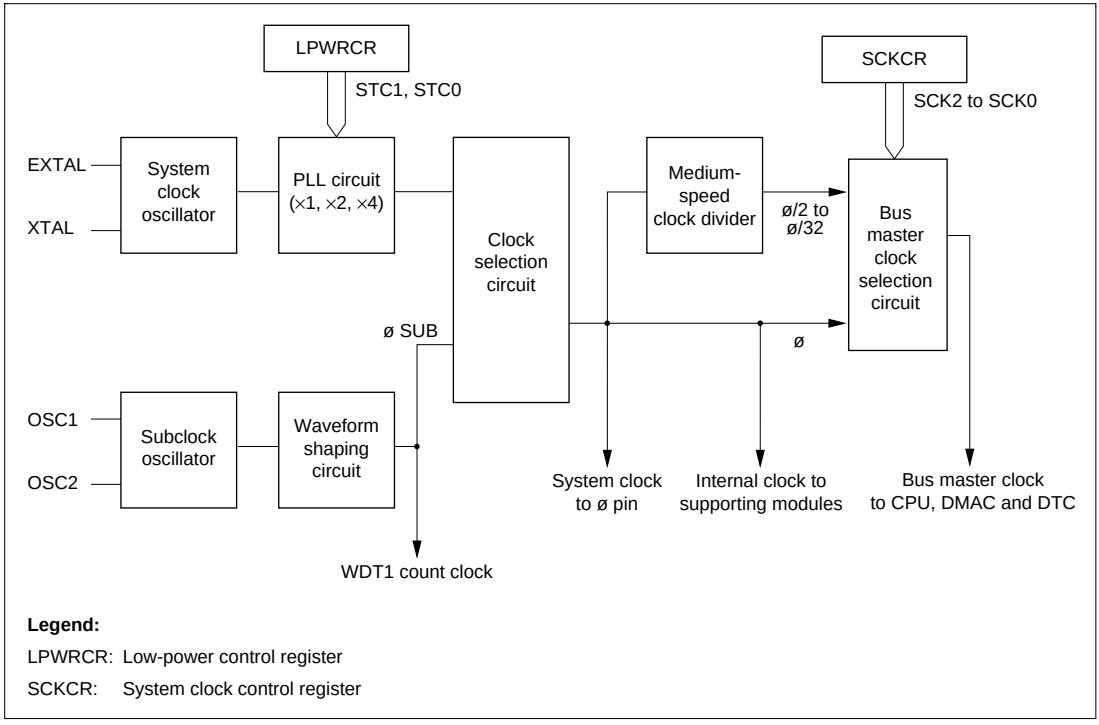


Figure 23-1 Block Diagram of Clock Pulse Generator

23.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR and LPWRCR. Table 23-1 shows the register configuration.

Table 23-1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC

Note:* Lower 16 bits of the address.

23.2 Register Descriptions

23.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): Controls ø output.

	Description			
Bit 7 PSTOP	**High-Speed Mode, Medium-Speed Mode, Sub-Active Mode**	**Sleep Mode Sub-Sleep Mode**	**Software Standby Mode, Watch Mode**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 and 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3	
STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode, watch mode, and subactive mode (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

23.2.2 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPWRCR is an 8-bit readable/writable register that performs power-down mode control. The following pertains to bits 1 and 0. For details of the other bits, see Section 24.2.3, Low Power Control Register (LPWRCR). LPWRCR is initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 1 and 0—Frequency Multiplication Factor (STC1, STC0): The STC bits specify the frequency multiplication factor of the PLL circuit.

Bit 1	Bit 0		
STC1	**STC0**	**Description**	
0	0	×1	(Initial value)
	1	×2	
1	0	×4	
	1	Setting prohibited	

Note: A system clock frequency multiplied by the multiplication factor (STC1 and STC0) should not exceed the maximum operating frequency defined in section 25 Electrical Characteristics.

23.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

23.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 23-2. Select the damping resistance R_d according to table 23-2. An AT-cut parallel-resonance crystal should be used.

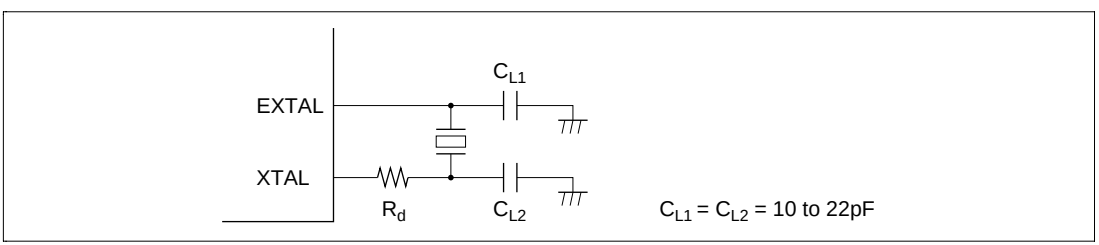


Figure 23-2 Connection of Crystal Resonator (Example)

Table 23-2 Damping Resistance Value

Frequency (MHz)	2	4	8	12	16	20	25
R_d (Ω)	1 k	500	200	0	0	0	0

Crystal Resonator: Figure 23-3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 23-3.

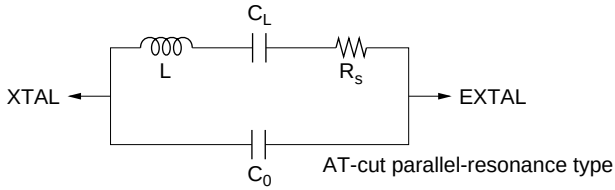


Figure 23-3 Crystal Resonator Equivalent Circuit

Table 23-3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	12	16	20	25
R_S max (Ω)	500	120	80	60	50	40	40
C_0 max (pF)	7	7	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 23-4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

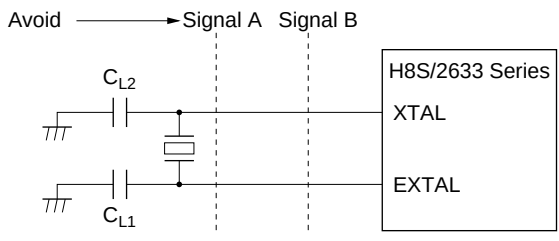


Figure 23-4 Example of Incorrect Board Design

External circuitry such as that shown below is recommended around the PLL.

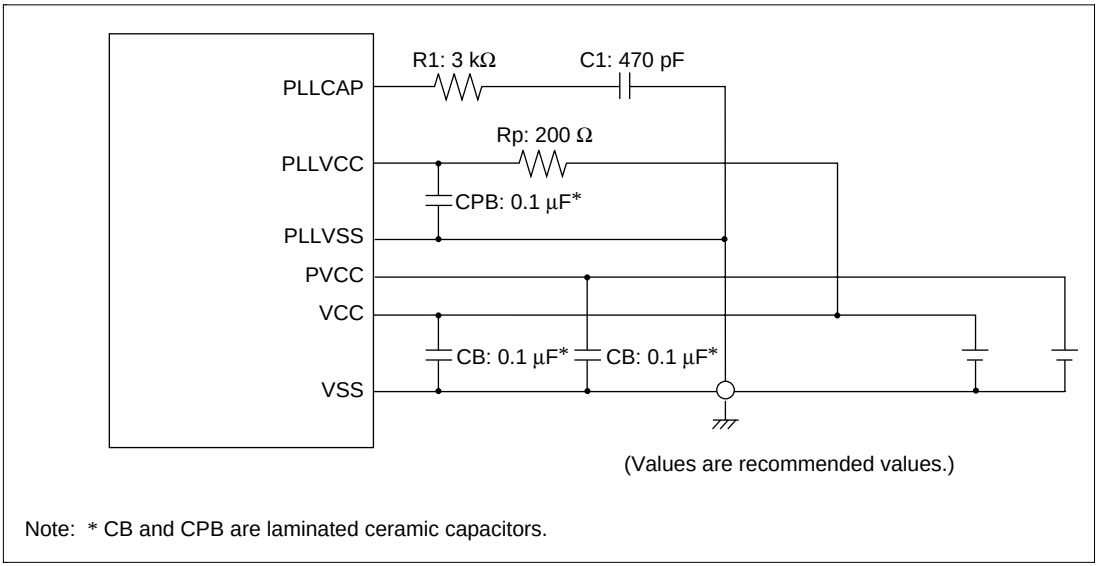


Figure 23-5 Points for Attention when Using PLL Oscillation Circuit

Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Supply the C1 ground from PLLVSS.

Separate PLLVCC and PLLVSS from the other VCC and VSS lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.

23.3.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 23-6. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode.

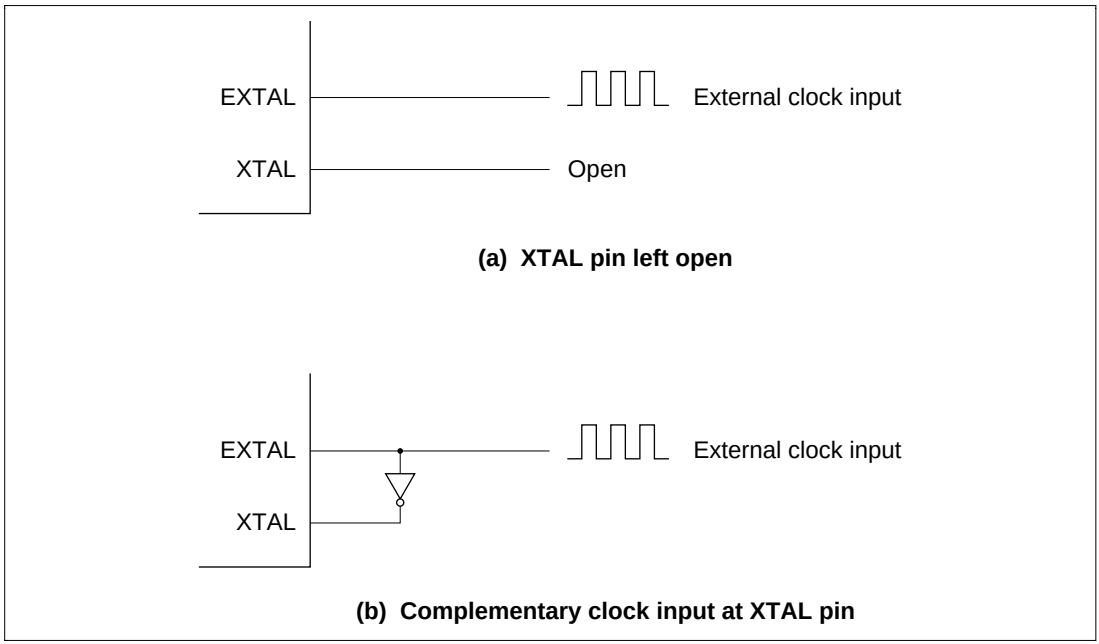


Figure 23-6 External Clock Input (Examples)

External Clock

Table 23-4 and figure 23-7 show the input conditions for the external clock.

Table 23-4 External Clock Input Conditions

Item	Symbol	V_{cc} = 3.0 V to 3.6 V, PV_{cc} = 3.0 V to 5.5 V		V_{cc} = 3.0 V to 3.6 V PV_{cc} = 5.0 V ±10%		Unit	Test Conditions	
		Min	Max	Min	Max			
External clock input low pulse width	t_{EXL}	20	—	15	—	ns	Figure 23-7	
External clock input high pulse width	t_{EXH}	20	—	15	—	ns		
External clock rise time	t_{EXr}	—	10	—	5	ns		
External clock fall time	t_{EXf}	—	10	—	5	ns		
Clock low pulse width level	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	Figure 25-2
		80	—	80	—	ns	ø < 5 MHz	
Clock high pulse width level	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	
		80	—	80	—	ns	ø < 5 MHz	

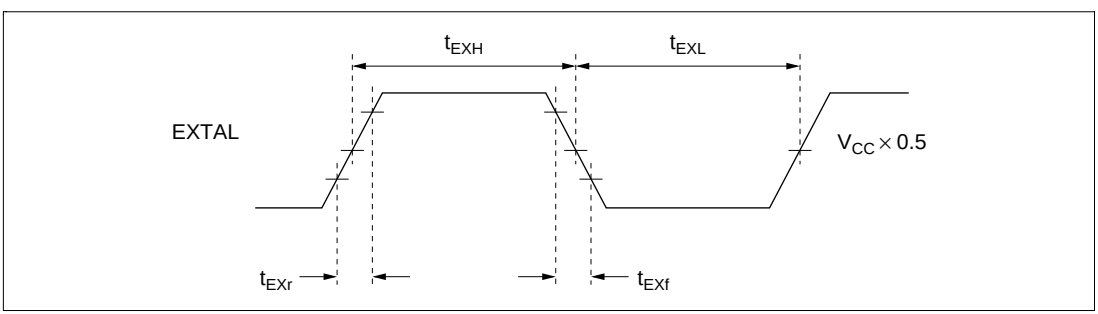


Figure 23-7 External Clock Input Timing

23.4 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC bits in LPWRCR. The phase of the rising edge of the internal clock is controlled so as to match that at the EXTAL pin. When setting the multiplication factor, ensure that the clock frequency after multiplication does not exceed the maximum operating frequency of the chip.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0 (initial value), the setting becomes valid after a transition to software standby mode, watch mode, or subactive mode. The transition time count is performed in accordance with the setting of bits STS2 to STS0 in SBYCR.

[1] The initial PLL circuit multiplication factor is 1.

[2] A value is set in bits STS2 to STS0 to give the specified transition time.

[3] The target value is set in STC1 and STC0, and a transition is made to software standby mode, watch mode, or subactive mode.

[4] The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.

[5] Software standby mode, watch mode, or subactive mode is cleared, and a transition time is secured in accordance with the setting in STS2 to STS0.

[6] After the set transition time has elapsed, the LSI resumes operation using the target multiplication factor.

If a PC break is set for the SLEEP instruction that causes a transition to software standby mode in [3], software standby mode is entered and break exception handling is executed after the oscillation stabilization time. In this case, the instruction following the SLEEP instruction is executed after execution of the RTE instruction.

When STCS = 1, the LSI operates on the changed multiplication factor immediately after bits STC1 and STC0 are rewritten.

23.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ø/2, ø/4, ø/8, ø/16, and ø/32.

23.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ø) or one of the medium-speed clocks (ø/2, ø/4, ø/8, ø/16, and ø/32) to be supplied to the bus master, according to the settings of the SCK2 to SCK0 bits in SCKCR.

23.7 Subclock Oscillator

(1) Connecting 32.768kHz Quartz Oscillator

To supply a clock to the subclock oscillator, connect a 32.768kHz quartz oscillator, as shown in Figure 23-8. See Section 23.3.1, Notes on Board Design for notes on connecting crystal oscillators.

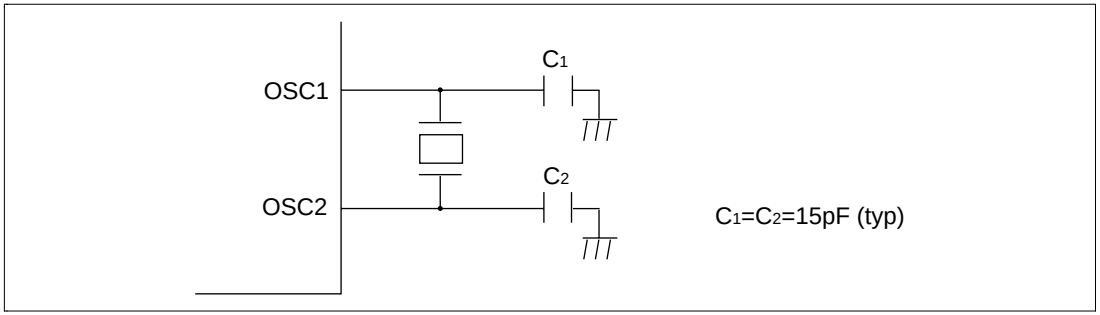


Figure 23-8 Example Connection of 32.768kHz Crystal Oscillator

Figure 23-9 shows the equivalence circuit for a 32.768kHz oscillator.

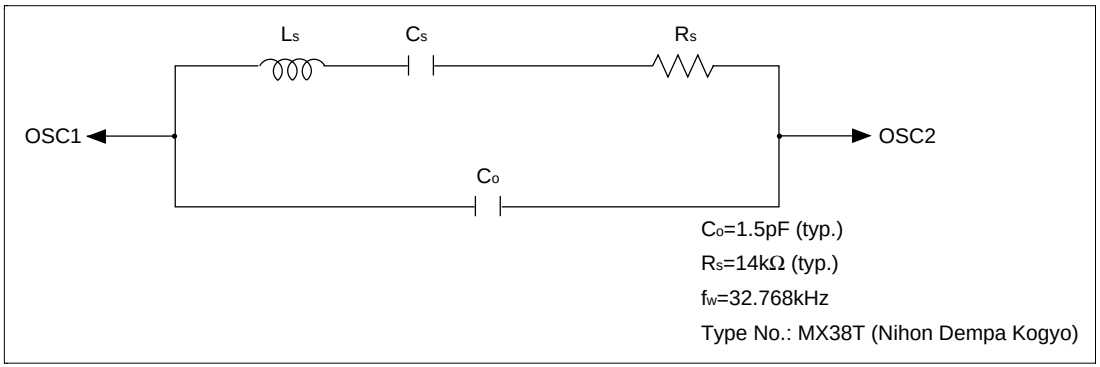


Figure 23-9 Equivalence Circuit for 32.768kHz Oscillator

(2) Handling pins when subclock not required

If no subclock is required, connect the OSC1 pin to Vcc and leave OSC2 open, as shown in Figure 23-10.

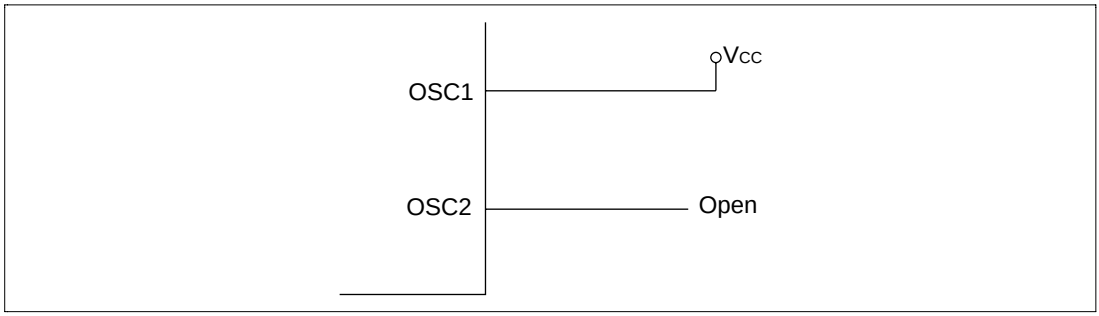


Figure 23-10 Pin Handling When Subclock Not Required

23.8 Subclock Waveform Shaping Circuit

To eliminate noise from the subclock input to OSCI, the subclock is sampled using the dividing clock ø. The sampling frequency is set using the NESEL bit of LPWRCR. For details, see Section 24.2.3, Low Power Control Register (LPWRCR).

No sampling is performed in sub-active mode, sub-sleep mode, or watch mode.

23.9 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, for both the mask versions and F-ZTAT versions, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

Section 24 Power-Down Modes

24.1 Overview

In addition to the normal program execution state, the H8S/2633 Series has eight power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The H8S/2633 Series operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Subactive mode
(4) Sleep mode
(5) Subsleep mode
(6) Watch mode
(7) Module stop mode
(8) Software standby mode
(9) Hardware standby mode

(2) to (9) are power down modes. Sleep mode and sub-sleep mode are CPU mode, medium-speed mode is a CPU and bus master mode, sub-active mode is a CPU and bus master and on-chip supporting module mode, and module stop mode is an on-chip supporting module mode (including bus masters other than the CPU) state. Some of these modes can be combined.

After a reset, the LSI is in high-speed mode, with modules other than the DMAC and DTC in module stop mode.

Table 24-1 shows the internal states of the LSI in the respective modes. Table 24-2 shows the conditions for shifting between the power-down modes.

Figure 24-1 is a mode transition diagram.

Table 24-1 LSI Internal States in Each Mode

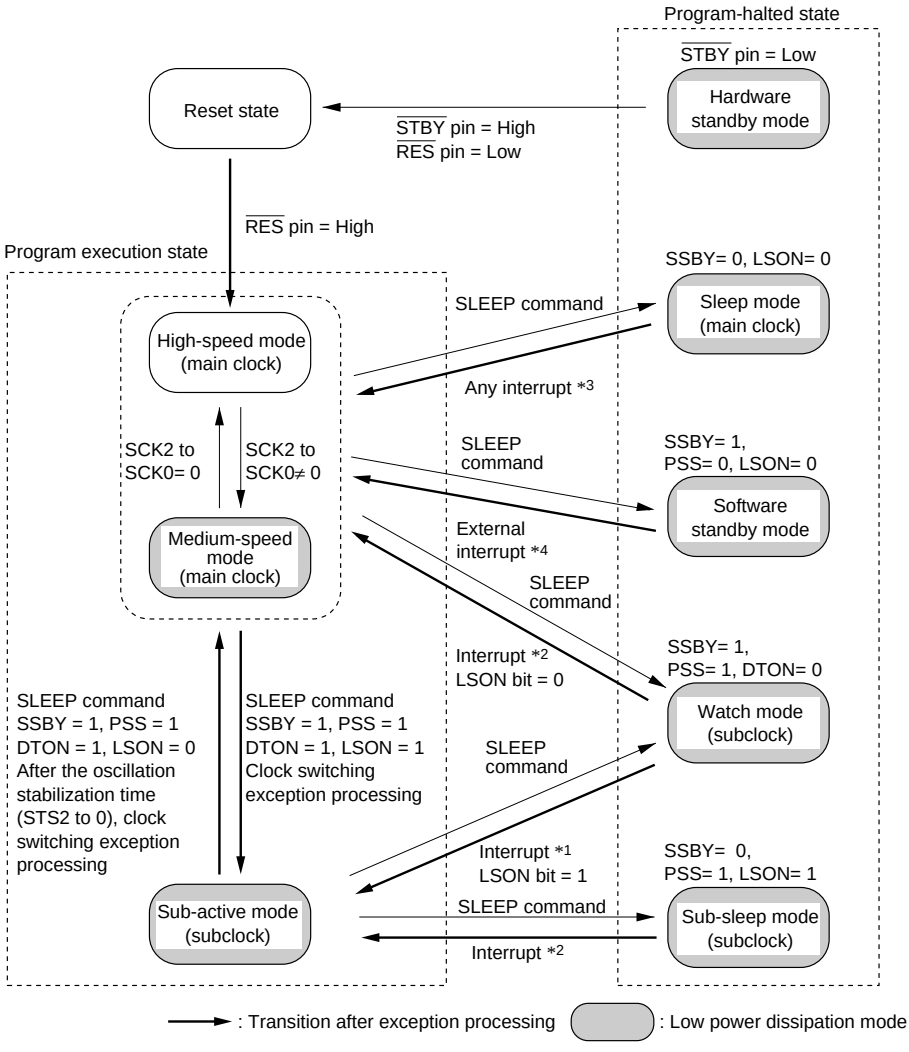
Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Subsleep	Software Standby	Hardware Standby
System clock pulse generator		Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted	Halted	Halted	Halted
Subclock pulse generator		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
CPU	Instructions Registers	Function-ing	Medium-speed operation	Halted (retained)	High/ medium-speed operation	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (undefined)
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
	IRQ0–IRQ7									
Peripheral functions	WDT1	Function-ing	Function-ing	Function-ing	Function-ing	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT0	Function-ing	Function-ing	Function-ing	Function-ing	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	TMR				Halted (retained)					
	DMAC	Function-ing	Medium-speed operation	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	DTC									
	TPU	Function-ing	Function-ing	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	IIC0									
	IIC1									
	PCB									
	PPG									
	D/A0, 1									
	SCI0	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1									
	SCI2									
	SCI3									
	SCI4									
	PWM0, 1									
	A/D									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing*	Retained	Retained	High impedance

Note: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

*: With the exception of ports D and E, an I/O port always returns a value of 1 when read in the H8S/2633F-ZTAT. Use as an output port is possible.



Notes: 1. NMI, IRQ0 to IRQ7, and WDT1 interrupts
2. NMI, IRQ0 to IRQ7, IWDT0 interrupts, WDT1 interrupt, and TMR0 to TMR3 interrupts
3. All interrupts
4. NMI and IRQ0 to IRQ7
   - When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
   - From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven Low.
   - From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
   - Always select high-speed mode before making a transition to watch mode or sub-active mode.

Figure 24-1 Mode Transition Diagram

Table 24.2 Power-Down Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition				State After Transition Invoked by SLEEP Command	State After Transition Back from Low Power Mode Invoked by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ Medium-speed	0	*	0	*	Sleep	High-speed/Medium-speed
	0	*	1	*	—	—
	1	0	0	*	Software standby	High-speed/Medium-speed
	1	0	1	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	—	—
	1	1	1	1	Sub-active	—
Sub-active	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	Sub-sleep	Sub-active
	1	0	*	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

* : Don’t care

—: Do not set.

24.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, LPWRCR, TCSR (WDT1), and MSTPCR registers. Table 24-3 summarizes these registers.

Table 24-3 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'08	H'FDE4
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC
Timer control/status register	TCSR	R/W	H'00	H'FFA2
Module stop control register A, B, C	MSTPCRA	R/W	H'3F	H'FDE8
	MSTPCRB	R/W	H'FF	H'FDE9
	MSTPCRC	R/W	H'FF	H'FDEA

Note: * Lower 16 bits of the address.

24.2 Register Descriptions

24.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Software Standby (SSBY): When making a low power dissipation mode transition by executing the SLEEP instruction, the operating mode is determined in combination with other control bits.

Note that the value of the SSBY bit does not change even when shifting between modes using interrupts.

Bit 7

SSBY	Description
0	Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts to sub-sleep mode when the SLEEP instruction is executed in sub-active mode. (Initial value)
1	Shifts to software standby mode, sub-active mode, and watch mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts to watch mode or high-speed mode when the SLEEP instruction is executed in sub-active mode.

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the MCU wait time for clock stabilization when shifting to high-speed mode or medium-speed mode by using a specific interrupt or command to cancel software standby mode, watch mode, or sub-active mode. With a crystal oscillator (Table 24-5), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, there are no specific wait requirements.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	(Initial value)
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	
	1	0	Reserved	
		1	Standby time = 16 states	

Bit 3—Output Port Enable (OPE): This bit specifies whether the output of the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{CAS}$, $\overline{OE}$) is retained or set to high-impedance state in the software standby mode, watch mode, and when making a direct transition.

Bit 3	
OPE	**Description**
0	In software standby mode, watch mode, and when making a direct transition, address bus and bus control signals are high-impedance.
1	In software standby mode, watch mode, and when making a direct transition, the output state of the address bus and bus control signals is retained. (Initial value)

Bits 2 to 0—Reserved: These bits are always read as 0 and cannot be modified.

24.2.2 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): In combination with the DDR of the applicable port, this bit controls ø output. See Section 24.12, ø Clock Output Disabling Function for details.

	Description			
Bit 7 PSTOP	**High-Speed Mode, Medium-Speed Mode, Sub-Active Mode**	**Sleep Mode, Sub-Sleep Mode**	**Software Standby Mode, Watch Mode**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 and 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3 STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode, watch mode, or subactive mode (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Bits 2 to 0—System clock select (SCK2 to SCK0): These bits select the bus master clock in high-speed mode, medium-speed mode, and sub-active mode.

Set SCK2 to SCK0 all to 0 when shifting to operation in watch mode or sub-active mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

24.2.3 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The LPWRCR is an 8-bit read/write register that controls the low power dissipation modes.

The LPWRCR is initialized to H'00 at a power-on reset and when in hardware standby mode. It is not initialized at a manual reset or when in software standby mode. The following describes bits 7 to 2. For details of other bits, see Section 23.2.2, Low-Power Control Register.

Bit 7—Direct Transition ON Flag (DTON): When shifting to low power dissipation mode by executing the SLEEP instruction, this bit specifies whether or not to make a direct transition between high-speed mode or medium-speed mode and the sub-active modes. The selected operating mode after executing the SLEEP instruction is determined by the combination of other control bits.

Bit 7

DTON	Description
0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode. (Initial value)
1	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts directly to sub-active mode*, or shifts to sleep mode or software standby mode. • When the SLEEP instruction is executed in sub-active mode, operation shifts directly to high-speed mode, or shifts to sub-sleep mode.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

Bit 6—Low-Speed ON Flag (LSON): When shifting to low power dissipation mode by executing the SLEEP instruction, this bit specifies the operating mode, in combination with other control bits. This bit also controls whether to shift to high-speed mode or sub-active mode when watch mode is cancelled.

Bit 6

LSON	Description
0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. • When the SLEEP instruction is executed in sub-active mode, operation shifts to watch mode or shifts directly to high-speed mode. • Operation shifts to high-speed mode when watch mode is cancelled. (Initial value)
1	• When the SLEEP instruction is executed in high-speed mode, operation shifts to watch mode or sub-active mode. • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode. • Operation shifts to sub-active mode when watch mode is cancelled.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

Bit 5—Noise Elimination Sampling Frequency Select (NESEL): This bit selects the sampling frequency of the subclock (øSUB) generated by the subclock oscillator is sampled by the clock (ø) generated by the system clock oscillator. Set this bit to 0 when ø=5MHz or more.

Bit 5

NESEL	Description	
0	Sampling using 1/32 xø	(Initial value)
1	Sampling using 1/4 xø	

Bit 4—Subclock enable (SUBSTP): This bit enables/disables subclock generation.

Bit 4

SUBSTP	Description	
0	Enables subclock generation	(Initial value)
1	Disables subclock generation	

Bit 3—Oscillation Circuit Feedback Resistance Control Bit (RFCUT): This bit turns the internal feedback resistance of the main clock oscillation circuit ON/OFF.

Bit 3

RFCUT	Description	
0	When the main clock is oscillating, sets the feedback resistance ON. When the main clock is stopped, sets the feedback resistance OFF.	(Initial value)
1	Sets the feedback resistance OFF.	

Bit 2—Reserved: Should always be written with 0.

24.2.4 Timer Control/Status Register (TCSR)

WDT1 TCSR

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only write 0 to clear the flag.

TCSR is an 8-bit read/write register that selects the clock input to WDT1 TCNT and the mode.

The following describes bit 4. For details of the other bits in this register, see Section 15.2.2, Timer Control/Status Register (TCSR).

The TCSR is initialized to H'00 at a reset and when in hardware standby mode. It is not initialized in software standby mode.

Bit 4—Prescaler select (PSS): This bit selects the clock source input to WDT1 TCNT.

It also controls operation when shifting low power dissipation modes. The operating mode selected after the SLEEP instruction is executed is determined in combination with other control bits.

For details, see the description for clock selection in Section 15.2.2, Timer Control/Status Register (TCSR), and this section.

Bit 4

PSS	Description
0	• TCNT counts the divided clock from the ø -based prescaler (PSM). • When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode or software standby mode. (Initial value)
1	• TCNT counts the divided clock from the øsubclock-based prescaler (PSS). • When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, watch mode*, or sub-active mode*. • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode, watch mode, or high-speed mode.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

24.2.5 Module Stop Control Register (MSTPCR)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	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

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	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

MSTPCR, comprising three 8-bit readable/writable registers, performs module stop mode control.

MSTPCR is initialized to H'3FFFFF by a reset and in hardware standby mode. It is not initialized in software standby mode.

MSTPCRA/MSTPCRB/MSTPCRC Bits 7 to 0—Module Stop (MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0): These bits specify module stop mode. See table 24-3 for the method of selecting the on-chip peripheral functions.

MSTPCRA/MSTPCRB/ MSTPCRC Bits 7 to 0	
MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0	**Description**
0	Module stop mode is cleared (initial value of MSTPA7 and MSTPA6)
1	Module stop mode is set (initial value of MSTPA5 to 0, MSTPB7 to 0, and MSTPC7 to 0)

24.3 Medium-Speed Mode

In high-speed mode, when the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (ø/2, ø/4, ø/8, ø/16, or ø/32) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (the DMAC and DTC) also operate in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ø).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, and LSON bit in LPWRCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LPWRCR LSON bit = 0, and TCSR (WDT1) PSS bit = 0, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pins are set Low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 24-2 shows the timing for transition to and clearance of medium-speed mode.

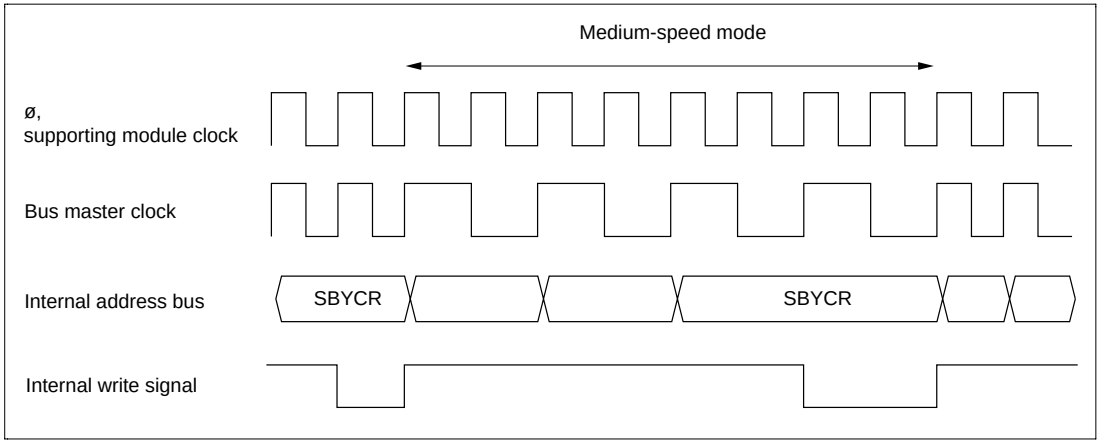


Figure 24-2 Medium-Speed Mode Transition and Clearance Timing

24.4 Sleep Mode

24.4.1 Sleep Mode

When the SLEEP instruction is executed when the SBYCR SSBY bit = 0 and the LPWRCR LSON bit = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPUís internal registers are retained. Other supporting modules do not stop.

24.4.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pins.

(1) Exiting Sleep Mode by Interrupts

When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.

(2) Exiting Sleep Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pins

Setting the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin level Low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pins High starts the CPU performing reset exception processing.

(3) Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

24.5 Module Stop Mode

24.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 24-4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, A/D converter and 14-bit PWM are retained.

After reset clearance, all modules other than DMAC and DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Table 24-4 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRA	MSTPA7	DMA controller (DMAC)
	MSTPA6	Data transfer controller (DTC)
	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA4	8-bit timer (TMR0, TMR1)
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2	D/A converter (channels 0, 1)
	MSTPA1	A/D converter
	MSTPA0	8-bit timer (TMR2, TMR3)
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4	I^2C bus interface 0 (IIC0)
	MSTPB3	I^2C bus interface 1 (IIC1)
	MSTPB2	14-bit PWM timer (PWM0)
	MSTPB1	14-bit PWM timer (PWM1)
	MSTPB0*	—
MSTPCRC	MSTPC7	Serial communication interface 3 (SCI3)
	MSTPC6	Serial communication interface 4 (SCI4)
	MSTPC5	D/A converter (channels 2, 3)
	MSTPC4	PC break controller (PBC)
	MSTPC3*	—
	MSTPC2*	—
	MSTPC1*	—
	MSTPC0*	—

Note: * Write 1 to bit MSTPB0 and bits MTSPC3 to MSTPC0.

24.5.2 Usage Notes

DMAC and DTC Module Stop: Depending on the operating status of the DMAC and DTC, the MSTPA7 and MSTPA6 bits may not be set to 1. Setting of the DMAC or DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, DMA Controller and section 9, Data Transfer Controller (DTC).

On-Chip Supporting Module Interrupt: Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC and DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

24.6 Software Standby Mode

24.6.1 Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed when the SBYCR SSBY bit = 1 and the LPWRCR LSON bit = 0, and the TCSR (WDT1) PSS bit = 0. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI, A/D converter, and 14-bit PWM, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

24.6.2 Exiting Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or by means of the $\overline{\text{RES}}$ pin, $\overline{\text{MRES}}$ pin or $\overline{\text{STBY}}$ pin.

(1) Exiting Software Standby Mode with an Interrupt

When an NMI or IRQ0 to IRQ7 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are supplied to the entire chip, software standby mode is exited, and interrupt exception handling is started.

When exiting software standby mode with an IRQ0 to IRQ7 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ7 is generated. Software standby mode cannot be exited if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

(2) Exiting Software Standby Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pins

When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin goes high, the CPU begins reset exception handling.

(3) Exiting Software Standby Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

24.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation stabilization time).

Table 24-5 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 24-5 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	25 MHz	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.32	0.41	0.51	0.65	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.65	0.82	1.0	1.3	1.6	2.0	2.7	4.1	[8.2]	
	1	0	32768 states	1.3	1.6	2.0	2.7	3.3	4.1	5.5	[8.2]	16.4	
		1	65536 states	2.6	3.3	4.1	5.5	6.6	[8.2]	[10.9]	16.4	32.8	
1	0	0	131072 states	5.2	6.6	[8.2]	[10.9]	[13.1]	16.4	21.8	32.8	65.5	
		1	262144 states	[10.4]	[13.1]	16.4	21.8	26.2	32.8	43.6	65.6	131.2	
	1	0	Reserved	—	—	—	—	—	—	—	—	—	µs
		1	16 states*	0.6	0.8	1.0	1.3	1.6	2.0	1.7	4.0	8.0	

[ ]: Recommended time setting

Note: * Do not use this setting in the version with built-in flash memory.

Using an External Clock: The PLL circuit requires a time for stabilization. Insert a wait of 2 ms min.

24.6.4 Software Standby Mode Application Example

Figure 24-3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

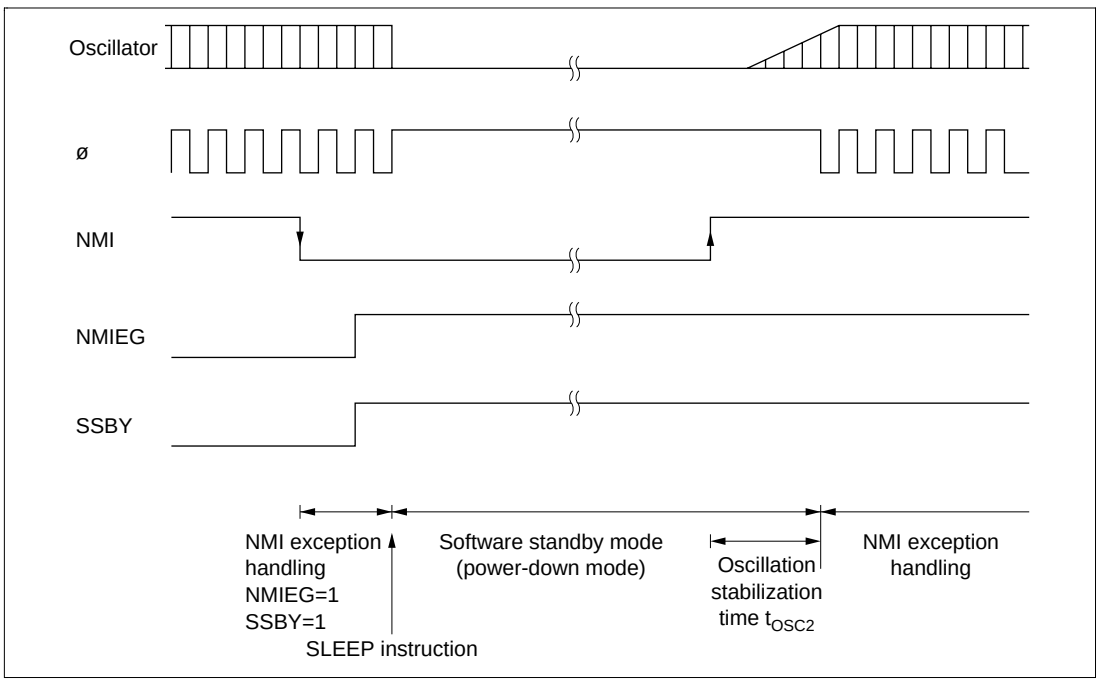


Figure 24-3 Software Standby Mode Application Example

24.6.5 Usage Notes

I/O Port Status: In software standby mode, I/O port states are retained. If the OPE bit is set to 1, the address bus and bus control signal output is also retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current Dissipation during Oscillation Stabilization Wait Period: Current dissipation increases during the oscillation stabilization wait period.

Write Data Buffer Function: The write data buffer function and software standby mode cannot be used at the same time. When the write data buffer function is used, the WDBE bit in BCRL should be cleared to 0 to cancel the write data buffer function before entering software standby mode. Also check that external writes have finished, by reading external addresses, etc., before executing a SLEEP instruction to enter software standby mode. See section 7.9, Write Data Buffer Function, for details of the write data buffer function.

24.7 Hardware Standby Mode

24.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the H8S/2633 Series is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—the oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

24.7.2 Hardware Standby Mode Timing

Figure 24-4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

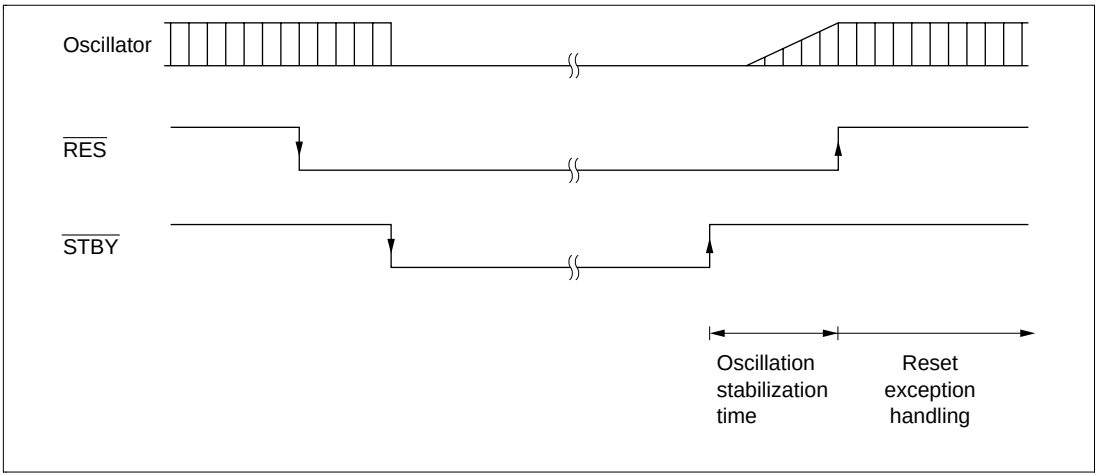


Figure 24-4 Hardware Standby Mode Timing

24.8 Watch Mode

24.8.1 Watch Mode

CPU operation makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or sub-active mode with SBYCR SSBY=1, LPWRCR DTON = 0, and TCSR (WDT1) PSS = 1.

In watch mode, the CPU is stopped and supporting modules other than WDT1 are also stopped. The contents of the CPUís internal registers, the data in internal RAM, and the statuses of the internal supporting modules (excluding the SCI, ADC, and 14-bit PWM) and I/O ports are retained.

24.8.2 Exiting Watch Mode

Watch mode is exited by any interrupt (WOVI1 interrupt, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pins.

(1) Exiting Watch Mode by Interrupts

When an interrupt occurs, watch mode is exited and a transition is made to high-speed mode or medium-speed mode when the LPWRCR LSON bit = 0 or to sub-active mode when the LSON bit = 1. When a transition is made to high-speed mode, a stable clock is supplied to all LSI circuits and interrupt exception processing starts after the time set in SBYCR STS2 to STS0 has elapsed. In the case of IRQ0 to IRQ7 interrupts, no transition is made from watch mode if the corresponding enable bit has been cleared to 0, and, in the case of interrupts from the internal supporting modules, the interrupt enable register has been set to disable the reception of that interrupt, or is masked by the CPU.

See Section 24.6.3, Setting Oscillation Stabilization Time After Clearing Software Standby Mode for how to set the oscillation stabilization time when making a transition from watch mode to high-speed mode.

(2) Exiting Watch Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pins

For exiting watch mode by the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins, see (2), Exiting Software Standby Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins in Section 24.6.2, Exiting Software Standby Mode.

(3) Exiting Watch Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

24.8.3 Notes

(1) I/O Port Status

The status of the I/O ports is retained in watch mode. Also, when the OPE bit is set to 1, the address bus and bus control signals continue to be output. Therefore, when a High level is output, the current consumption is not diminished by the amount of current to support the High level output.

(2) Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during stabilization of oscillation.

24.9 Sub-Sleep Mode

24.9.1 Sub-Sleep Mode

When the SLEEP instruction is executed with the SBYCR SSBY bit = 0, LPWRCR LSON bit = 1, and TCSR (WDT1) PSS bit = 1, CPU operation shifts to sub-sleep mode.

In sub-sleep mode, the CPU is stopped. Supporting modules other than TMR0 to TMR3, WDT0, and WDT1 are also stopped. The contents of the CPUís internal registers, the data in internal RAM, and the statuses of the internal supporting modules (excluding the SCI, ADC, and 14-bit PWM) and I/O ports are retained.

24.9.2 Exiting Sub-Sleep Mode

Sub-sleep mode is exited by an interrupt (interrupts from internal supporting modules, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Sleep Mode by Interrupts

When an interrupt occurs, sub-sleep mode is exited and interrupt exception processing starts.

In the case of IRQ0 to IRQ7 interrupts, sub-sleep mode is not cancelled if the corresponding enable bit has been cleared to 0, and, in the case of interrupts from the internal supporting modules, the interrupt enable register has been set to disable the reception of that interrupt, or is masked by the CPU.

(2) Exiting Sub-Sleep Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pins

For exiting sub-sleep mode by the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins, see (2), Exiting Software Standby Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins in Section 24.6.2, Exiting Software Standby Mode.

(3) Exiting Sub-Sleep Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

24.10 Sub-Active Mode

24.10.1 Sub-Active Mode

When the SLEEP instruction is executed in high-speed mode with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 1, LSON bit = 1, and TCSR (WDT1) PSS bit = 1, CPU operation shifts to sub-active mode. When an interrupt occurs in watch mode, and if the LSON bit of LPWRCR is 1, a transition is made to sub-active mode. And if an interrupt occurs in sub-sleep mode, a transition is made to sub-active mode.

In sub-active mode, the CPU operates at low speed on the subclock, and the program is executed step by step. Supporting modules other than TMR0 to TMR3, WDT0, and WDT1 are also stopped.

When operating the CPU in sub-active mode, the SCKCR SCK2 to SCK0 bits must be set to 0.

24.10.2 Exiting Sub-Active Mode

Sub-active mode is exited by the SLEEP instruction or the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Active Mode by SLEEP Instruction

When the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 0, and TCSR (WDT1) PSS bit = 1, the CPU exits sub-active mode and a transition is made to watch mode. When the SLEEP instruction is executed with the SBYCR SSBY bit = 0, LPWRCR LSON bit = 1, and TCSR (WDT1) PSS bit = 1, a transition is made to sub-sleep mode. Finally, when the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 1, LSON bit = 0, and TCSR (WDT1) PSS bit = 1, a direct transition is made to high-speed mode (SCK0 to SCK2 all 0).

See Section 24.11, Direct Transitions for details of direct transitions.

(2) Exiting Sub-Active Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pins

For exiting sub-active mode by the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins, see (2), Exiting Software Standby Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pins in Section 24.6.2, Exiting Software Standby Mode.

(3) Exiting Sub-Active Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

24.11 Direct Transitions

24.11.1 Overview of Direct Transitions

There are three modes, high-speed, medium-speed, and sub-active, in which the CPU executes programs. When a direct transition is made, there is no interruption of program execution when shifting between high-speed and sub-active modes. Direct transitions are enabled by setting the LPWRCR DTON bit to 1, then executing the SLEEP instruction. After a transition, direct transition interrupt exception processing starts.

(1) Direct Transitions from High-Speed Mode to Sub-Active Mode

Execute the SLEEP instruction in high-speed mode when the SBYCR SSBY bit = 1, LPWRCR LSON bit = 1, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a transition to sub-active mode.

(2) Direct Transitions from Sub-Active Mode to High-Speed Mode

Execute the SLEEP instruction in sub-active mode when the SBYCR SSBY bit = 1, LPWRCR LSON bit = 0, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a direct transition to high-speed mode after the time set in SBYCR STS2 to STS0 has elapsed.

24.12 ø Clock Output Disabling Function

Output of the ø clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ø clock stops at the end of the bus cycle, and ø output goes high. ø clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ø clock output is disabled and input port mode is set. Table 24-6 shows the state of the ø pin in each processing state.

Table 24-6 ø Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby mode, watch mode, and direct transition	High impedance	Fixed high	Fixed high
Sleep mode and subsleep mode	High impedance	ø output	Fixed high
High-speed mode, medium-speed mode, and subactive mode	High impedance	ø output	Fixed high

25.1 Absolute Maximum Ratings

Table 25-1 lists the absolute maximum ratings. Unless specified otherwise, PV_{CC} refers to both $PV_{CC}1$ and $PV_{CC}2$.

Table 25-1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} $PLLV_{CC}$	–0.3 to +4.3	V
	$PV_{CC1,2}$	–0.3 to +7.0	V
Input voltage (XTAL, EXTAL, OSC1, OSC2)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (except XTAL, EXTAL, OSC1, OSC2, port 4 and 9)	V_{in}	–0.3 to PV_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

25.2 DC Characteristics

Table 25-2 lists the DC characteristics. Table 25-3 lists the permissible output currents.

Table 25-2 DC Characteristics (1)

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*[1]

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ7}$ to $\overline{IRQ0}$	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$PV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE*[5], MD2 to MD0	V_{IH}	PV_{CC} – 0.7	—	PV_{CC} + 0.3	V	
	EXTAL, OSC1		$V_{CC} \times 0.8$	—	V_{CC} + 0.3	V	
	Port 1, 3, 7, A toG		2.2	—	PV_{CC} + 0.3	V	
	Port 4 and 9		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE*[5], MD2 to MD0	V_{IL}	–0.3	—	0.5	V	
	EXTAL, OSC1		–0.3	—	$V_{CC} \times 0.2$	V	
	Port 1, 3, 4, 7, 9, A toG		–0.3	—	0.8	V	
Output high voltage	All output pins except P34 and P35	V_{OH}	PV_{CC} –0.5	—	—	V	I_{OH} = –200 µA
	P34, P35		PV_{CC} –2.5				I_{OH} = –100 µA
	All output pins except P34 and P35		3.5				I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$, FWE*[5]	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
	$\overline{STBY}$, NMI, MD2 to MD0		—	—	1.0	µA	
	Port 4, 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Port 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
MOS input pull-up current	Port A to E	$-I_P$	50	—	300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*[2]	Normal operation	I_{CC}*[4]	—	72 V_{CC} = 3.3 V	85 V_{CC} = 3.6 V	mA	f = 25 MHz
	Sleep mode		—	58 V_{CC} = 3.3 V	75 V_{CC} = 3.6 V	mA	f = 25 MHz
	All modules stopped		—	50	—	mA	f = 25 MHz, V_{CC} = 3.3 V (reference values)
	Medium-speed mode (ø/32)		—	40	—	mA	f = 25 MHz, V_{CC} = 3.3 V (reference values)
	Subactive mode		—	120 V_{CC} = 3.0 V T_a = 25 °C	200	µA	Using 32.768 kHz crystal resonator
	Subsleep mode		—	70 V_{CC} = 3.0 V T_a = 25 °C	150	µA	Using 32.768 kHz crystal resonator
	Watch mode		—	20 V_{CC} = 3.0 V T_a = 25 °C	50	µA	Using 32.768 kHz crystal resonator
	Standby mode*[3]		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20		50°C < T_a

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Port power supply current*[2]	Operating	PI_{CC}	—	17 PV_{CC} = 5.0 V	25	mA	
	Subclock operation		—	—	50	µA	
	Standby*[3]		—	0.01	5.0		$T_a \leq 50$ °C
	Watch mode		—	—	20		50 °C < T_a
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.6	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.01 T_a = 25 °C	5.0	µA	
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	4.0	5.0	mA	AV_{ref} = 5.0 V
	Idle		—	0.01 T_a = 25 °C	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage between 3.3 V and 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to PV_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

*2 Current dissipation values are for $V_{IH} = V_{CC}$ (EXTAL, OSC1), AV_{CC} (ports 4 and 9), or PV_{CC} (other), and V_{IL} = 0 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = V_{CC} – 0.1 V, and V_{IL} max = 0.1 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 1.0 (mA) + 0.93 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 1.0 (mA) + 0.77 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

*5 The FWE pin is used only in the flash memory version.

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*[1]

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ7}$ to $\overline{IRQ0}$	V_T^-	$PV_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$PV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$PV_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, FWE*[6], NMI, MD2 to MD0	V_{IH}	$PV_{CC} \times 0.9$	—	PV_{CC} + 0.3	V	
	EXTAL, OSC1		$V_{CC} \times 0.8$	—	V_{CC} + 0.3	V	
	Port 1, 3, 7, A to G		$PV_{CC} \times 0.8$	—	PV_{CC} + 0.3	V	
	Port 4 and 9		$AV_{CC} \times 0.8$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE*[6], MD2 to MD0	V_{IL}	–0.3	—	$PV_{CC} \times 0.1$	V	
	EXTAL, OSC1		–0.3	—	$V_{CC} \times 0.2$	V	
	Port 1, 3, 7, A to G		–0.3	—	$PV_{CC} \times 0.2$	V	
	Port 4 and 9		–0.3	—	$AV_{CC} \times 0.2$	V	
Output high voltage	All output pins except P34 and P35	V_{OH}	PV_{CC} –0.5	—	—	V	I_{OH} = –200 µA
	P34, P35		PV_{CC} –2.5	—	—		I_{OH} = –100 µA*[2]
	All output pins except P34 and P35		PV_{CC} –1.0		—		I_{OH} = –1mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$, FWE*[6]	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
	$\overline{STBY}$, NMI, MD2 to MD0		—	—	1.0	µA	
	Port 4, 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Port 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
MOS input pull-up current	Port A to E	$-I_P$	25	—	300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz Ta = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*[3]	Normal operation	I_{CC}*[5]	—	40 V_{CC} = 3.3 V	60 V_{CC} = 3.6 V	mA	f = 16 MHz
	Sleep mode		—	35 V_{CC} = 3.3 V	45 V_{CC} = 3.6 V	mA	f = 16 MHz
	All modules stopped		—	30	—	mA	f = 16 MHz, V_{CC} = 3.3 V (reference values)
	Medium-speed mode (ø/32)		—	25	—	mA	f = 16 MHz, V_{CC} = 3.3 V (reference values)
	Subactive mode		—	120 V_{CC} = 3.0 V T_a = 25 °C	200	µA	Using 32.768 kHz crystal resonator
	Subsleep mode		—	70 V_{CC} = 3.0 V T_a = 25 °C	150	µA	Using 32.768 kHz crystal resonator
	Watch mode		—	20 V_{CC} = 3.0 V T_a = 25 °C	50	µA	Using 32.768 kHz crystal resonator
	Standby mode*[4]		—	0.01	5.0	µA	$T_a \leq 50$°C
			—	—	20		50°C < T_a

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Port power supply current*3	Operating	PI_{CC}	—	10 PV_{CC} = 5.0 V	16	mA	
	Subclock operation		—	—	50	µA	
	Standby*4		—	0.01	5.0		$T_a \leq 50°C$
	Watch mode		—	—	20		50 °C < T_a
Analog power supply current	During A/D and D/A conversions	AI_{CC}	—	0.6	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.01 T_a = 25 °C	5.0	µA	
Reference current	During A/D and D/A conversions	AI_{CC}	—	4.0	5.0	mA	AV_{ref} = 5.0 V
	Idle		—	0.01 T_a = 25 °C	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 3.3 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to PV_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

*2 When using P34 and P35 as output pins, set PV_{CC} = 4.5 V to 5.5 V.

*3 Current dissipation values are for $V_{IH} = V_{CC}$ (EXTAL, OSC1), AV_{CC} (ports 4 and 9), or PV_{CC} (other), and V_{IL} = 0 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

*4 The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = V_{CC} – 0.1 V, and V_{IL} max = 0.1 V.

*5 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 1.0 (mA) + 0.93 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 1.0 (mA) + 0.77 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

*6 The FWE pin is used only in the flash memory version.

Table 25-3 Permissible Output Currents

Conditions: V_{CC} = PLVCC = 3.0 V to 3.6 V, PVCC = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*[1]

Item			Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	All output pins	PV_{CC} = 3.0 to 5.5 V	I_{OL}	—	—	10	mA
Permissible output low current (total)	Total of all output pins	PV_{CC} = 3.0 to 5.5 V	$\sum I_{OL}$	—	—	120	mA
Permissible output high current (per pin)	All output pins	PV_{CC} = 3.0 to 5.5 V	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	PV_{CC} = 3.0 to 5.5 V	$\sum -I_{OH}$	—	—	40	mA

Note: To protect chip reliability, do not exceed the output current values in table 25-3.

Table 25-4 Bus Drive Characteristics

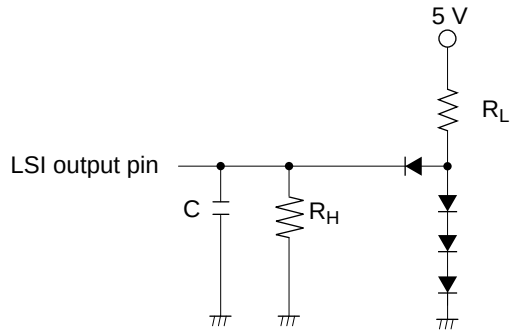
Condition : V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Applicable Pins: SCL1-0, SDA1-0

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	V_T-	$PV_{CC} \times 0.3$	—	—	V	
	V_T+	—	—	$PV_{CC} \times 0.7$		
	V_T+ - V_T-	0.4	—	—		PV_{CC} = 4.5 V to 5.5V
		0.2	—	—		PV_{CC} = 3.0 V to 4.5V
Input high voltage	V_{IH}	$PV_{CC} \times 0.7$	—	PV_{CC} + 0.5	V	
Input low voltage	V_{IL}	- 0.5	—	$PV_{CC} \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.7		I_{OL} = 8 mA, PV_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 3 mA, PV_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 1.6 mA, PV_{CC} = 3.0 V to 5.5 V
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0V, f = 1MHz, Ta = 25°C
Three - state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} =0.5 to V_{CC} - 0.5V
SCL, SDA, output fall time	t_{Of}	20 + 0.1 Cb	—	250	ns	

25.3 AC Characteristics

Figure 25-1 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, 70 to 73, A to G
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
· Low level : 0.8 V
· High level : 2.0 V

Figure 25-1 Output Load Circuit

25.3.1 Clock Timing

Table 25-5 lists the clock timing

Table 25-5 Clock Timing

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz, 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz, 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
		16MHz		25MHz			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	62.5	500	40	500	ns	Figure 25-2
Clock high pulse width	t_{CH}	18	—	15	—	ns	
Clock low pulse width	t_{CL}	18	—	15	—	ns	
Clock rise time	t_{Cr}	—	12	—	5	ns	
Clock fall time	t_{Cf}	—	12	—	5	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	10	—	10	—	ms	Figure 25-3
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	5	—	ms	Figure 24-3
External clock output stabilization delay time	t_{DEXT}	2	—	2	—	ms	Figure 25-3
32 kHz clock oscillation settling time	t_{OSC3}	—	2	—	2	s	
Sub clock oscillator frequency	f_{SUB}	32.768		32.768		kHz	
Sub clock ($ø_{SUB}$) cycle time	t_{SUB}	30.5		30.5		µs	

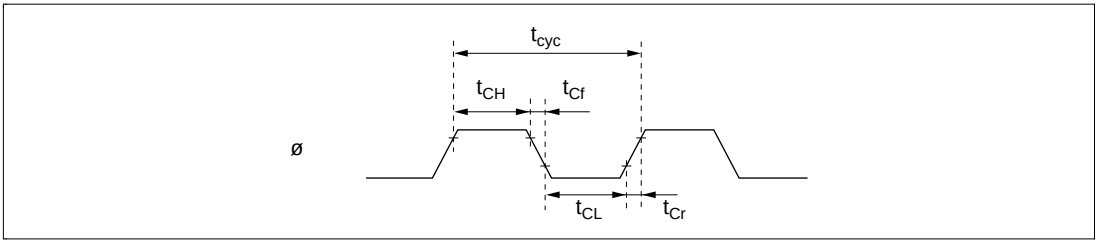


Figure 25-2 System Clock Timing

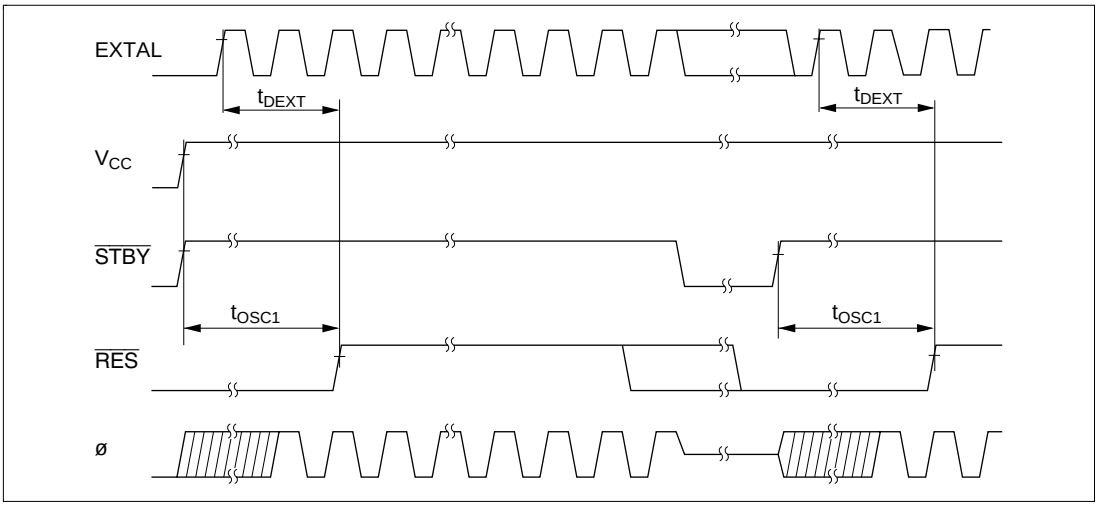


Figure 25-3 Oscillator Settling Timing

25.3.2 Control Signal Timing

Table 25-6 lists the control signal timing.

Table 25-6 Control Signal Timing

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz, 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz, 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	ns	Figure 25-4
$\overline{RES}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	150	—	ns	Figure 25-5
NMI hold time	t_{NMIH}	10	—	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	200	—	ns	

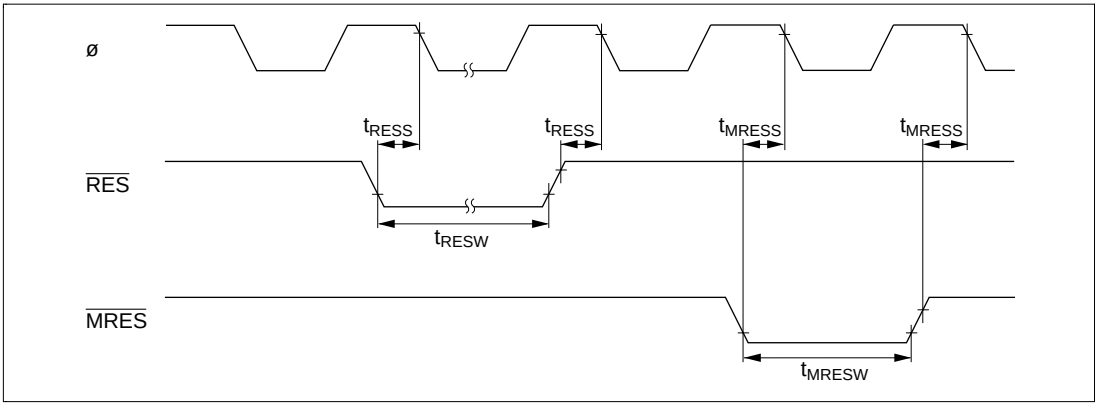


Figure 25-4 Reset Input Timing

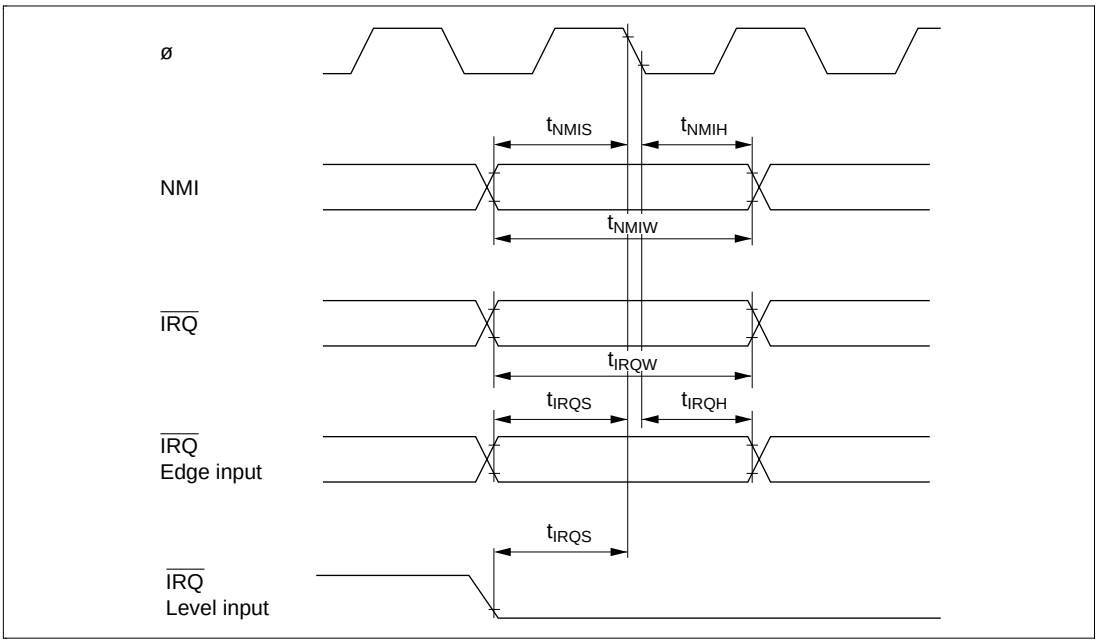


Figure 25-5 Interrupt Input Timing

25.3.3 Bus Timing

Table 25-7 lists the bus timing.

Table 25-7 Bus Timing

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	30	—	20	ns	Figure 25-6 to
Address setup time	t_{AS}	0.5 × t_{cyc} – 30	—	0.5 × t_{cyc} – 15	—	ns	Figure 25-11
Address hold time	t_{AH}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	30	—	20	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	30	—	18	ns	
$\overline{AS}$ delay time	t_{ASD}	—	30	—	18	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	30	—	18	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	30	—	18	ns	
Read data setup time	t_{RDS}	30	—	15	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Read data access time1	t_{ACC1}	—	1.0 × t_{cyc} – 35	—	1.0 × t_{cyc} – 25	ns	
Read data access time2	t_{ACC2}	—	1.5 × t_{cyc} – 35	—	1.5 × t_{cyc} – 25	ns	
Read data access time3	t_{ACC3}	—	2.0 × t_{cyc} – 35	—	2.0 × t_{cyc} – 25	ns	
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 35	—	2.5 × t_{cyc} – 25	ns	

Item	Symbol	Condition A		Condition B		Unit	Test Conditions
		Min	Max	Min	Max		
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 35	—	3.0 × t_{cyc} – 25	ns	Figure 25-6 to Figure 25-11
$\overline{WR}$ delay time 1	t_{WRD1}	—	30	—	18	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	30	—	18	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} – 30	—	1.0 × t_{cyc} – 15	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} – 30	—	1.5 × t_{cyc} – 15	—	ns	
Write data delay time	t_{WDD}	—	30	—	22	ns	
Write data setup time	t_{WDS}	0.5 × t_{cyc} – 27	—	0.5 × t_{cyc} – 15	—	ns	
Write data hold time	t_{WDH}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{WR}$ setup time	t_{WCS}	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{WR}$ hold time	t_{WCH}	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{RAS}$ precharge time	t_{PCH}	1.5 × t_{cyc} – 30	—	1.5 × t_{cyc} – 15	—	ns	Figure 25-11 to Figure 25-13
$\overline{CAS}$ precharge time1	t_{CP1}	1.0 × t_{cyc} – 20	—	1.0 × t_{cyc} – 8	—	ns	
$\overline{CAS}$ precharge time2	t_{CP2}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{CAS}$ delay time1	t_{CASD1}	—	30	—	20	ns	
$\overline{CAS}$ delay time2	t_{CASD2}	—	30	—	18	ns	
$\overline{OE}$ delay time1	t_{OED1}	—	30	—	18	ns	
$\overline{OE}$ delay time2	t_{OED2}	—	30	—	18	ns	
$\overline{CAS}$ setup time	t_{CSR}	0.5 × t_{cyc} – 25	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	40	—	25	—	ns	Figure 25-8
$\overline{WAIT}$ hold time	t_{WTH}	10	—	5	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	60	—	30	—	ns	Figure 25-14
$\overline{BACK}$ delay time	t_{BACD}	—	30	—	15	ns	
Bus-floating time	t_{BZD}	—	60	—	40	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	40	—	25	ns	Figure 25-15

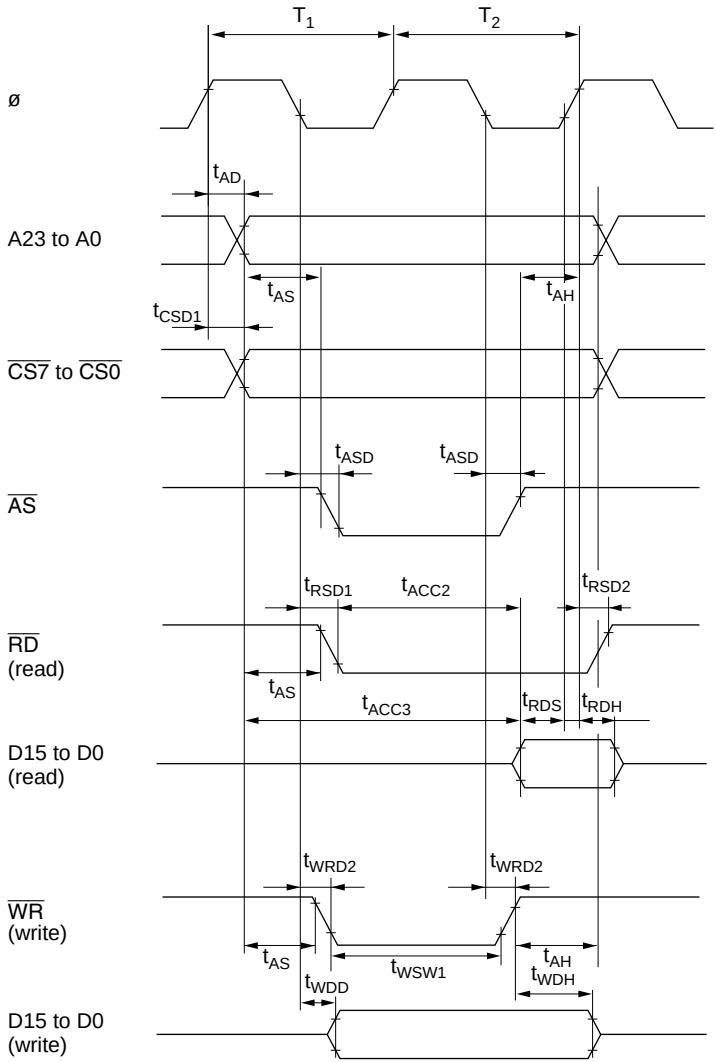


Figure 25-6 Basic Bus Timing (Two-State Access)

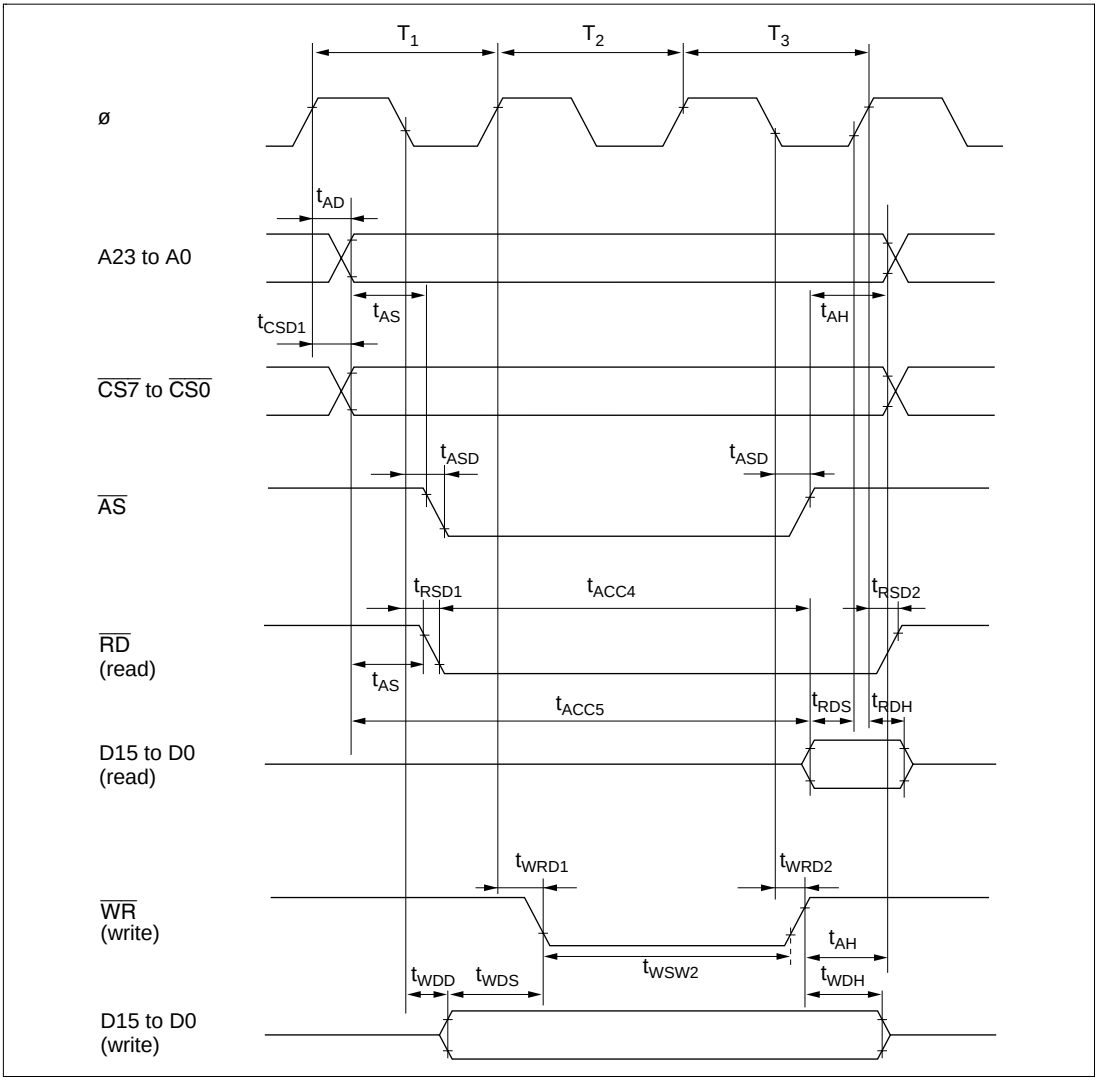


Figure 25-7 Basic Bus Timing (Three-State Access)

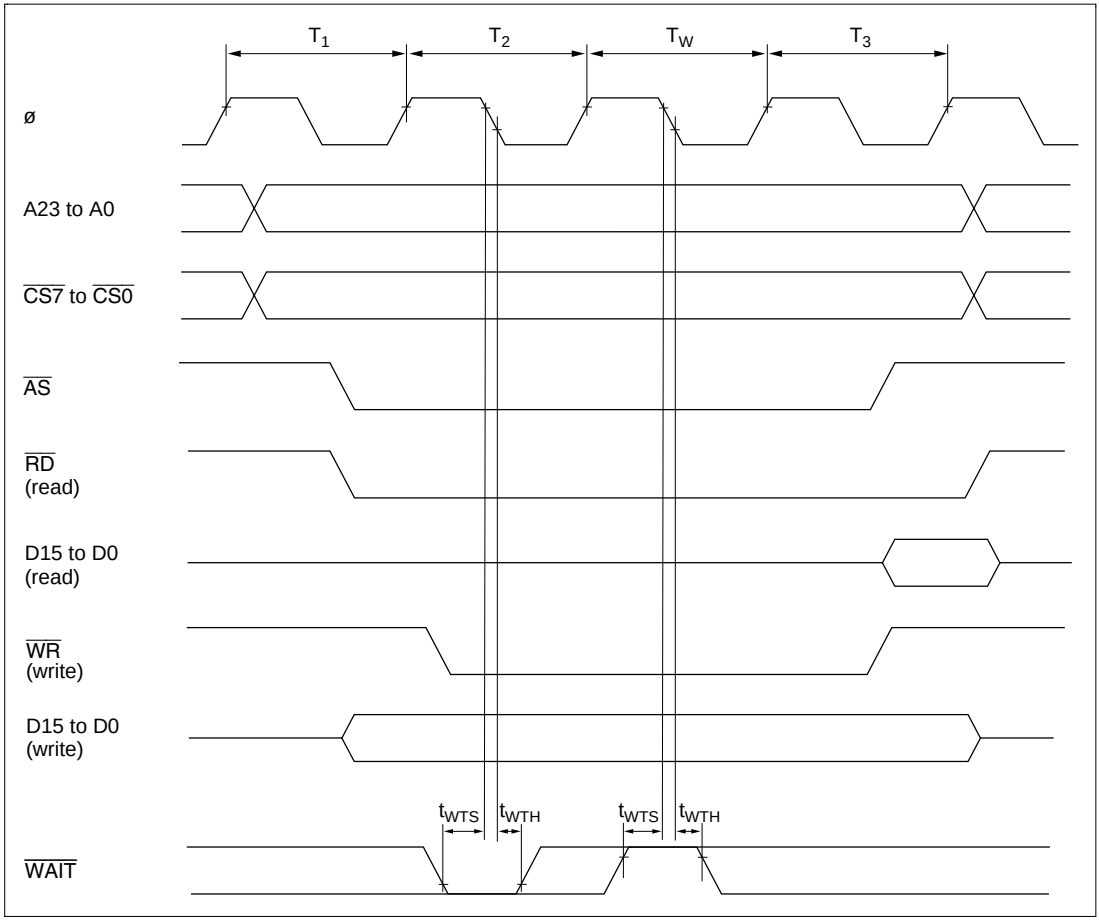


Figure 25-8 Basic Bus Timing (Three-State Access with One Wait State)

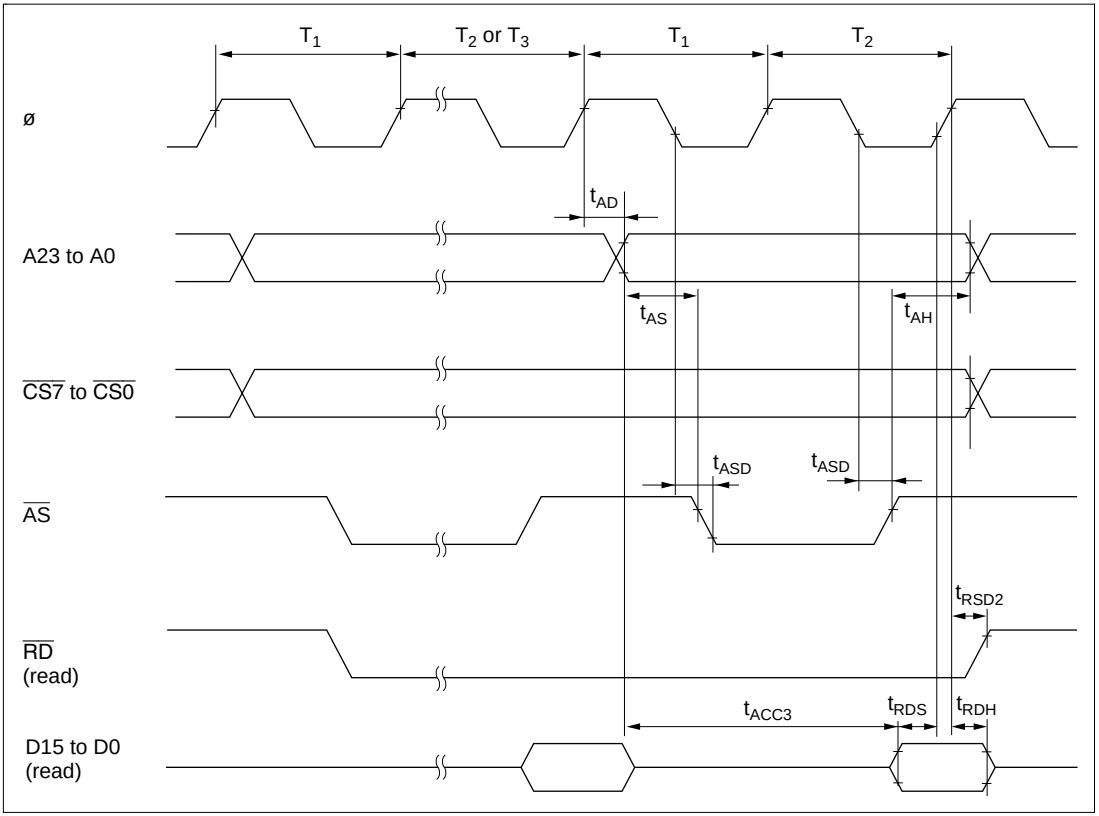


Figure 25-9 Burst ROM Access Timing (Two-State Access)

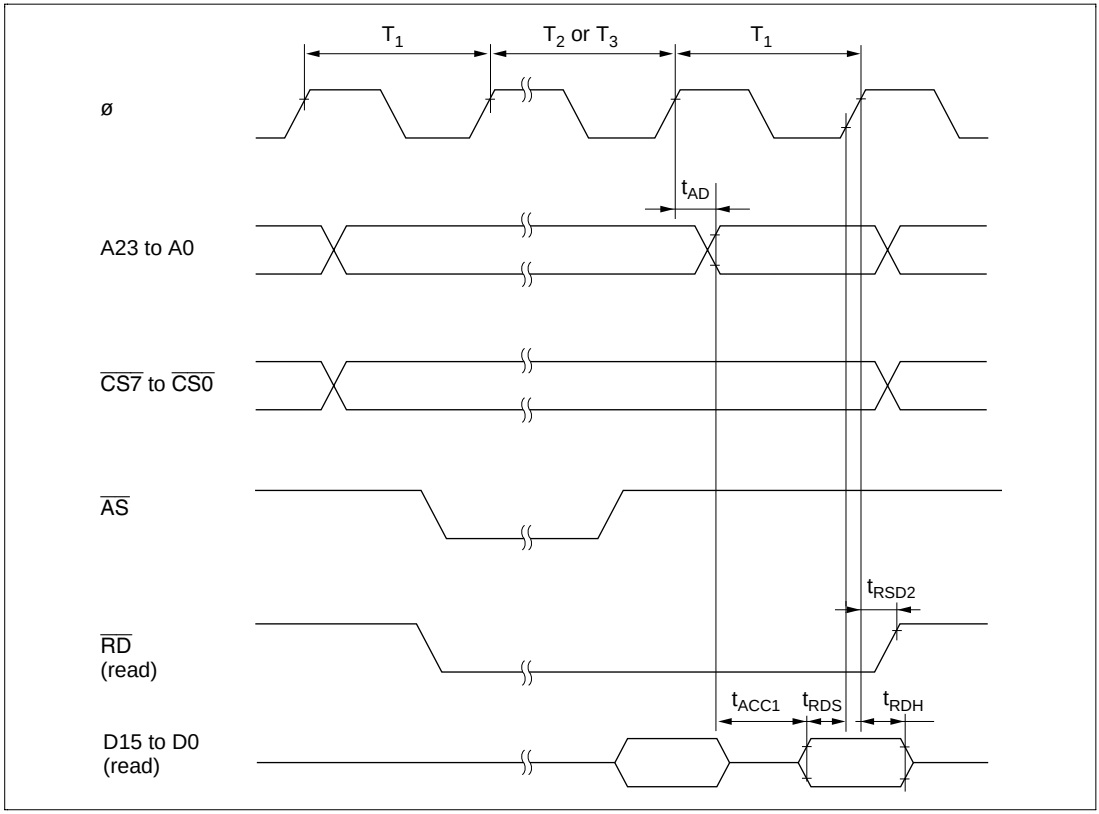


Figure 25-10 Burst ROM Access Timing (One-State Access)

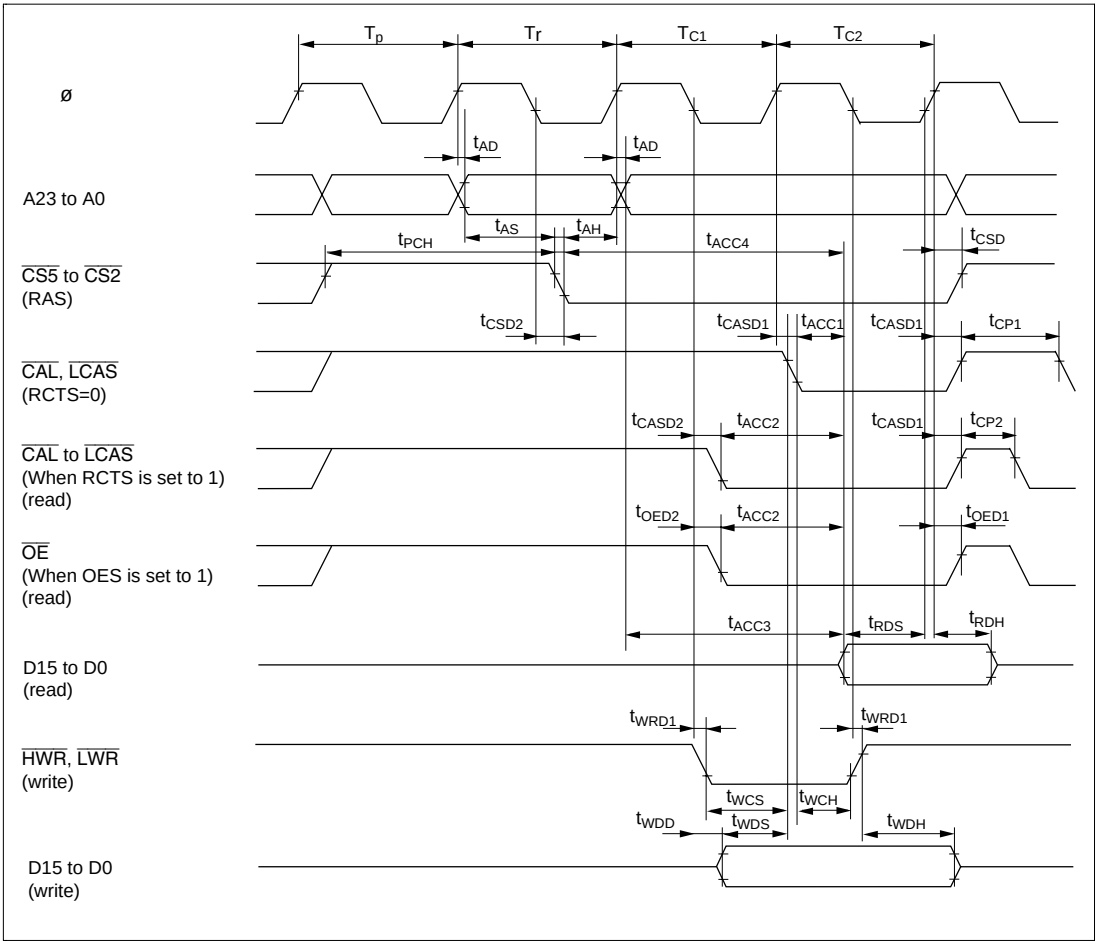


Figure 25-11 DRAM Access Timing

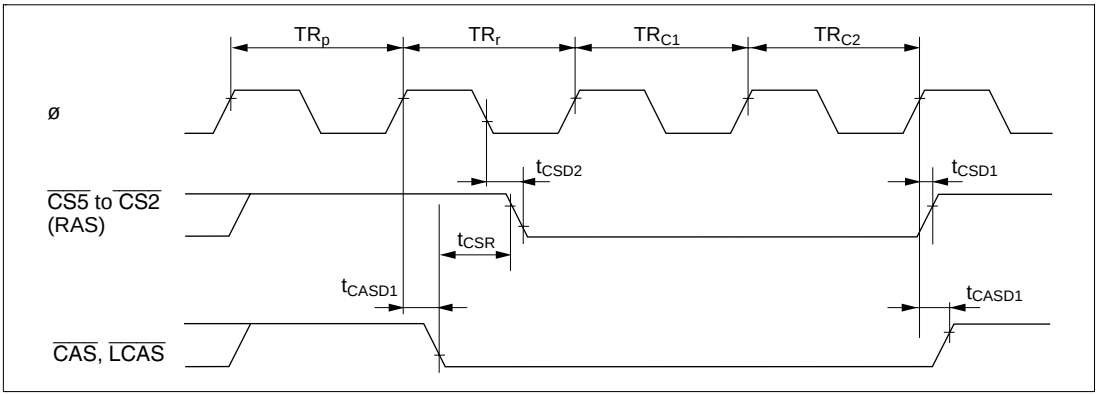


Figure 25-12 DRAM CBR Refresh Timing

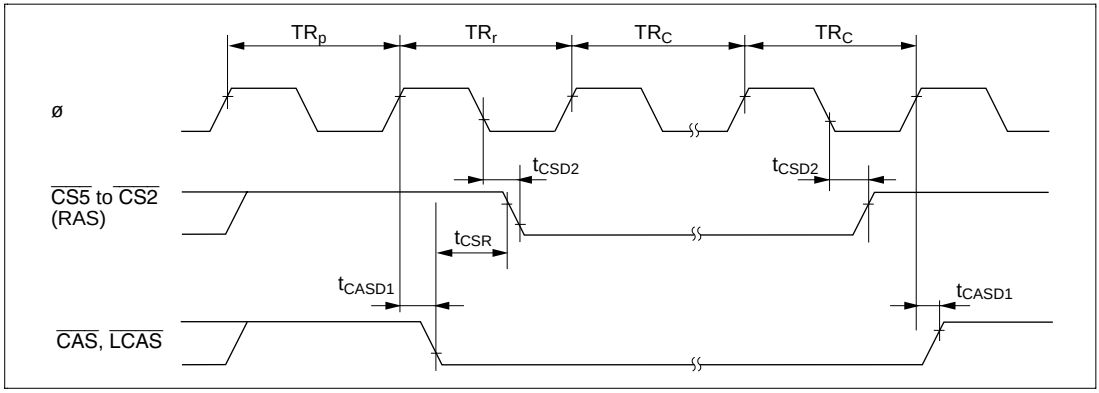


Figure 25-13 DRAM Self-Refresh Timing

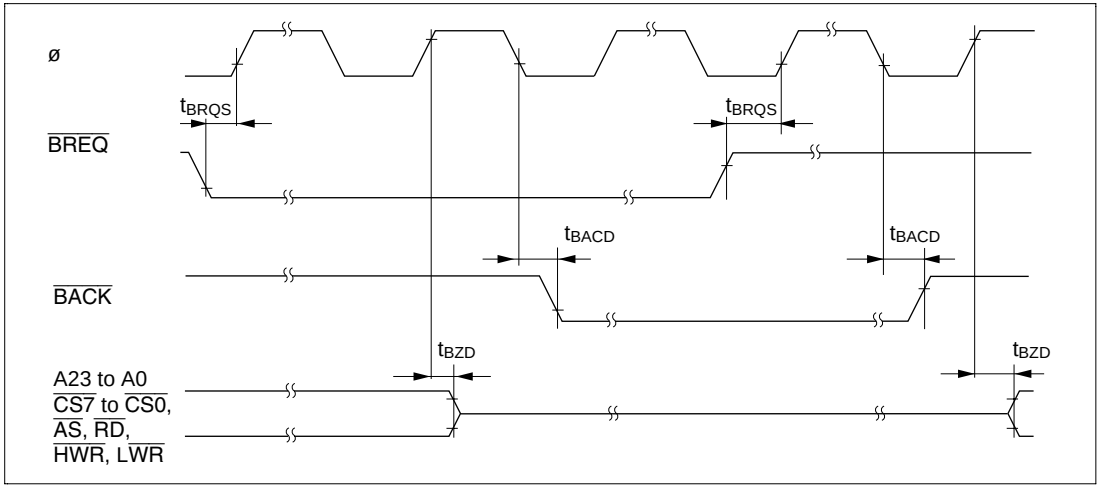


Figure 25-14 External Bus Release Timing

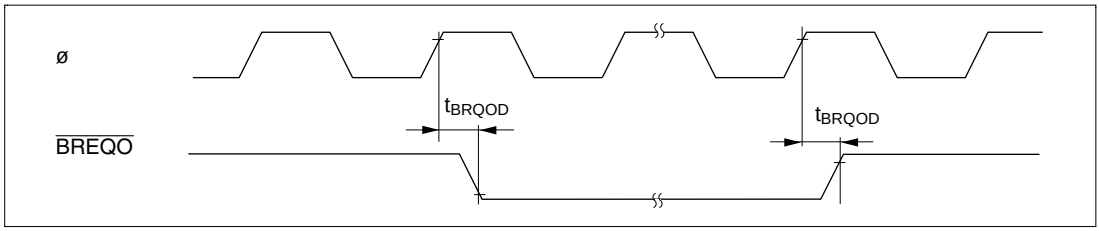


Figure 25-15 External Bus Request Output Timing

25.3.4 DMAC Timing

Table 25-8 shows the DMAC timing.

Table 25-8 DMAC Timing

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	40	—	25	—	ns	Figure 25-19
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—	10	—		
$\overline{\text{TEND}}$ delay time	t_{TED}	—	30	—	20		Figure 25-18
$\overline{\text{DACK}}$ delay time1	t_{DACD1}	—	30	—	18	ns	Figure 25-16
$\overline{\text{DACK}}$ delay time2	t_{DACD2}	—	30	—	18		Figure 25-17

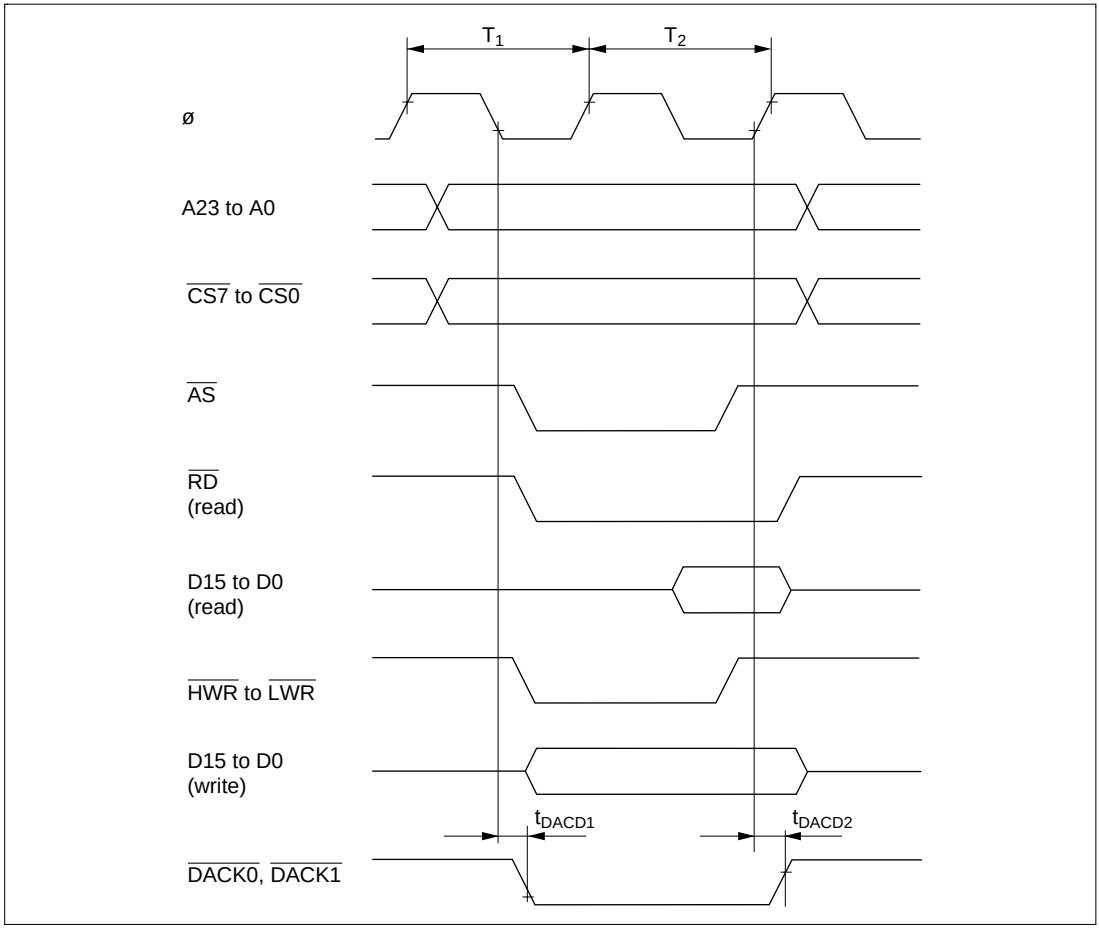


Figure 25-16 DMAC Single Address Transfer Timing / Two-State Access

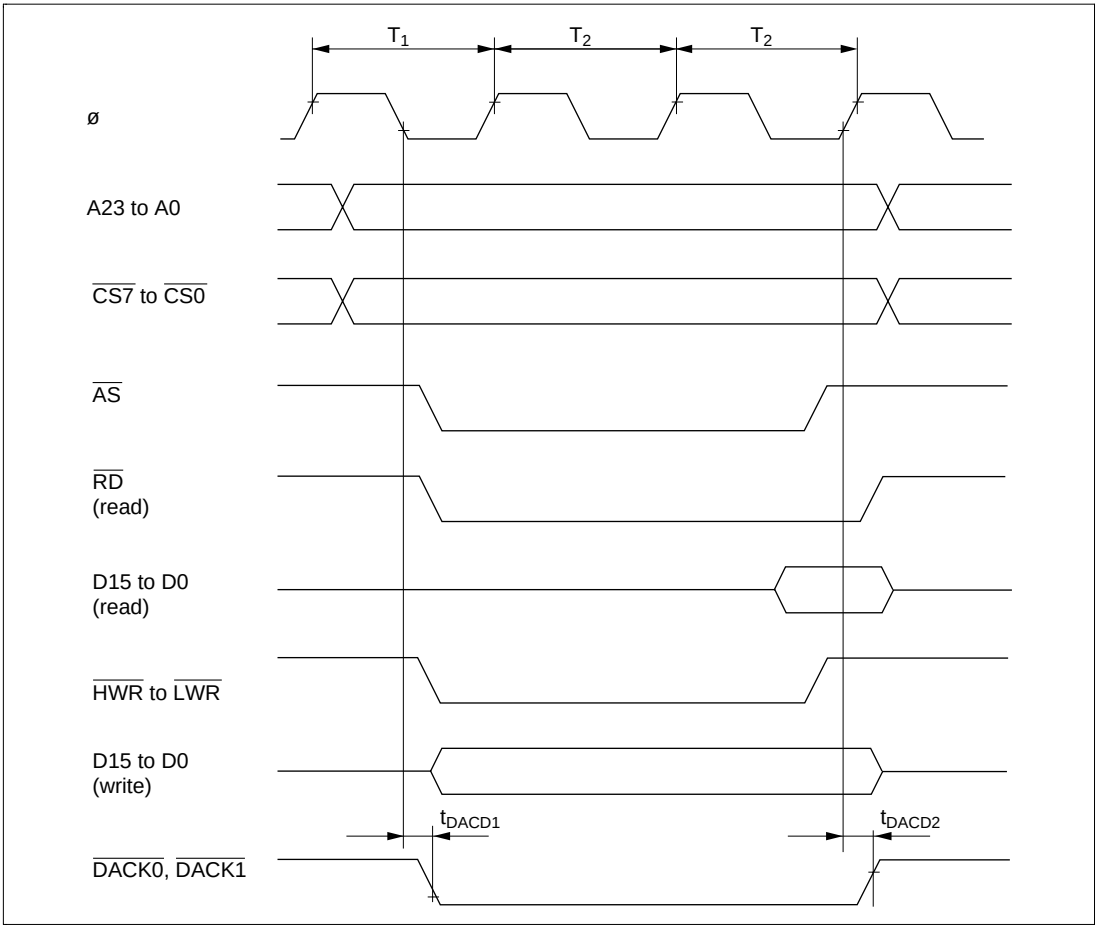


Figure 25-17 DMAC Single Address Transfer Timing / Three-State Access

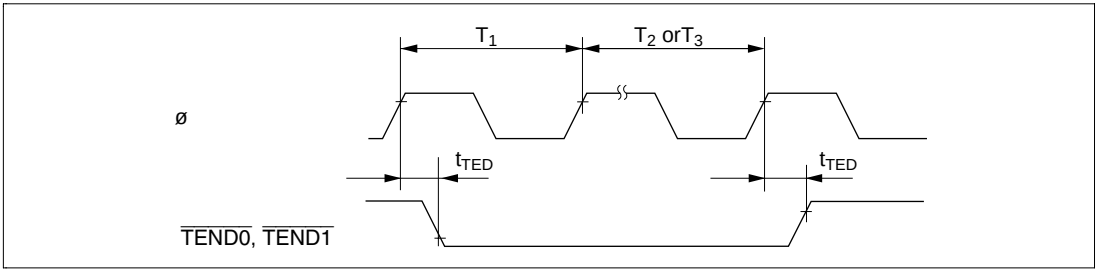


Figure 25-18 DMAC TEND Output Timing

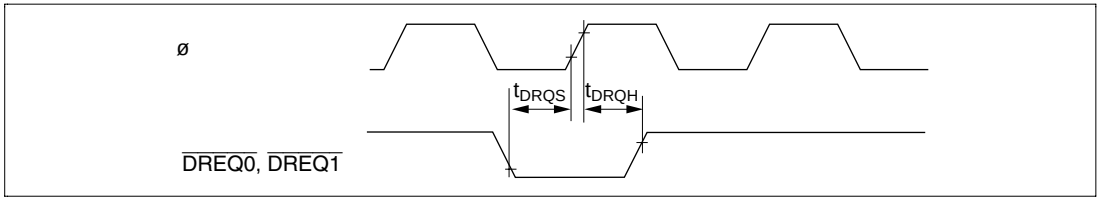


Figure 25-19 DMAC DREQ Input Timing

25.3.5 Timing of On-Chip Supporting Modules

Table 25-9 lists the timing of on-chip supporting modules.

Table 25-9 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz*, 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 32.768 kHz*, 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition A		Condition B			
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O port	Output data delay time		t_{PWD}	—	60	—	40	ns	Figure 25-20
	Input data setup time		t_{PRS}	40	—	25	—		
	Input data hold time		t_{PRH}	40	—	25	—		
PPG	Pulse output delay time		t_{POD}	—	60	—	40	ns	Figure 25-21
TPU	Timer output delay time		t_{TOCD}	—	60	—	40	ns	Figure 25-22
	Timer input setup time		t_{TICS}	40	—	25	—		
	Timer clock input setup time		t_{TCKS}	40	—	25	—	ns	Figure 25-23
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		

Item			Symbol	Condition A Min	Condition A Max	Condition B Min	Condition B Max	Unit	Test Conditions
TMR	Timer output delay time		t_{TMOD}	—	60	—	40	ns	Figure 25-24
	Timer reset input setup time		t_{TMRS}	40	—	25	—	ns	Figure 25-26
	Timer clock input setup time		t_{TMCS}	40	—	25	—	ns	Figure 25-25
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—	2.5	—		
WDT0	Overflow output delay time		t_{WOVD}	—	60	—	40	ns	Figure 25-27
WDT1	Buzz output delay time		t_{BUZD}	—	60	—	40	ns	Figure 25-28
PWM	Pulse output delay time		t_{PWOD}	—	60	—	40	ns	Figure 25-29
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	4	—	t_{cyc}	Figure 25-30
		Synchronous		6	—	6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5		
	Transmit data delay time		t_{TXD}	—	60	—	40	ns	Figure 25-31
	Receive data setup time (synchronous)		t_{RXS}	60	—	40	—		
	Receive data hold time (synchronous)		t_{RXH}	60	—	40	—		
A/D converter	Trigger input setup time		t_{TRGS}	60	—	40	—	ns	Figure 25-32

Note: * Only available I/O port, TMR, WDT0, and WDT1.

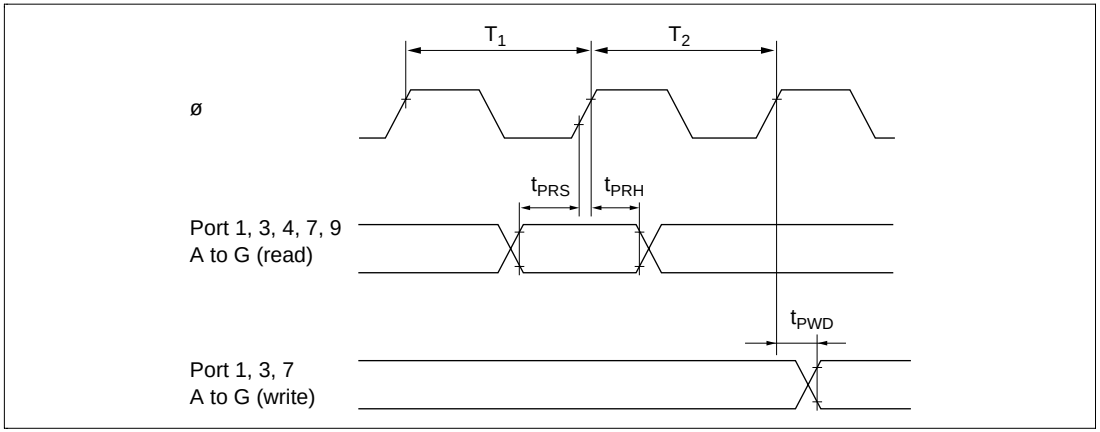


Figure 25-20 I/O Port Input/Output Timing

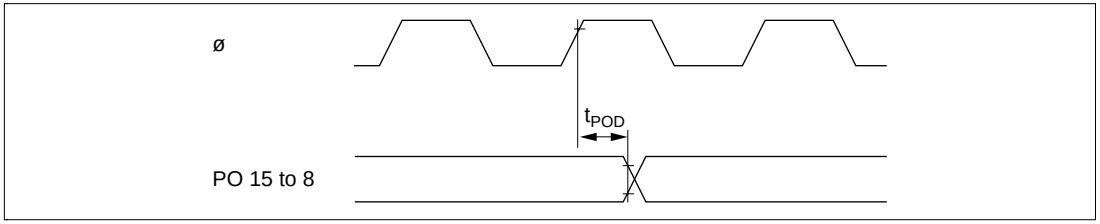


Figure 25-21 PPG Output Timing

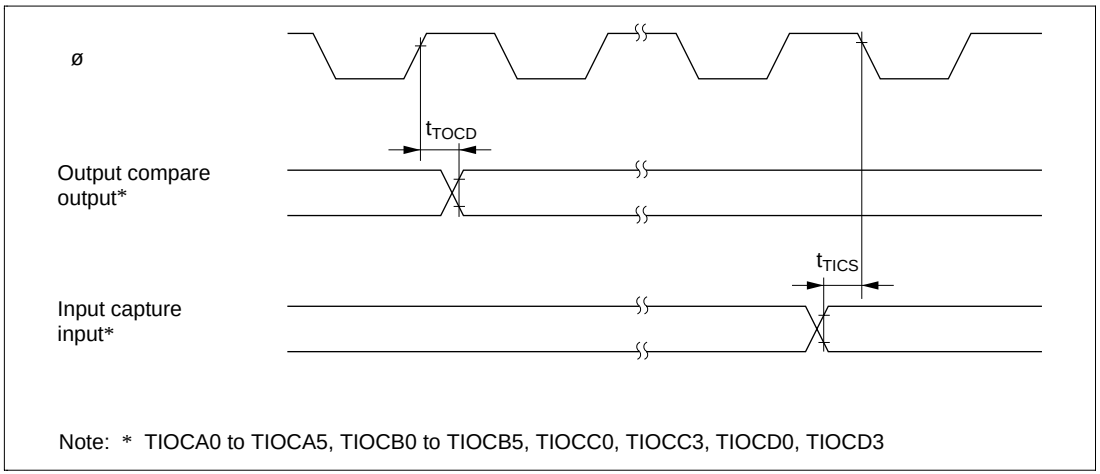


Figure 25-22 TPU Input/Output Timing

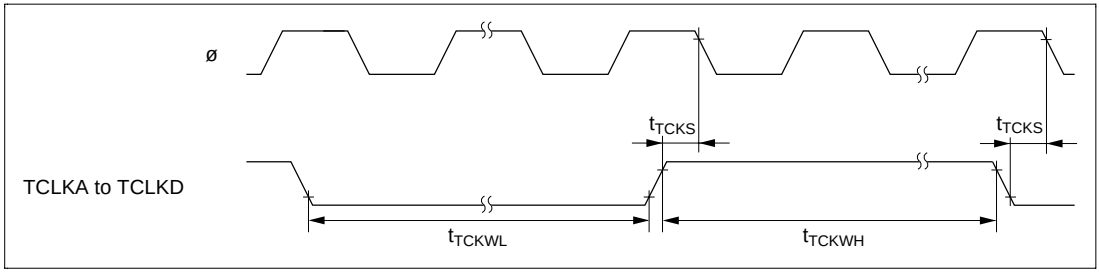


Figure 25-23 TPU Clock Input Timing

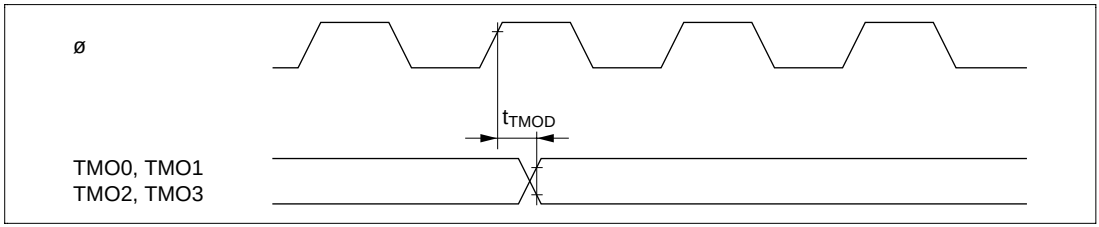


Figure 25-24 8-bit Timer Output Timing

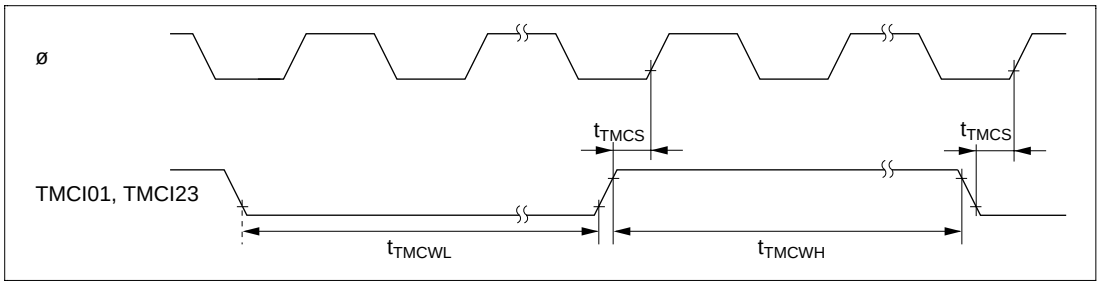


Figure 25-25 8-bit Timer Clock Input Timing

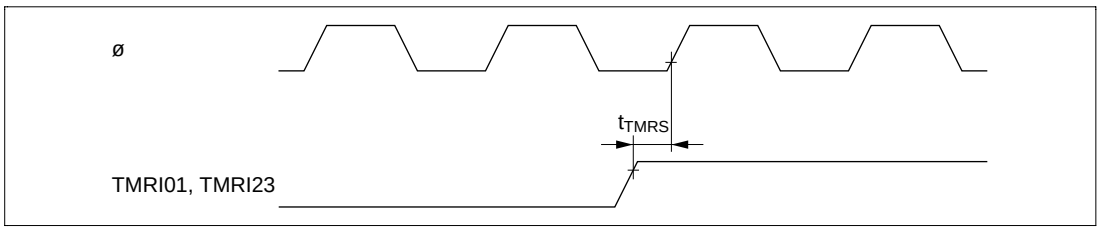


Figure 25-26 8-bit Timer Reset Input Timing

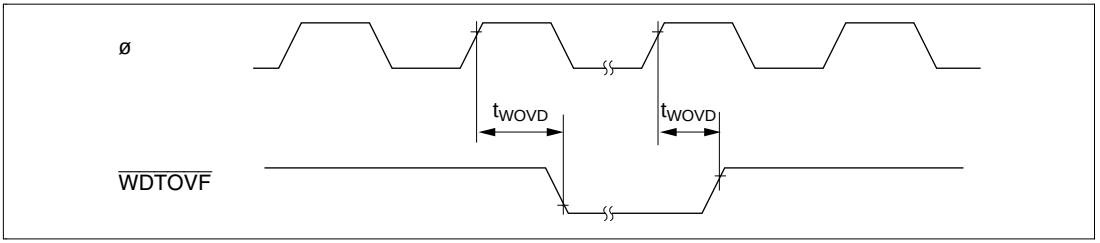


Figure 25-27 WDT0 Output Timing

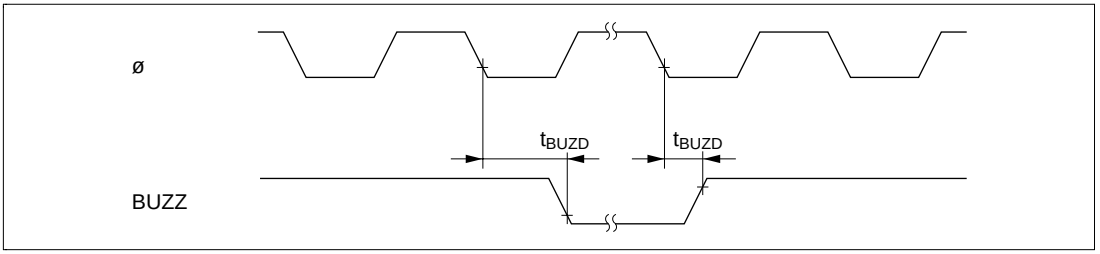


Figure 25-28 WDT1 Output Timing

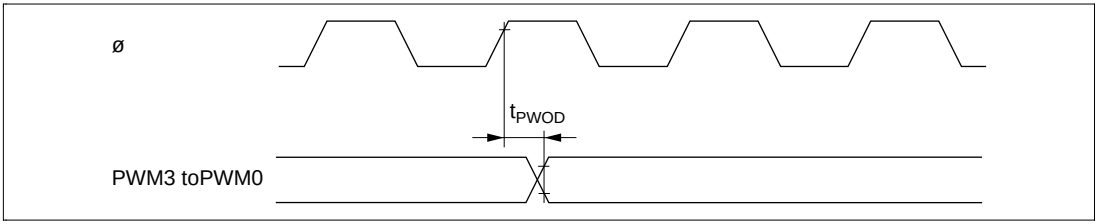


Figure 25-29 PWM Output Timing

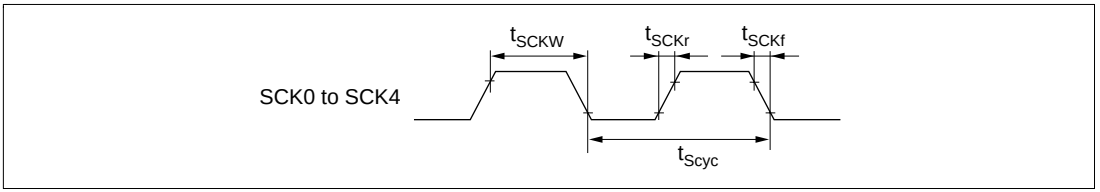


Figure 25-30 SCK Clock Input Timing

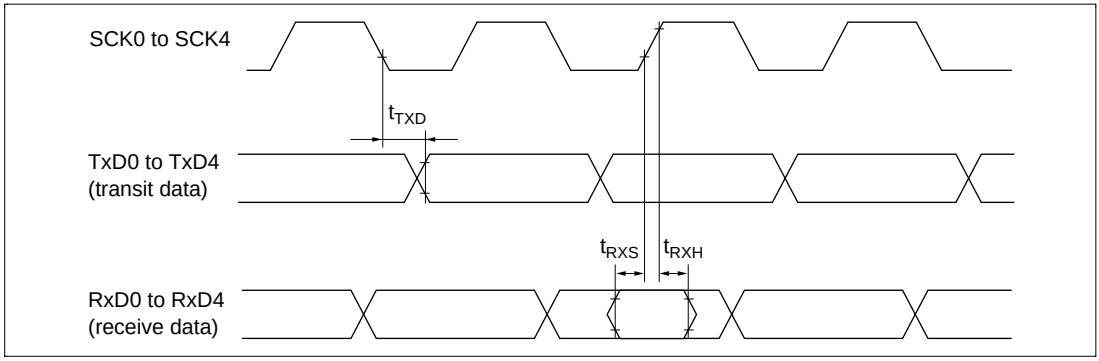


Figure 25-31 SCI Input/Output Timing (Clock Synchronous Mode)

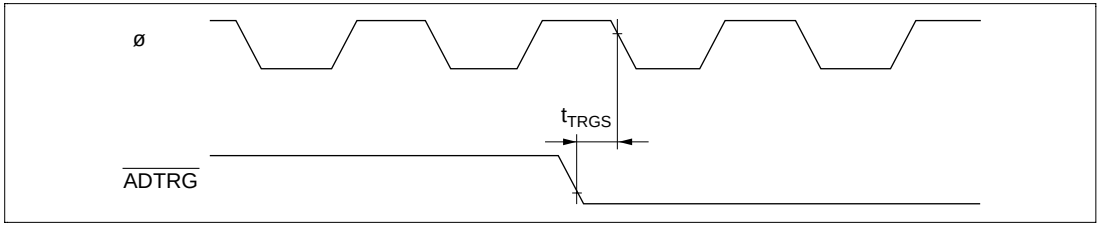


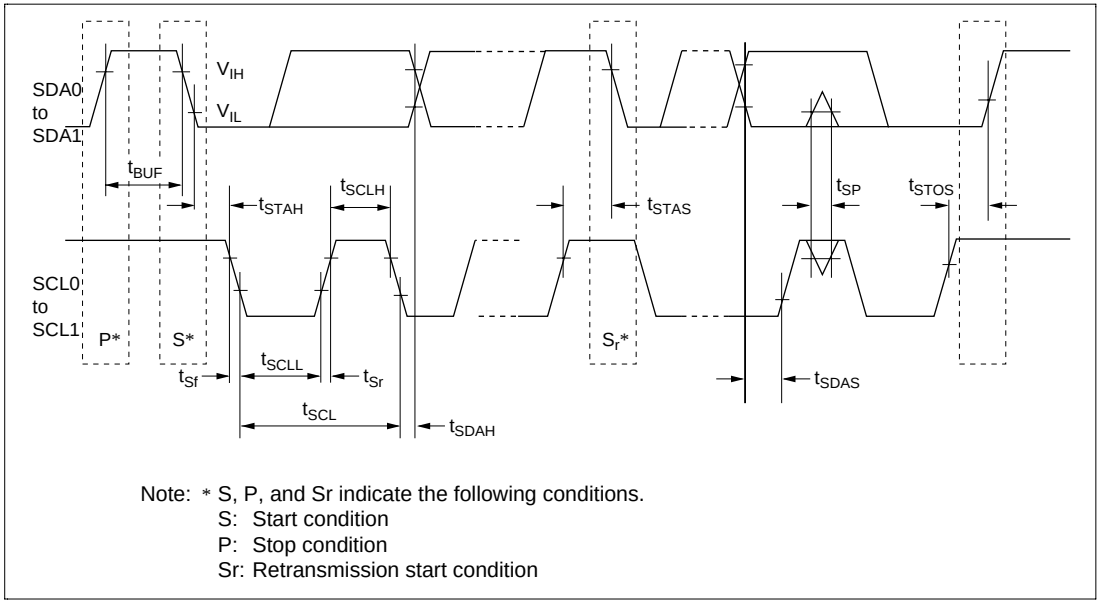
Figure 25-32 A/D Converter External Trigger Input Timing

Table 25-10 I^2C Bus Timing

Conditions: V_{CC} = PLV_{CC} = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, V_{SS} = 0 V, ø = 5 MHz to maximum operating frequency, T_a = –20°C to +75°C

		Ratings				
Item	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Notes**
SCL input cycle time	t_{SCL}	$12t_{cyc}$	—	—	ns	Figure 25-33
SCL input high pulse width	t_{SCLH}	$3t_{cyc}$	—	—	ns	
SCL input low pulse width	t_{SCLL}	$5t_{cyc}$	—	—	ns	
SCL, SDA input rise time	t_{Sr}	—	—	$7.5t_{cyc}$*	ns	
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	$1t_{cyc}$	ns	
SDA input bus free time	t_{BUF}	$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}	$3t_{cyc}$	—	—	ns	
Data input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns	
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: * $17.5t_{cyc}$ can be set according to the clock selected for use by the I^2C module. For details, see section 18.4, Usage Notes.



Note: * S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 25-33 I²C Bus Inteface Input/Output Timing (Option)

25.4 A/D Conversion Characteristics

Table 25-11 lists the A/D conversion characteristics.

Table 25-11 A/D Conversion Characteristics

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	bits
Conversion time	16	—	—	10	—	—	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.5	—	—	±3.5	LSB
Offset error	—	—	±7.5	—	—	±3.5	LSB
Full-scale error	—	—	±7.0	—	—	±3.5	LSB
Quantization	—	±0.5	—	—	±0.5	—	LSB
Absolute accuracy	—	—	±8.0	—	—	±4.0	LSB

25.5 D/A Conversion Characteristics

Table 25-12 shows the D/A conversion characteristics.

Table 25-12 D/A Conversion Characteristics

Condition A: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.3 V to 5.5 V, V_{ref} = 3.3 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ø = 2 to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	8	8	8	bits	
Conversion time	—	—	10	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	± 2.0	± 3.0	—	± 1.5	± 2.0	LSB	2-MΩ resistive load
	—	—	± 2.0	—	—	± 1.5	LSB	4-MΩ resistive load

25.6 Flash Memory Characteristics

Table 25-14 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Programming time*1, *2, *4		t_P	—	10	200	ms/128 bytes
Erase time*1, *3, *5		t_E	—	100	1000	ms/block
Number of rewrites		N_{WEC}	—	50	100	Times
Programming	Wait time after SWE1 bit setting*1	x0	1	—	—	µs
	Wait time after PSU1 bit setting*1	y	50	—	—	µs
	Wait time after P1 bit setting*1, *4	z0	—	—	30	µs
		z1	—	—	10	µs
		z2	—	—	200	µs
	Wait time after P1 bit clearing*1	α	5	—	—	µs
	Wait time after PSU1 bit clearing*1	β	5	—	—	µs
	Wait time after PV1 bit setting*1	γ	4	—	—	µs
	Wait time after H'FF dummy write*1	ε	2	—	—	µs
	Wait time after PV1 bit clearing*1	η	2	—	—	µs
	Maximum number of writes*1, *4	N1	—	—	6	Times
		N2	—	—	994	Times
Common	Wait time after SWE1 bit clearing*1	x1	100	—	—	µs
Erasing	Wait time after SWE1 bit setting*1	x	1	—	—	µs
	Wait time after ESU1 bit setting*1	y	100	—	—	µs
	Wait time after E1 bit setting*1, *5	z	—	—	10	ms
	Wait time after E1 bit clearing*1	α	10	—	—	µs
	Wait time after ESU1 bit clearing*1	β	10	—	—	µs
	Wait time after EV1 bit setting*1	γ	6	—	—	µs
	Wait time after H'FF dummy write*1	ε	2	—	—	µs
	Wait time after EV1 bit clearing*1	η	4	—	—	µs
	Maximum number of erases*1, *5	N	—	—	100	Times

Notes: *1 Follow the program/erase algorithms when making the time settings.

*2 Programming time per 128 bytes. (Indicates the total time during which the P1 bit is set in flash memory control register 1 (FLMCR1). Does not include the program-verify time.)

*3 Time to erase one block. (Indicates the time during which the E1 bit is set in FLMCR1. Does not include the erase-verify time.)

*4 Maximum programming time

(t_P(max) = Wait time after P1 bit setting (z) × maximum number of writes (N))

(z0 + z1) × 6 + z2 × 994

*5 Maximum erase time

(t_E(max) = Wait time after E1 bit setting (z) × maximum number of erases (N))

25.7 Usage Note

Although both the F-ZTAT and mask ROM versions fully meet the electrical specifications listed in this manual, due to differences in the fabrication process, the on-chip ROM, and the layout patterns, there will be differences in the actual values of the electrical characteristics, the operating margins, the noise margins, and other aspects.

Therefore, if a system is evaluated using the F-ZTAT version, a similar evaluation should also be performed using the mask ROM version.

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
¬	Logical NOT (logical complement)
( ) < >	Contents of operand
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Condition Code Notation

Symbol	
↕	Changes according to the result of instruction
*	Undetermined (no guaranteed value)
0	Always cleared to 0
1	Always set to 1
—	Not affected by execution of the instruction

Table A-1 Instruction Set

(1) Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
MOV	MOV.W @(d:16,ERs),Rd	W				4						@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3
	MOV.W @(d:32,ERs),Rd	W				8						@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5
	MOV.W @ERs+,Rd	W					2					@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3
	MOV.W @aa:16,Rd	W						4				@aa:16→Rd16	—	—	↕	↕	0	—	3
	MOV.W @aa:32,Rd	W						6				@aa:32→Rd16	—	—	↕	↕	0	—	4
	MOV.W Rs,@ERd	W			2							Rs16→@ERd	—	—	↕	↕	0	—	2
	MOV.W Rs,@(d:16,ERd)	W				4						Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.W Rs,@(d:32,ERd)	W				8						Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.W Rs,@-ERd	W					2					ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:16	W						4				Rs16→@aa:16	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:32	W						6				Rs16→@aa:32	—	—	↕	↕	0	—	4
	MOV.L #xx:32,ERd	L	6									#xx:32→ERd32	—	—	↕	↕	0	—	3
	MOV.L ERs,ERd	L		2								ERs32→ERd32	—	—	↕	↕	0	—	1
	MOV.L @ERs,ERd	L			4							@ERs→ERd32	—	—	↕	↕	0	—	4
	MOV.L @(d:16,ERs),ERd	L				6						@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5
	MOV.L @(d:32,ERs),ERd	L				10						@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7
	MOV.L @ERs+,ERd	L					4					@ERs→ERd32,ERs32+4→@ERs32	—	—	↕	↕	0	—	5
	MOV.L @aa:16,ERd	L						6				@aa:16→ERd32	—	—	↕	↕	0	—	5
	MOV.L @aa:32,ERd	L						8				@aa:32→ERd32	—	—	↕	↕	0	—	6

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
MOV	MOV.L ERs,@ERd	L			4							ERs32→@ERd	—	—	↕	↕	0	—	4
	MOV.L ERs,@(d:16,ERd)	L				6						ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5
	MOV.L ERs,@(d:32,ERd)	L				10						ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7
	MOV.L ERs,@-ERd	L					4					ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:16	L						6				ERs32→@aa:16	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:32	L						8				ERs32→@aa:32	—	—	↕	↕	0	—	6
POP	POP.W Rn	W									2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3
	POP.L ERn	L									4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5
PUSH	PUSH.W Rn	W									2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3
	PUSH.L ERn	L									4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5
LDM	LDM @SP+,(ERm-ERn)	L									4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—	7/9/11 [1]
STM	STM (ERm-ERn),@-SP	L									4	(SP-4→SP,ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—	7/9/11 [1]
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the H8S/2633 Series																	[2]
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the H8S/2633 Series																	[2]

(2) Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1
DAA	DAA Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	↕	1
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SUB	SUB.W Rs,Rd	W		2								Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—	—	—	—	1
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—	—	—	—	1
	SUBS #4,ERd	L		2								ERd32-4→ERd32	—	—	—	—	—	—	1
DEC	DEC.B Rd	B		2								Rd8-1→Rd8	—	—	↕	↕	↕	—	1
	DEC.W #1,Rd	W		2								Rd16-1→Rd16	—	—	↕	↕	↕	—	1
	DEC.W #2,Rd	W		2								Rd16-2→Rd16	—	—	↕	↕	↕	—	1
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕	↕	↕	—	1
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕	↕	↕	—	1
DAS	DAS Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	—	1
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	3
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	4
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	4
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	5

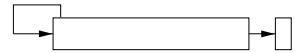
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20
DIVXS	DIVXS.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1
EXTU	EXTU.W Rd	W		2								0→(<bit 15 to 8> of Rd16)	—	—	0	↕	0	—	1
	EXTU.L ERd	L		2								0→(<bit 31 to 16> of ERd32)	—	—	0	↕	0	—	1

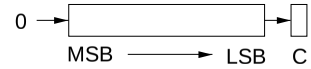
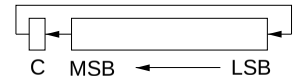
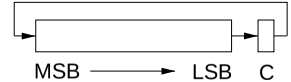
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—	↕	↕	0	—	1
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—	↕	↕	0	—	1
TAS*3	TAS @ERd	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—	↕	↕	0	—	4
MAC	MAC @ERn+, @ERm+	—					4					@ERnx@ERm+MAC→MAC (signal multiplication) @ERn+2→ERn, ERm+2→ERm	—	—	— [11]	— [11]	— [11]	—	4
CLRMAC	CLRMAC	—									2	0→MACH, MACL	—	—	—	—	—	—	2 [12]
LDMAC	LDMAC ERs,MACH	L		2								ERs→MACH	—	—	—	—	—	—	2 [12]
	LDMAC ERs,MACL	L		2								ERs→MACL	—	—	—	—	—	—	2 [12]
STMAC	STMAC MACH,ERd	L		2								MACH→ERd	—	—	↕	↕	↕	—	1 [12]
	STMAC MACL,ERd	L		2								MACL→ERd	—	—	↕	↕	↕	—	1 [12]

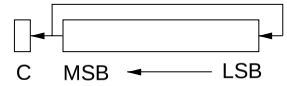
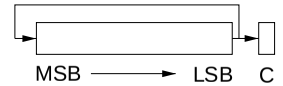
(3) Logical Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0	—	1
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0	—	1
	NOT.L ERd	L		2								¬ ERd32→ERd32	—	—	↕	↕	0	—	1

(4) Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	1
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SHLR	SHLR.B Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.B #2,Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.W Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.W #2,Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.L ERd	L		2									—	—	0	↕	0	↕	1
	SHLR.L #2,ERd	L		2									—	—	0	↕	0	↕	1
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

(5) Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BCLR	BCLR Rn,@aa:32	B						8				(Rn8 of @aa:32)←0	—	—	—	—	—	—	6
BNOT	BNOT #xx:3,Rd	B		2								(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—	—	—	—	1
	BNOT #xx:3,@ERd	B			4							(#xx:3 of @ERd)← [¬ (#xx:3 of @ERd)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)← [¬ (#xx:3 of @aa:8)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)← [¬ (#xx:3 of @aa:16)]	—	—	—	—	—	—	5
	BNOT #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)← [¬ (#xx:3 of @aa:32)]	—	—	—	—	—	—	6
	BNOT Rn,Rd	B		2								(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—	—	—	—	1
	BNOT Rn,@ERd	B			4							(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:8	B						4				(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:16	B						6				(Rn8 of @aa:16)← [¬ (Rn8 of @aa:16)]	—	—	—	—	—	—	5
	BNOT Rn,@aa:32	B						8				(Rn8 of @aa:32)← [¬ (Rn8 of @aa:32)]	—	—	—	—	—	—	6
BTST	BTST #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BTST	BTST #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5
	BTST Rn,Rd	B		2								¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST Rn,@ERd	B			4							¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:8	B						4				¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:16	B						6				¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4
	BTST Rn,@aa:32	B						8				¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5
BLD	BLD #xx:3,Rd	B		2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BLD #xx:3,@ERd	B			4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BLD #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BILD	BILD #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BILD #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BILD #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BST	BST #xx:3,Rd	B		2								C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BST #xx:3,@ERd	B			4							C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BST #xx:3,@aa:8	B						4				C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BST	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BIST	BIST #xx:3,Rd	B		2								¬ C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BIST #xx:3,@ERd	B			4							¬ C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:8	B						4				¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:16	B						6				¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BIST #xx:3,@aa:32	B						8				¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BAND	BAND #xx:3,Rd	B		2								C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BAND #xx:3,@ERd	B			4							C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:8	B						4				C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:16	B						6				C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BAND #xx:3,@aa:32	B						8				C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIAND	BIAND #xx:3,Rd	B		2								C∧[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIAND #xx:3,@ERd	B			4							C∧[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:8	B						4				C∧[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:16	B						6				C∧[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIAND #xx:3,@aa:32	B						8				C∧[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BOR	BOR #xx:3,Rd	B		2								C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BOR #xx:3,@ERd	B			4							C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BOR	BOR #xx:3,@aa:8	B						4				C∨(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BOR #xx:3,@aa:16	B						6				C∨(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BOR #xx:3,@aa:32	B						8				C∨(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIOR	BIOR #xx:3,Rd	B		2								C∨[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIOR #xx:3,@ERd	B			4							C∨[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:8	B						4				C∨[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:16	B						6				C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIOR #xx:3,@aa:32	B						8				C∨[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BXOR	BXOR #xx:3,Rd	B		2								C⊕(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BXOR #xx:3,@ERd	B			4							C⊕(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:8	B						4				C⊕(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:16	B						6				C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BXOR #xx:3,@aa:32	B						8				C⊕(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIXOR	BIXOR #xx:3,Rd	B		2								C⊕[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIXOR #xx:3,@ERd	B			4							C⊕[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:8	B						4				C⊕[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:16	B						6				C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIXOR #xx:3,@aa:32	B						8				C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5

(6) Branch Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Branching Condition	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z	V	C	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then	Always	—	—	—	—	—	—	2
	BRA d:16(BT d:16)	—							4			PC←PC+d		—	—	—	—	—	—	3
	BRN d:8(BF d:8)	—							2			else next;	Never	—	—	—	—	—	—	2
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—	3
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—	2
	BHI d:16	—							4					—	—	—	—	—	—	3
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—	2
	BLS d:16	—							4					—	—	—	—	—	—	3
	BCC d:B(BHS d:8)	—							2				C=0	—	—	—	—	—	—	2
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—	3
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—	2
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—	3
	BNE d:8	—							2				Z=0	—	—	—	—	—	—	2
	BNE d:16	—							4					—	—	—	—	—	—	3
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—	2
	BEQ d:16	—							4					—	—	—	—	—	—	3
	BVC d:8	—							2				V=0	—	—	—	—	—	—	2
	BVC d:16	—							4					—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Branching Condition	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
Bcc	BVS d:8	—							2				V=1	—	—	—	—	—	—	2
	BVS d:16	—							4					—	—	—	—	—	—	3
	BPL d:8	—							2				N=0	—	—	—	—	—	—	2
	BPL d:16	—							4					—	—	—	—	—	—	3
	BMI d:8	—							2				N=1	—	—	—	—	—	—	2
	BMI d:16	—							4					—	—	—	—	—	—	3
	BGE d:8	—							2				N⊕V=0	—	—	—	—	—	—	2
	BGE d:16	—							4					—	—	—	—	—	—	3
	BLT d:8	—							2				N⊕V=1	—	—	—	—	—	—	2
	BLT d:16	—							4					—	—	—	—	—	—	3
	BGT d:8	—							2				Z∨(N⊕V)=0	—	—	—	—	—	—	2
	BGT d:16	—							4					—	—	—	—	—	—	3
	BLE d:8	—							2				Z∨(N⊕V)=1	—	—	—	—	—	—	2
	BLE d:16	—							4					—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
JMP	JMP @ERn	—			2							PC←ERn	—	—	—	—	—	—	2
	JMP @aa:24	—						4				PC←aa:24	—	—	—	—	—	—	3
	JMP @@aa:8	—								2		PC←@aa:8	—	—	—	—	—	—	5
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—	—	—	—	4
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16	—	—	—	—	—	—	5
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn	—	—	—	—	—	—	4
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24	—	—	—	—	—	—	5
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8	—	—	—	—	—	—	6
RTS	RTS	—									2	PC←@SP+	—	—	—	—	—	—	5

(7) System Control Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
STC	STC CCR,Rd	B		2								CCR→Rd8	—	—	—	—	—	—	1
	STC EXR,Rd	B		2								EXR→Rd8	—	—	—	—	—	—	1
	STC CCR,@ERd	W			4							CCR→@ERd	—	—	—	—	—	—	3
	STC EXR,@ERd	W			4							EXR→@ERd	—	—	—	—	—	—	3
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—	—	—	—	—	4
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—	—	—	—	—	4
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—	—	—	—	—	5
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—	—	—	—	—	5
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—	—	—	—	—	2
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—	—	—	—	—	2
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—	—	—	—	—	2
NOP	NOP	—									2	PC←PC+2	—	—	—	—	—	—	1

(8) Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n *2
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n *2

Notes: *1 The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.
*2 n is the initial value of R4L or R4.
*3 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.
[1] Seven states for saving or restoring two registers, nine states for three registers, or eleven states for four registers.
[2] Cannot be used in the H8S/2633 Series.
[3] Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
[4] Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
[5] Retains its previous value when the result is zero; otherwise cleared to 0.
[6] Set to 1 when the divisor is negative; otherwise cleared to 0.
[7] Set to 1 when the divisor is zero; otherwise cleared to 0.
[8] Set to 1 when the quotient is negative; otherwise cleared to 0.
[9] One additional state is required for execution when EXR is valid.

HITACHI

A.2 Instruction Codes

Table A-2 shows the instruction codes.

Table A-2 Instruction Codes

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM								
	ADD.B Rs,Rd	B	0 8	rs rd								
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM							
	ADD.W Rs,Rd	W	0 9	rs rd								
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM							
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd								
ADDS	ADDS #1,ERd	L	0 B	0 0 erd								
	ADDS #2,ERd	L	0 B	8 0 erd								
	ADDS #4,ERd	L	0 B	9 0 erd								
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM								
	ADDX Rs,Rd	B	0 E	rs rd								
AND	AND.B #xx:8,Rd	B	E rd	IMM								
	AND.B Rs,Rd	B	1 6	rs rd								
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM							
	AND.W Rs,Rd	W	6 6	rs rd								
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM							
	AND.L ERs,ERd	L	0 1	F 0	6 6	0 ers 0 erd						
ANDC	ANDC #xx:8,CCR	B	0 6	IMM								
	ANDC #xx:8,EXR	B	0 1	4 1	0 6	IMM						
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd								
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7 6	0 IMM 0						
	BAND #xx:3,@aa:8	B	7 E	abs	7 6	0 IMM 0						
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0				
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM 0		
Bcc	BRA d:8 (BT d:8)	—	4 0	disp								
	BRA d:16 (BT d:16)	—	5 8	0 0	disp							
	BRN d:8 (BF d:8)	—	4 1	disp								
	BRN d:16 (BF d:16)	—	5 8	1 0	disp							

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
Bcc	BHI d:8	—	4 2	disp								
	BHI d:16	—	5 8	2 0	disp							
	BLS d:8	—	4 3	disp								
	BLS d:16	—	5 8	3 0	disp							
	BCC d:8 (BHS d:8)	—	4 4	disp								
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp							
	BCS d:8 (BLO d:8)	—	4 5	disp								
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp							
	BNE d:8	—	4 6	disp								
	BNE d:16	—	5 8	6 0	disp							
	BEQ d:8	—	4 7	disp								
	BEQ d:16	—	5 8	7 0	disp							
	BVC d:8	—	4 8	disp								
	BVC d:16	—	5 8	8 0	disp							
	BVS d:8	—	4 9	disp								
	BVS d:16	—	5 8	9 0	disp							
	BPL d:8	—	4 A	disp								
	BPL d:16	—	5 8	A 0	disp							
	BMI d:8	—	4 B	disp								
	BMI d:16	—	5 8	B 0	disp							
	BGE d:8	—	4 C	disp								
	BGE d:16	—	5 8	C 0	disp							
	BLT d:8	—	4 D	disp								
	BLT d:16	—	5 8	D 0	disp							
	BGT d:8	—	4 E	disp								
	BGT d:16	—	5 8	E 0	disp							
	BLE d:8	—	4 F	disp								
	BLE d:16	—	5 8	F 0	disp							

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BCLR	BCLR #xx:3,Rd	B	7	2	0 IMM	rd														
	BCLR #xx:3,@ERd	B	7	D	0 erd	0	7	2	0 IMM	0										
	BCLR #xx:3,@aa:8	B	7	F	abs		7	2	0 IMM	0										
	BCLR #xx:3,@aa:16	B	6	A	1	8	abs				7	2	0 IMM	0						
	BCLR #xx:3,@aa:32	B	6	A	3	8	abs								7	2	0 IMM	0		
	BCLR Rn,Rd	B	6	2	rn	rd														
	BCLR Rn,@ERd	B	7	D	0 erd	0	6	2	rn	0										
	BCLR Rn,@aa:8	B	7	F	abs		6	2	rn	0										
	BCLR Rn,@aa:16	B	6	A	1	8	abs				6	2	rn	0						
	BCLR Rn,@aa:32	B	6	A	3	8	abs								6	2	rn	0		
BIAND	BIAND #xx:3,Rd	B	7	6	1 IMM	rd														
	BIAND #xx:3,@ERd	B	7	C	0 erd	0	7	6	1 IMM	0										
	BIAND #xx:3,@aa:8	B	7	E	abs		7	6	1 IMM	0										
	BIAND #xx:3,@aa:16	B	6	A	1	0	abs				7	6	1 IMM	0						
	BIAND #xx:3,@aa:32	B	6	A	3	0	abs								7	6	1 IMM	0		
BILD	BILD #xx:3,Rd	B	7	7	1 IMM	rd														
	BILD #xx:3,@ERd	B	7	C	0 erd	0	7	7	1 IMM	0										
	BILD #xx:3,@aa:8	B	7	E	abs		7	7	1 IMM	0										
	BILD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	1 IMM	0						
	BILD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	1 IMM	0		
BIOR	BIOR #xx:3,Rd	B	7	4	1 IMM	rd														
	BIOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	1 IMM	0										
	BIOR #xx:3,@aa:8	B	7	E	abs		7	4	1 IMM	0										
	BIOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	1 IMM	0						
	BIOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	1 IMM	0		

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BIST	BIST #xx:3,Rd	B	6	7	1 IMM	rd														
	BIST #xx:3,@ERd	B	7	D	0 erd	0	6	7	1 IMM	0										
	BIST #xx:3,@aa:8	B	7	F	abs		6	7	1 IMM	0										
	BIST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	1 IMM	0						
	BIST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	1 IMM	0		
BIXOR	BIXOR #xx:3,Rd	B	7	5	1 IMM	rd														
	BIXOR #xx:3,@ERd	B	7	C	0 erd	0	7	5	1 IMM	0										
	BIXOR #xx:3,@aa:8	B	7	E	abs		7	5	1 IMM	0										
	BIXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	1 IMM	0						
	BIXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	1 IMM	0		
BLD	BLD #xx:3,Rd	B	7	7	0 IMM	rd														
	BLD #xx:3,@ERd	B	7	C	0 erd	0	7	7	0 IMM	0										
	BLD #xx:3,@aa:8	B	7	E	abs		7	7	0 IMM	0										
	BLD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	0 IMM	0						
	BLD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	0 IMM	0		
BNOT	BNOT #xx:3,Rd	B	7	1	0 IMM	rd														
	BNOT #xx:3,@ERd	B	7	D	0 erd	0	7	1	0 IMM	0										
	BNOT #xx:3,@aa:8	B	7	F	abs		7	1	0 IMM	0										
	BNOT #xx:3,@aa:16	B	6	A	1	8	abs				7	1	0 IMM	0						
	BNOT #xx:3,@aa:32	B	6	A	3	8	abs								7	1	0 IMM	0		
	BNOT Rn,Rd	B	6	1	rn	rd														
	BNOT Rn,@ERd	B	7	D	0 erd	0	6	1	rn	0										
	BNOT Rn,@aa:8	B	7	F	abs		6	1	rn	0										
	BNOT Rn,@aa:16	B	6	A	1	8	abs				6	1	rn	0						
	BNOT Rn,@aa:32	B	6	A	3	8	abs								6	1	rn	0		

Instruction	Mnemonic	Size	Instruction Format																			
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10th byte	
BOR	BOR #xx:3,Rd	B	7	4	0 IMM	rd																
	BOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	0 IMM	0												
	BOR #xx:3,@aa:8	B	7	E	abs		7	4	0 IMM	0												
	BOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	0 IMM	0								
	BOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	0 IMM	0				
BSET	BSET #xx:3,Rd	B	7	0	0 IMM	rd																
	BSET #xx:3,@ERd	B	7	D	0 erd	0	7	0	0 IMM	0												
	BSET #xx:3,@aa:8	B	7	F	abs		7	0	0 IMM	0												
	BSET #xx:3,@aa:16	B	6	A	1	8	abs				7	0	0 IMM	0								
	BSET #xx:3,@aa:32	B	6	A	3	8	abs								7	0	0 IMM	0				
	BSET Rn,Rd	B	6	0	rn	rd																
	BSET Rn,@ERd	B	7	D	0 erd	0	6	0	rn	0												
	BSET Rn,@aa:8	B	7	F	abs		6	0	rn	0												
	BSET Rn,@aa:16	B	6	A	1	8	abs				6	0	rn	0								
	BSET Rn,@aa:32	B	6	A	3	8	abs								6	0	rn	0				
BSR	BSR d:8	—	5	5	disp																	
	BSR d:16	—	5	C	0	0	disp															
BST	BST #xx:3,Rd	B	6	7	0 IMM	rd																
	BST #xx:3,@ERd	B	7	D	0 erd	0	6	7	0 IMM	0												
	BST #xx:3,@aa:8	B	7	F	abs		6	7	0 IMM	0												
	BST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	0 IMM	0								
	BST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	0 IMM	0				
BTST	BTST #xx:3,Rd	B	7	3	0 IMM	rd																
	BTST #xx:3,@ERd	B	7	C	0 erd	0	7	3	0 IMM	0												
	BTST #xx:3,@aa:8	B	7	E	abs		7	3	0 IMM	0												
	BTST #xx:3,@aa:16	B	6	A	1	0	abs				7	3	0 IMM	0								
	BTST #xx:3,@aa:32	B	6	A	3	0	abs								7	3	0 IMM	0				
	BTST Rn,Rd	B	6	3	rn	rd																
	BTST Rn,@ERd	B	7	C	0 erd	0	6	3	rn	0												

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BTST	BTST Rn,@aa:8	B	7	E	abs		6	3	rn	0										
	BTST Rn,@aa:16	B	6	A	1	0	abs				6	3	rn	0						
	BTST Rn,@aa:32	B	6	A	3	0	abs								6	3	rn	0		
BXOR	BXOR #xx:3,Rd	B	7	5	0	IMM	rd													
	BXOR #xx:3,@ERd	B	7	C	0	erd	0	7	5	0	IMM	0								
	BXOR #xx:3,@aa:8	B	7	E	abs		7	5	0	IMM	0									
	BXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	0	IMM	0					
	BXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	0	IMM	0	
CLRMAC	CLRMAC	—	0	1	A	0														
CMP	CMP.B #xx:8,Rd	B	A	rd	IMM															
	CMP.B Rs,Rd	B	1	C	rs	rd														
	CMP.W #xx:16,Rd	W	7	9	2	rd	IMM													
	CMP.W Rs,Rd	W	1	D	rs	rd														
	CMP.L #xx:32,ERd	L	7	A	2	0	erd	IMM												
	CMP.L ERs,ERd	L	1	F	1	ers	0	erd												
DAA	DAA Rd	B	0	F	0	rd														
DAS	DAS Rd	B	1	F	0	rd														
DEC	DEC.B Rd	B	1	A	0	rd														
	DEC.W #1,Rd	W	1	B	5	rd														
	DEC.W #2,Rd	W	1	B	D	rd														
	DEC.L #1,ERd	L	1	B	7	0	erd													
	DEC.L #2,ERd	L	1	B	F	0	erd													
DIVXS	DIVXS.B Rs,Rd	B	0	1	D	0	5	1	rs	rd										
	DIVXS.W Rs,ERd	W	0	1	D	0	5	3	rs	0	erd									
DIVXU	DIVXU.B Rs,Rd	B	5	1	rs	rd														
	DIVXU.W Rs,ERd	W	5	3	rs	0	erd													
EEPMOV	EEPMOV.B	—	7	B	5	C	5	9	8	F										
	EEPMOV.W	—	7	B	D	4	5	9	8	F										

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
EXTS	EXTS.W Rd	W	1	7	D	rd														
	EXTS.L ERd	L	1	7	F	0 erd														
EXTU	EXTU.W Rd	W	1	7	5	rd														
	EXTU.L ERd	L	1	7	7	0 erd														
INC	INC.B Rd	B	0	A	0	rd														
	INC.W #1,Rd	W	0	B	5	rd														
	INC.W #2,Rd	W	0	B	D	rd														
	INC.L #1,ERd	L	0	B	7	0 erd														
	INC.L #2,ERd	L	0	B	F	0 erd														
JMP	JMP @ERn	—	5	9	0 ern	0														
	JMP @aa:24	—	5	A	abs															
	JMP @@aa:8	—	5	B	abs															
JSR	JSR @ERn	—	5	D	0 ern	0														
	JSR @aa:24	—	5	E	abs															
	JSR @@aa:8	—	5	F	abs															
LDC	LDC #xx:8,CCR	B	0	7	IMM															
	LDC #xx:8,EXR	B	0	1	4	1	0	7	IMM											
	LDC Rs,CCR	B	0	3	0	rs														
	LDC Rs,EXR	B	0	3	1	rs														
	LDC @ERs,CCR	W	0	1	4	0	6	9	0 ers	0										
	LDC @ERs,EXR	W	0	1	4	1	6	9	0 ers	0										
	LDC @(d:16,ERs),CCR	W	0	1	4	0	6	F	0 ers	0	disp									
	LDC @(d:16,ERs),EXR	W	0	1	4	1	6	F	0 ers	0	disp									
	LDC @(d:32,ERs),CCR	W	0	1	4	0	7	8	0 ers	0	6	B	2	0	disp					
	LDC @(d:32,ERs),EXR	W	0	1	4	1	7	8	0 ers	0	6	B	2	0	disp					
	LDC @ERs+,CCR	W	0	1	4	0	6	D	0 ers	0										
	LDC @ERs+,EXR	W	0	1	4	1	6	D	0 ers	0										
	LDC @aa:16,CCR	W	0	1	4	0	6	B	0	0	abs									
	LDC @aa:16,EXR	W	0	1	4	1	6	B	0	0	abs									

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
LDC	LDC @aa:32,CCR	W	0	1	4	0	6	B	2	0	abs					
	LDC @aa:32,EXR	W	0	1	4	1	6	B	2	0	abs					
LDM	LDM.L @SP+, (ERn-ERn+1)	L	0	1	1	0	6	D	7e	0 erm+1						
	LDM.L @SP+, (ERn-ERn+2)	L	0	1	2	0	6	D	7	0 erm+2						
	LDM.L @SP+, (ERn-ERn+3)	L	0	1	3	0	6	D	7	0 erm+3						
LDMAC	LDMAC ERs,MACH	L	0	3	2	0 ers										
	LDMAC ERs,MACL	L	0	3	3	0 ers										
MAC	MAC @ERn+,@ERm+	—	0	1	6	0	6	D	0 ern	0 erm						
MOV	MOV.B #xx:8,Rd	B	F	rd	IMM											
	MOV.B Rs,Rd	B	0	C	rs	rd										
	MOV.B @ERs,Rd	B	6	8	0 ers	rd										
	MOV.B @(d:16,ERs),Rd	B	6	E	0 ers	rd	disp									
	MOV.B @(d:32,ERs),Rd	B	7	8	0 ers	0	6	A	2	rd	disp					
	MOV.B @ERs+,Rd	B	6	C	0 ers	rd										
	MOV.B @aa:8,Rd	B	2	rd	abs											
	MOV.B @aa:16,Rd	B	6	A	0	rd	abs									
	MOV.B @aa:32,Rd	B	6	A	2	rd	**abs**									
	MOV.B Rs,@ERd	B	6	8	1 erd	rs										
	MOV.B Rs,@(d:16,ERd)	B	6	E	1 erd	rs	disp									
	MOV.B Rs,@(d:32,ERd)	B	7	8	0 erd	0	6	A	A	rs	disp					
	MOV.B Rs,@-ERd	B	6	C	1 erd	rs										
	MOV.B Rs,@aa:8	B	3	rs	abs											
	MOV.B Rs,@aa :16	B	6	A	8	rs	abs									
	MOV.B Rs,@aa:32	B	6	A	A	rs	abs									
	MOV.W #xx:16,Rd	W	7	9	0	rd	IMM									
	MOV.W Rs,Rd	W	0	D	rs	rd										
	MOV.W @ERs,Rd	W	6	9	0 ers	rd										
	MOV.W @(d:16,ERs),Rd	W	6	F	0 ers	rd	disp									
	MOV.W @(d:32,ERs),Rd	W	7	8	0 ers	0	6	B	2	rd	disp					

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
MOV	MOV.W @ERs+,Rd	W	6	D	0	ers	rd													
	MOV.W @aa:16,Rd	W	6	B	0	rd	abs													
	MOV.W @aa:32,Rd	W	6	B	2	rd	abs													
	MOV.W Rs,@ERd	W	6	9	1	erd	rs													
	MOV.W Rs,@(d:16,ERd)	W	6	F	1	erd	rs	disp												
	MOV.W Rs,@(d:32,ERd)	W	7	8	0	erd	0	6	B	A	rs	disp								
	MOV.W Rs,@-ERd	W	6	D	1	erd	rs													
	MOV.W Rs,@aa:16	W	6	B	8	rs	abs													
	MOV.W Rs,@aa:32	W	6	B	A	rs	abs													
	MOV.L #xx:32,Rd	L	7	A	0	0	erd	IMM												
	MOV.L ERs,ERd	L	0	F	1	ers	0	erd												
	MOV.L @ERs,ERd	L	0	1	0	0	6	9	0	ers	0	erd								
	MOV.L @(d:16,ERs),ERd	L	0	1	0	0	6	F	0	ers	0	erd	disp							
	MOV.L @(d:32,ERs),ERd	L	0	1	0	0	7	8	0	ers	0	6	B	2	0	erd	disp			
	MOV.L @ERs+,ERd	L	0	1	0	0	6	D	0	ers	0	erd								
	MOV.L @aa:16 ,ERd	L	0	1	0	0	6	B	0	0	erd	abs								
	MOV.L @aa:32 ,ERd	L	0	1	0	0	6	B	2	0	erd	abs								
	MOV.L ERs,@ERd	L	0	1	0	0	6	9	1	erd	0	ers								
	MOV.L ERs,@(d:16,ERd)	L	0	1	0	0	6	F	1	erd	0	ers	disp							
	MOV.L ERs,@(d:32,ERd)*[1]	L	0	1	0	0	7	8	0	erd	0	6	B	A	0	ers	disp			
	MOV.L ERs,@-ERd	L	0	1	0	0	6	D	1	erd	0	ers								
	MOV.L ERs,@aa:16	L	0	1	0	0	6	B	8	0	ers	abs								
	MOV.L ERs,@aa:32	L	0	1	0	0	6	B	A	0	ers	abs								
MOVFPE	MOVFPE @aa:16,Rd	B	Cannot be used in the H8S/2633 Series																	
MOVTPE	MOVTPE Rs,@aa:16	B																		
MULXS	MULXS.B Rs,Rd	B	0	1	C	0	5	0	rs	rd										
	MULXS.W Rs,ERd	W	0	1	C	0	5	2	rs	0	erd									
MULXU	MULXU.B Rs,Rd	B	5	0	rs	rd														
	MULXU.W Rs,ERd	W	5	2	rs	0	erd													

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10th byte	
NEG	NEG.B Rd	B	1	7	8	rd														
	NEG.W Rd	W	1	7	9	rd														
	NEG.L ERd	L	1	7	B	0 erd														
NOP	NOP	—	0	0	0	0														
NOT	NOT.B Rd	B	1	7	0	rd														
	NOT.W Rd	W	1	7	1	rd														
	NOT.L ERd	L	1	7	3	0 erd														
OR	OR.B #xx:8,Rd	B	C	rd	IMM															
	OR.B Rs,Rd	B	1	4	rs	rd														
	OR.W #xx:16,Rd	W	7	9	4	rd	IMM													
	OR.W Rs,Rd	W	6	4	rs	rd														
	OR.L #xx:32,ERd	L	7	A	4	0 erd	IMM													
	OR.L ERs,ERd	L	0	1	F	0	6	4	0 ers		0 erd									
ORC	ORC #xx:8,CCR	B	0	4	IMM															
	ORC #xx:8,EXR	B	0	1	4	1	0	4	IMM											
POP	POP.W Rn	W	6	D	7	rn														
	POP.L ERn	L	0	1	0	0	6	D	7	0 ern										
PUSH	PUSH.W Rn	W	6	D	F	rn														
	PUSH.L ERn	L	0	1	0	0	6	D	F	0 ern										
ROTL	ROTL.B Rd	B	1	2	8	rd														
	ROTL.B #2, Rd	B	1	2	C	rd														
	ROTL.W Rd	W	1	2	9	rd														
	ROTL.W #2, Rd	W	1	2	D	rd														
	ROTL.L ERd	L	1	2	B	0 erd														
	ROTL.L #2, ERd	L	1	2	F	0 erd														

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ROTR	ROTR.B Rd	B	1	3	8	rd								
	ROTR.B #2, Rd	B	1	3	C	rd								
	ROTR.W Rd	W	1	3	9	rd								
	ROTR.W #2, Rd	W	1	3	D	rd								
	ROTR.L ERd	L	1	3	B	0 erd								
	ROTR.L #2, ERd	L	1	3	F	0 erd								
ROTXL	ROTXL.B Rd	B	1	2	0	rd								
	ROTXL.B #2, Rd	B	1	2	4	rd								
	ROTXL.W Rd	W	1	2	1	rd								
	ROTXL.W #2, Rd	W	1	2	5	rd								
	ROTXL.L ERd	L	1	2	3	0 erd								
	ROTXL.L #2, ERd	L	1	2	7	0 erd								
ROTXR	ROTXR.B Rd	B	1	3	0	rd								
	ROTXR.B #2, Rd	B	1	3	4	rd								
	ROTXR.W Rd	W	1	3	1	rd								
	ROTXR.W #2, Rd	W	1	3	5	rd								
	ROTXR.L ERd	L	1	3	3	0 erd								
	ROTXR.L #2, ERd	L	1	3	7	0 erd								
RTE	RTE	—	5	6	7	0								
RTS	RTS	—	5	4	7	0								
SHAL	SHAL.B Rd	B	1	0	8	rd								
	SHAL.B #2, Rd	B	1	0	C	rd								
	SHAL.W Rd	W	1	0	9	rd								
	SHAL.W #2, Rd	W	1	0	D	rd								
	SHAL.L ERd	L	1	0	B	0 erd								
	SHAL.L #2, ERd	L	1	0	F	0 erd								

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte			5th byte				6th byte			7th byte	8th byte	9th byte	10th byte
SHAR	SHAR.B Rd	B	1	1	8	rd																
	SHAR.B #2, Rd	B	1	1	C	rd																
	SHAR.W Rd	W	1	1	9	rd																
	SHAR.W #2, Rd	W	1	1	D	rd																
	SHAR.L ERd	L	1	1	B	0 erd																
	SHAR.L #2, ERd	L	1	1	F	0 erd																
SHLL	SHLL.B Rd	B	1	0	0	rd																
	SHLL.B #2, Rd	B	1	0	4	rd																
	SHLL.W Rd	W	1	0	1	rd																
	SHLL.W #2, Rd	W	1	0	5	rd																
	SHLL.L ERd	L	1	0	3	0 erd																
	SHLL.L #2, ERd	L	1	0	7	0 erd																
SHLR	SHLR.B Rd	B	1	1	0	rd																
	SHLR.B #2, Rd	B	1	1	4	rd																
	SHLR.W Rd	W	1	1	1	rd																
	SHLR.W #2, Rd	W	1	1	5	rd																
	SHLR.L ERd	L	1	1	3	0 erd																
	SHLR.L #2, ERd	L	1	1	7	0 erd																
SLEEP	SLEEP	—	0	1	8	0																
STC	STC.B CCR,Rd	B	0	2	0	rd																
	STC.B EXR,Rd	B	0	2	1	rd																
	STC.W CCR,@ERd	W	0	1	4	0	6	9	1 erd	0												
	STC.W EXR,@ERd	W	0	1	4	1	6	9	1 erd	0												
	STC.W CCR,@(d:16,ERd)	W	0	1	4	0	6	F	1 erd	0	disp											
	STC.W EXR,@(d:16,ERd)	W	0	1	4	1	6	F	1 erd	0	disp											
	STC.W CCR,@(d:32,ERd)	W	0	1	4	0	7	8	0 erd	0	6	B	A	0	disp							
	STC.W EXR,@(d:32,ERd)	W	0	1	4	1	7	8	0 erd	0	6	B	A	0	disp							
	STC.W CCR,@-ERd	W	0	1	4	0	6	D	1 erd	0												
	STC.W EXR,@-ERd	W	0	1	4	1	6	D	1 erd	0												

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte		
STC	STC.W CCR,@aa:16	W	0	1	4	0	6	B	8	0	abs							
	STC.W EXR,@aa:16	W	0	1	4	1	6	B	8	0	abs							
	STC.W CCR,@aa:32	W	0	1	4	0	6	B	A	0	abs							
	STC.W EXR,@aa:32	W	0	1	4	1	6	B	A	0	abs							
STM	STM.L(ERn-ERn+1), @-SP	L	0	1	1	0	6	D	F	0 ern								
	STM.L (ERn-ERn+2), @-SP	L	0	1	2	0	6	D	F	0 ern								
	STM.L (ERn-ERn+3), @-SP	L	0	1	3	0	6	D	F	0 ern								
STMAC	STMAC MACH,ERd	L	0	2	2	0 ers												
	STMAC MACL,ERd	L	0	2	3	0 ers												
SUB	SUB.B Rs,Rd	B	1	8	rs	rd												
	SUB.W #xx:16,Rd	W	7	9	3	rd	IMM											
	SUB.W Rs,Rd	W	1	9	rs	rd												
	SUB.L #xx:32,ERd	L	7	A	3	0 erd	IMM											
	SUB.L ERs,ERd	L	1	A	1 ers	0 erd												
SUBS	SUBS #1,ERd	L	1	B	0	0 erd												
	SUBS #2,ERd	L	1	B	8	0 erd												
	SUBS #4,ERd	L	1	B	9	0 erd												
SUBX	SUBX #xx:8,Rd	B	B	rd	IMM													
	SUBX Rs,Rd	B	1	E	rs	rd												
TAS*2	TAS @ERd	B	0	1	E	0	7	B	0 erd	C								
TRAPA	TRAPA #x:2	—	5	7	00 IMM	0												
XOR	XOR.B #xx:8,Rd	B	D	rd	IMM													
	XOR.B Rs,Rd	B	1	5	rs	rd												
	XOR.W #xx:16,Rd	W	7	9	5	rd	IMM											
	XOR.W Rs,Rd	W	6	5	rs	rd												
	XOR.L #xx:32,ERd	L	7	A	5	0 erd	IMM											
	XOR.L ERs,ERd	L	0	1	F	0	6	5	0 ers	0 erd								

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
XORC	XORC #xx:8,CCR	B	0 5	IMM								
	XORC #xx:8,EXR	B	0 1	4 1	0 5	IMM						

Notes: *1 Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.
*2 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

Legend

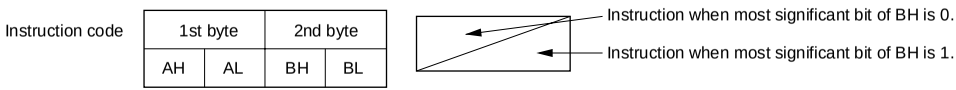
IMM:	Immediate data (2, 3, 8, 16, or 32 bits)
abs:	Absolute address (8, 16, 24, or 32 bits)
disp:	Displacement (8, 16, or 32 bits)
rs, rd, rn:	Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to operand symbols Rs, Rd,and Rn.)
ers, erd, ern, erm:	Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and erm correspond to operand symbols ERs, ERd, ERn, and ERm.)

The register fields specify general registers as follows.

Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

Table A-3 shows the operation code map.

Table A-3 Operation Code Map (1)



AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A.3(2)	STC / *STMAC	LDC / *LDMAC	ORC	XORC	ANDC	LDC	ADD		Table A.3(2)	Table A.3(2)	MOV		ADDX	Table A.3(2)
1	Table A.3(2)	Table A.3(2)	Table A.3(2)	Table A.3(2)	OR	XOR	AND	Table A.3(2)	SUB		Table A.3(2)	Table A.3(2)	CMP		SUBX	Table A.3(2)
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.3(2)	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV		Table A.3(2)	MOV				
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A.3(2)	Table A.3(2)	EEPMOV	Table A.3(3)			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Note: * Cannot be used in the H8S/2633 Series.

Table A-3 Operation Code Map (2)

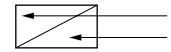
Instruction code	1st byte		2nd byte	
	AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV	LDM / STM			LDC / STC		MAC*		SLEEP		CLRMAC*		Table A.3(3)	Table A.3(3)	TAS	Table A.3(3)
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL				SHLL	SHAL			SHAL				SHAL
11	SHLR			SHLR				SHLR	SHAR			SHAR				SHAR
12	ROTXL			ROTXL				ROTXL	ROTL			ROTL				ROTL
13	ROTXR			ROTXR				ROTXR	ROTR			ROTR				ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
6A	MOV	Table A.3(4)	MOV	Table A.3(4)	MOVFPE*				MOV		MOV		MOVTPE*			
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Note: * Cannot be used in the H8S/2633 Series.

Table A-3 Operation Code Map (3)

Instruction code	1st byte		2nd byte		3rd byte		4th byte	
	AH	AL	BH	BL	CH	CL	DH	DL



CL / AH AL BH BL CH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01C05	MULXS		MULXS													
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7Cr06 *1				BTST												
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST								
7Dr07 *1	BSET	BNOT	BCLR													
7Eaa6 *2				BTST												
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST								
7Faa7 *2	BSET	BNOT	BCLR													

Notes: *1 r is the register specification field.
*2 aa is the absolute address specification.

Table A-3 Operation Code Map (4)

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction when most significant bit of FH is 0.
Instruction when most significant bit of FH is 1.

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A10aaaa6*				BTST												
6A10aaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A18aaaa7*																

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH	HL

Instruction when most significant bit of HH is 0.
Instruction when most significant bit of HH is 1.

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A30aaaaaaaa6*				BTST												
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A38aaaaaaaa7*																

Note: * aa is the absolute address specification.

A.4 Number of States Required for Instruction Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the CPU. Table A-5 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A-4 indicates the number of states required for each cycle. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Advanced mode, program code and stack located in external memory, on-chip supporting modules accessed in two states with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFC7:8
   From table A-5:

   $I = L = 2, \quad J = K = M = N = 0$

   From table A-4:

   $S_I = 4, \quad S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30
   From table A-5:

   $I = J = K = 2, \quad L = M = N = 0$

   From table A-4:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Table A-4 Number of States per Cycle

		Access Conditions					
		On-Chip Supporting Module		External Device			
				8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	2	4	6 + 2m	2	3 + m
Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		2		2	3 + m		
Word data access	S_M		4		4	6 + 2m		
Internal operation	S_N	1	1	1	1	1	1	1

Legend

m: Number of wait states inserted into external device access

Table A-5 Number of Cycles in Instruction Execution

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8,Rd	1					
	ADD.B Rs,Rd	1					
	ADD.W #xx:16,Rd	2					
	ADD.W Rs,Rd	1					
	ADD.L #xx:32,ERd	3					
	ADD.L ERs,ERd	1					
ADDS	ADDS #1/2/4,ERd	1					
ADDX	ADDX #xx:8,Rd	1					
	ADDX Rs,Rd	1					
AND	AND.B #xx:8,Rd	1					
	AND.B Rs,Rd	1					
	AND.W #xx:16,Rd	2					
	AND.W Rs,Rd	1					
	AND.L #xx:32,ERd	3					
	AND.L ERs,ERd	2					
ANDC	ANDC #xx:8,CCR	1					
	ANDC #xx:8,EXR	2					
BAND	BAND #xx:3,Rd	1					
	BAND #xx:3,@ERd	2			1		
	BAND #xx:3,@aa:8	2			1		
	BAND #xx:3,@aa:16	3			1		
	BAND #xx:3,@aa:32	4			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
BCLR	BCLR #xx:3,Rd	1					
	BCLR #xx:3,@ERd	2			2		
	BCLR #xx:3,@aa:8	2			2		
	BCLR #xx:3,@aa:16	3			2		
	BCLR #xx:3,@aa:32	4			2		
	BCLR Rn,Rd	1					
	BCLR Rn,@ERd	2			2		
	BCLR Rn,@aa:8	2			2		
	BCLR Rn,@aa:16	3			2		
	BCLR Rn,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
BIAND	BIAND #xx:3,Rd	1					
	BIAND #xx:3,@ERd	2			1		
	BIAND #xx:3,@aa:8	2			1		
	BIAND #xx:3,@aa:16	3			1		
	BIAND #xx:3,@aa:32	4			1		
BILD	BILD #xx:3,Rd	1					
	BILD #xx:3,@ERd	2			1		
	BILD #xx:3,@aa:8	2			1		
	BILD #xx:3,@aa:16	3			1		
	BILD #xx:3,@aa:32	4			1		
BIOR	BIOR #xx:8,Rd	1					
	BIOR #xx:8,@ERd	2			1		
	BIOR #xx:8,@aa:8	2			1		
	BIOR #xx:8,@aa:16	3			1		
	BIOR #xx:8,@aa:32	4			1		
BIST	BIST #xx:3,Rd	1					
	BIST #xx:3,@ERd	2			2		
	BIST #xx:3,@aa:8	2			2		
	BIST #xx:3,@aa:16	3			2		
	BIST #xx:3,@aa:32	4			2		
BIXOR	BIXOR #xx:3,Rd	1					
	BIXOR #xx:3,@ERd	2			1		
	BIXOR #xx:3,@aa:8	2			1		
	BIXOR #xx:3,@aa:16	3			1		
	BIXOR #xx:3,@aa:32	4			1		
BLD	BLD #xx:3,Rd	1					
	BLD #xx:3,@ERd	2			1		
	BLD #xx:3,@aa:8	2			1		
	BLD #xx:3,@aa:16	3			1		
	BLD #xx:3,@aa:32	4			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BNOT	BNOT #xx:3,Rd	1					
	BNOT #xx:3,@ERd	2			2		
	BNOT #xx:3,@aa:8	2			2		
	BNOT #xx:3,@aa:16	3			2		
	BNOT #xx:3,@aa:32	4			2		
	BNOT Rn,Rd	1					
	BNOT Rn,@ERd	2			2		
	BNOT Rn,@aa:8	2			2		
	BNOT Rn,@aa:16	3			2		
	BNOT Rn,@aa:32	4			2		
BOR	BOR #xx:3,Rd	1					
	BOR #xx:3,@ERd	2			1		
	BOR #xx:3,@aa:8	2			1		
	BOR #xx:3,@aa:16	3			1		
	BOR #xx:3,@aa:32	4			1		
BSET	BSET #xx:3,Rd	1					
	BSET #xx:3,@ERd	2			2		
	BSET #xx:3,@aa:8	2			2		
	BSET #xx:3,@aa:16	3			2		
	BSET #xx:3,@aa:32	4			2		
	BSET Rn,Rd	1					
	BSET Rn,@ERd	2			2		
	BSET Rn,@aa:8	2			2		
	BSET Rn,@aa:16	3			2		
	BSET Rn,@aa:32	4			2		
BSR	BSR d:8	2		2			
	BSR d:16	2		2			1
BST	BST #xx:3,Rd	1					
	BST #xx:3,@ERd	2			2		
	BST #xx:3,@aa:8	2			2		
	BST #xx:3,@aa:16	3			2		
	BST #xx:3,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3,Rd	1					
	BTST #xx:3,@ERd	2			1		
	BTST #xx:3,@aa:8	2			1		
	BTST #xx:3,@aa:16	3			1		
	BTST #xx:3,@aa:32	4			1		
	BTST Rn,Rd	1					
	BTST Rn,@ERd	2			1		
	BTST Rn,@aa:8	2			1		
	BTST Rn,@aa:16	3			1		
	BTST Rn,@aa:32	4			1		
BXOR	BXOR #xx:3,Rd	1					
	BXOR #xx:3,@ERd	2			1		
	BXOR #xx:3,@aa:8	2			1		
	BXOR #xx:3,@aa:16	3			1		
	BXOR #xx:3,@aa:32	4			1		
CLRMAC	CLRMAC	1					1*[3]
CMP	CMP.B #xx:8,Rd	1					
	CMP.B Rs,Rd	1					
	CMP.W #xx:16,Rd	2					
	CMP.W Rs,Rd	1					
	CMP.L #xx:32,ERd	3					
	CMP.L ERs,ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2,Rd	1					
	DEC.L #1/2,ERd	1					
DIVXS	DIVXS.B Rs,Rd	2					11
	DIVXS.W Rs,ERd	2					19
DIVXU	DIVXU.B Rs,Rd	1					11
	DIVXU.W Rs,ERd	1					19

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
EEPMOV	EEPMOV.B	2			2n+2*2		
	EEPMOV.W	2			2n+2*2		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					
INC	INC.B Rd	1					
	INC.W #1/2,Rd	1					
	INC.L #1/2,ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					1
	JMP @@aa:8	2	2				1
JSR	JSR @ERn	2		2			
	JSR @aa:24	2		2			1
	JSR @@aa:8	2	2	2			
LDC	LDC #xx:8,CCR	1					
	LDC #xx:8,EXR	2					
	LDC Rs,CCR	1					
	LDC Rs,EXR	1					
	LDC @ERs,CCR	2				1	
	LDC @ERs,EXR	2				1	
	LDC @(d:16,ERs),CCR	3				1	
	LDC @(d:16,ERs),EXR	3				1	
	LDC @(d:32,ERs),CCR	5				1	
	LDC @(d:32,ERs),EXR	5				1	
	LDC @ERs+,CCR	2				1	1
	LDC @ERs+,EXR	2				1	1
	LDC @aa:16,CCR	3				1	
	LDC @aa:16,EXR	3				1	
	LDC @aa:32,CCR	4				1	
	LDC @aa:32,EXR	4				1	

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
LDM	LDM.L @SP+, (ERn-ERn+1)	2		4			1
	LDM.L @SP+, (ERn-ERn+2)	2		6			1
	LDM.L @SP+, (ERn-ERn+3)	2		8			1
LDMAC	LDMAC ERs,MACH	1					1*[3]
	LDMAC ERs,MACL	1					1*[3]
MAC	MAC @ERn+,@ERm+	2				2	
MOV	MOV.B #xx:8,Rd	1					
	MOV.B Rs,Rd	1					
	MOV.B @ERs,Rd	1			1		
	MOV.B @(d:16,ERs),Rd	2			1		
	MOV.B @(d:32,ERs),Rd	4			1		
	MOV.B @ERs+,Rd	1			1		1
	MOV.B @aa:8,Rd	1			1		
	MOV.B @aa:16,Rd	2			1		
	MOV.B @aa:32,Rd	3			1		
	MOV.B Rs,@ERd	1			1		
	MOV.B Rs,@(d:16,ERd)	2			1		
	MOV.B Rs,@(d:32,ERd)	4			1		
	MOV.B Rs,@-ERd	1			1		1
	MOV.B Rs,@aa:8	1			1		
	MOV.B Rs,@aa:16	2			1		
	MOV.B Rs,@aa:32	3			1		
	MOV.W #xx:16,Rd	2					
	MOV.W Rs,Rd	1					
	MOV.W @ERs,Rd	1				1	
	MOV.W @(d:16,ERs),Rd	2				1	
	MOV.W @(d:32,ERs),Rd	4				1	
	MOV.W @ERs+,Rd	1				1	1
	MOV.W @aa:16,Rd	2				1	
	MOV.W @aa:32,Rd	3				1	
	MOV.W Rs,@ERd	1				1	

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.W Rs,@(d:16,ERd)	2				1	
	MOV.W Rs,@(d:32,ERd)	4				1	
	MOV.W Rs,@-ERd	1				1	1
	MOV.W Rs,@aa:16	2				1	
	MOV.W Rs,@aa:32	3				1	
	MOV.L #xx:32,ERd	3					
	MOV.L ERs,ERd	1					
	MOV.L @ERs,ERd	2				2	
	MOV.L @(d:16,ERs),ERd	3				2	
	MOV.L @(d:32,ERs),ERd	5				2	
	MOV.L @ERs+,ERd	2				2	1
	MOV.L @aa:16,ERd	3				2	
	MOV.L @aa:32,ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs,@(d:16,ERd)	3				2	
	MOV.L ERs,@(d:32,ERd)	5				2	
	MOV.L ERs,@-ERd	2				2	1
	MOV.L ERs,@aa:16	3				2	
	MOV.L ERs,@aa:32	4				2	
MOVFPE	MOVFPE @:aa:16,Rd	Can not be used in the H8S/2633 Series					
MOVTPE	MOVTPE Rs,@:aa:16						
MULXS	MULXS.B Rs,Rd	2					2*[3]
	MULXS.W Rs,ERd	2					3*[3]
MULXU	MULXU.B Rs,Rd	1					2*[3]
	MULXU.W Rs,ERd	1					3*[3]
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
OR	OR.B #xx:8,Rd	1					
	OR.B Rs,Rd	1					
	OR.W #xx:16,Rd	2					
	OR.W Rs,Rd	1					
	OR.L #xx:32,ERd	3					
	OR.L ERs,ERd	2					
ORC	ORC #xx:8,CCR	1					
	ORC #xx:8,EXR	2					
POP	POP.W Rn	1				1	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.B #2,Rd	1					
	ROTR.W Rd	1					
	ROTR.W #2,Rd	1					
	ROTR.L ERd	1					
	ROTR.L #2,ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.B #2,Rd	1					
	ROTXL.W Rd	1					
	ROTXL.W #2,Rd	1					
	ROTXL.L ERd	1					
	ROTXL.L #2,ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.B #2,Rd	1					
	ROTXR.W Rd	1					
	ROTXR.W #2,Rd	1					
	ROTXR.L ERd	1					
	ROTXR.L #2,ERd	1					
RTE	RTE	2		2/3*[1]			1
RTS	RTS	2		2			1
SHAL	SHAL.B Rd	1					
	SHAL.B #2,Rd	1					
	SHAL.W Rd	1					
	SHAL.W #2,Rd	1					
	SHAL.L ERd	1					
	SHAL.L #2,ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.B #2,Rd	1					
	SHAR.W Rd	1					
	SHAR.W #2,Rd	1					
	SHAR.L ERd	1					
	SHAR.L #2,ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.B #2,Rd	1					
	SHLL.W Rd	1					
	SHLL.W #2,Rd	1					
	SHLL.L ERd	1					
	SHLL.L #2,ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.B #2,Rd	1					
	SHLR.W Rd	1					
	SHLR.W #2,Rd	1					
	SHLR.L ERd	1					
	SHLR.L #2,ERd	1					
SLEEP	SLEEP	1					1

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
STC	STC.B CCR,Rd	1					
	STC.B EXR,Rd	1					
	STC.W CCR,@ERd	2				1	
	STC.W EXR,@ERd	2				1	
	STC.W CCR,@(d:16,ERd)	3				1	
	STC.W EXR,@(d:16,ERd)	3				1	
	STC.W CCR,@(d:32,ERd)	5				1	
	STC.W EXR,@(d:32,ERd)	5				1	
	STC.W CCR,@-ERd	2				1	1
	STC.W EXR,@-ERd	2				1	1
	STC.W CCR,@aa:16	3				1	
	STC.W EXR,@aa:16	3				1	
	STC.W CCR,@aa:32	4				1	
	STC.W EXR,@aa:32	4				1	
STM	STM.L (ERn-ERn+1), @-SP	2		4			1
	STM.L (ERn-ERn+2), @-SP	2		6			1
	STM.L (ERn-ERn+3), @-SP	2		8			1
STMAC	STMAC MACH,ERd	1					*3
	STMAC MACL,ERd	1					*3
SUB	SUB.B Rs,Rd	1					
	SUB.W #xx:16,Rd	2					
	SUB.W Rs,Rd	1					
	SUB.L #xx:32,ERd	3					
	SUB.L ERs,ERd	1					
SUBS	SUBS #1/2/4,ERd	1					
SUBX	SUBX #xx:8,Rd	1					
	SUBX Rs,Rd	1					
TAS*4	TAS @ERd	2			2		
TRAPA	TRAPA #x:2	2	2	2/3*1			2

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
XOR	XOR.B #xx:8,Rd	1					
	XOR.B Rs,Rd	1					
	XOR.W #xx:16,Rd	2					
	XOR.W Rs,Rd	1					
	XOR.L #xx:32,ERd	3					
	XOR.L ERs,ERd	2					
XORC	XORC #xx:8,CCR	1					
	XORC #xx:8,EXR	2					

Notes: *1 2 when EXR is invalid, 3 when EXR is valid.

*2 When n bytes of data are transferred.

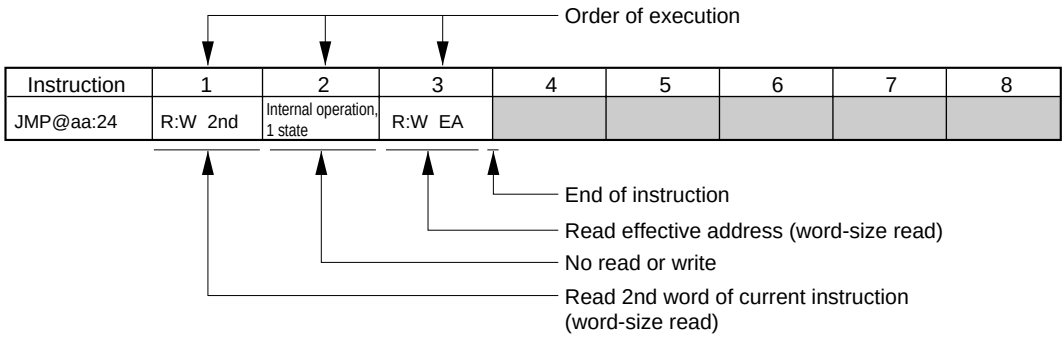
*3 An internal operation may require between 0 and 3 additional states, depending on the preceding instruction.

*4 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

A.5 Bus States During Instruction Execution

Table A-6 indicates the types of cycles that occur during instruction execution by the CPU. See table A-4 for the number of states per cycle.

How to Read the Table:



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Address of next instruction
EA	Effective address
VEC	Vector address

Figure A-1 shows timing waveforms for the address bus and the $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ signals during execution of the above instruction with an 8-bit bus, using three-state access with no wait states.

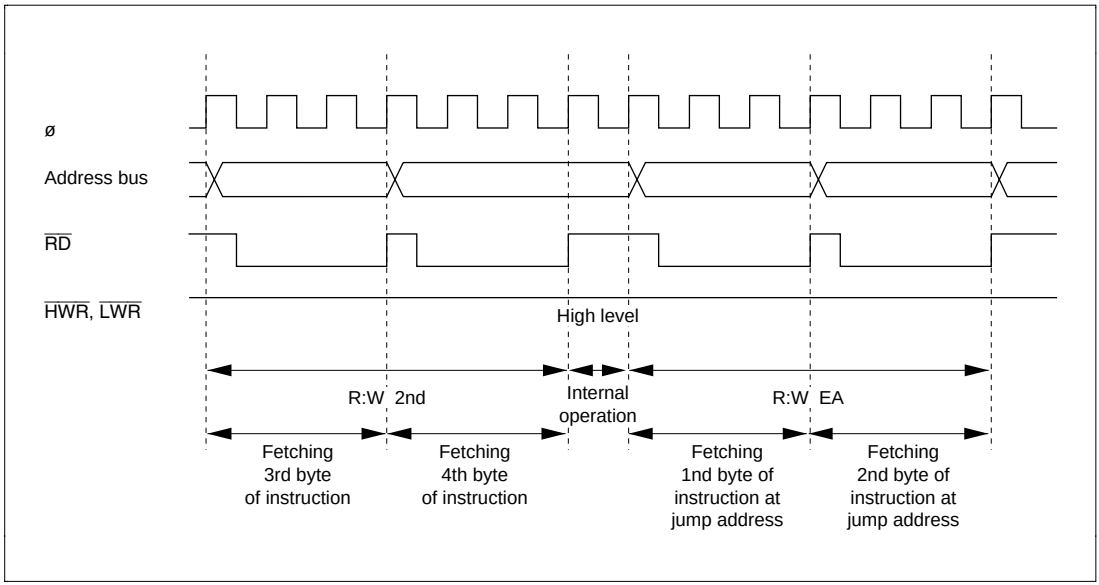


Figure A-1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Table A-6 Instruction Execution Cycles

Instruction	1	2	3	4	5	6	7	8	9
ADD.B #xx:8,Rd	R:W NEXT								
ADD.B Rs,Rd	R:W NEXT								
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT							
ADD.W Rs,Rd	R:W NEXT								
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
ADD.L ERs,ERd	R:W NEXT								
ADDS #1/2/4,ERd	R:W NEXT								
ADDX #xx:8,Rd	R:W NEXT								
ADDX Rs,Rd	R:W NEXT								
AND.B #xx:8,Rd	R:W NEXT								
AND.B Rs,Rd	R:W NEXT								
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT							
AND.W Rs,Rd	R:W NEXT								
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
AND.L ERs,ERd	R:W 2nd	R:W NEXT							
ANDC #xx:8,CCR	R:W NEXT								
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT							
BAND #xx:3,Rd	R:W NEXT								
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BRA d:8 (BT d:8)	R:W NEXT	R:W EA							
BRN d:8 (BF d:8)	R:W NEXT	R:W EA							
BHI d:8	R:W NEXT	R:W EA							
BLS d:8	R:W NEXT	R:W EA							
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA							
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA							
BNE d:8	R:W NEXT	R:W EA							
BEQ d:8	R:W NEXT	R:W EA							
BVC d:8	R:W NEXT	R:W EA							
BVS d:8	R:W NEXT	R:W EA							
BPL d:8	R:W NEXT	R:W EA							
BMI d:8	R:W NEXT	R:W EA							
BGE d:8	R:W NEXT	R:W EA							
BLT d:8	R:W NEXT	R:W EA							
BGT d:8	R:W NEXT	R:W EA							

Instruction	1	2	3	4	5	6	7	8	9
BLE d:8	R:W NEXT	R:W EA							
BRA d:16 (BT d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BRN d:16 (BF d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BHI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCC d:16 (BHS d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BCS d:16 (BLO d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BNE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BEQ d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVC d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BPL d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BMI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCLR #xx:3,Rd	R:W NEXT								
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				

Instruction	1	2	3	4	5	6	7	8	9
BCLR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,Rd	R:W NEXT								
BCLR Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BIAND #xx:3,Rd	R:W NEXT								
BIAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BILD #xx:3,Rd	R:W NEXT								
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIOR #xx:3,Rd	R:W NEXT								
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIST #xx:3,Rd	R:W NEXT								
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BIXOR #xx:3,Rd	R:W NEXT								
BIXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BLD #xx:3,Rd	R:W NEXT								
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BNOT #xx:3,Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,Rd	R:W NEXT								
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BOR #xx:3,Rd	R:W NEXT								
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BSET #xx:3,Rd	R:W NEXT								
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,Rd	R:W NEXT								
BSET Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSR d:8	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
BSR d:16	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
BST #xx:3,Rd	R:W NEXT								
BST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BTST #xx:3,Rd	R:W NEXT								
BTST #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						

Instruction	1	2	3	4	5	6	7	8	9
BTST #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BTST Rn,Rd	R:W NEXT								
BTST Rn,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BXOR #xx:3,Rd	R:W NEXT								
BXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
CLRMAC	R:W NEXT	Internal operation, 1 state							
CMP.B #xx:8,Rd	R:W NEXT								
CMP.B Rs,Rd	R:W NEXT								
CMP.W #xx:16,Rd	R:W 2nd	R:W NEXT							
CMP.W Rs,Rd	R:W NEXT								
CMP.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
CMP.L ERs,ERd	R:W NEXT								
DAA Rd	R:W NEXT								
DAS Rd	R:W NEXT								
DEC.B Rd	R:W NEXT								
DEC.W #1/2,Rd	R:W NEXT								
DEC.L #1/2,ERd	R:W NEXT								
DIVXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states						
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states						
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states							
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states							
EEPMOV.B	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EEPMOV.W	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EXTS.W Rd	R:W NEXT			← Repeated n times*2 →					
EXTS.L ERd	R:W NEXT								
EXTU.W Rd	R:W NEXT								
EXTU.L ERd	R:W NEXT								
INC.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
INC.W #1/2,Rd	R:W NEXT								
INC.L #1/2,ERd	R:W NEXT								
JMP @ERn	R:W NEXT	R:W EA							
JMP @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA						
JMP @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA				
JSR @ERn	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
JSR @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
JSR @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M stack (H)	W:W stack (L)	R:W EA			
LDC #xx:8,CCR	R:W NEXT								
LDC #xx:8,EXR	R:W 2nd	R:W NEXT							
LDC Rs,CCR	R:W NEXT								
LDC Rs,EXR	R:W NEXT								
LDC @ERs,CCR	R:W 2nd	R:W NEXT	R:W EA						
LDC @ERs,EXR	R:W 2nd	R:W NEXT	R:W EA						
LDC @(d:16,ERs),CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @(d:16,ERs),EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @(d:32,ERs),CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC @(d:32,ERs),EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC @ERs+,CCR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @aa:16,CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:16,EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:32,CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDC @aa:32,EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDM.L @SP+, (ERn–ERn+1)	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDM.L @SP+,(ERn–ERn+2)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDM.L @SP+,(ERn–ERn+3)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDMAC ERs,MACH	R:W NEXT	Internal operation, 1 state		←Repeated n times *3→					

Instruction	1	2	3	4	5	6	7	8	9
LDMAC ERs,MACL	R:W NEXT	Internal operation, 1 state							
MAC @ERn+,@ERm+	R:W 2nd	R:W NEXT	R:W EAh	R:W EAm					
MOV.B #xx:8,Rd	R:W NEXT								
MOV.B Rs,Rd	R:W NEXT								
MOV.B @ERs,Rd	R:W NEXT	R:B EA							
MOV.B @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:B EA						
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA				
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:B EA						
MOV.B @aa:8,Rd	R:W NEXT	R:B EA							
MOV.B @aa:16,Rd	R:W 2nd	R:W NEXT	R:B EA						
MOV.B @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.B Rs,@ERd	R:W NEXT	W:B EA							
MOV.B Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA				
MOV.B Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:B EA						
MOV.B Rs,@aa:8	R:W NEXT	W:B EA							
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA					
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT							
MOV.W Rs,Rd	R:W NEXT								
MOV.W @ERs,Rd	R:W NEXT	R:W EA							
MOV.W @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
MOV.W @ERs+, Rd	R:W NEXT	Internal operation, 1 state	R:W EA						
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.W Rs,@ERd	R:W NEXT	W:W EA							
MOV.W Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA						
MOV.W Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:E 4th	R:W NEXT	W:W EA				
MOV.W Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:W EA						
MOV.W Rs,@aa:16	R:W 2nd	R:W NEXT	W:W EA						
MOV.W Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					

Instruction	1	2	3	4	5	6	7	8	9
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
MOV.L ERs,ERd	R:W NEXT								
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2					
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
MOV.L @aa:16,ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @aa:32,ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2					
MOV.L ERs,@(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@–ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2			
MOVFPE @aa:16,Rd	Cannot be used in the H8S/2633 Series								
MOVTPE Rs,@aa:16									
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 2 states						
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 3 states						
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 2 states							
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 3 states							
NEG.B Rd	R:W NEXT								
NEG.W Rd	R:W NEXT								
NEG.L ERd	R:W NEXT								
NOP	R:W NEXT								
NOT.B Rd	R:W NEXT								
NOT.W Rd	R:W NEXT								
NOT.L ERd	R:W NEXT								
OR.B #xx:8,Rd	R:W NEXT								
OR.B Rs,Rd	R:W NEXT								
OR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
OR.W Rs,Rd	R:W NEXT								
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
OR.L ERs,ERd	R:W 2nd	R:W NEXT							
ORC #xx:8,CCR	R:W NEXT								
ORC #xx:8,EXR	R:W 2nd	R:W NEXT							

Instruction	1	2	3	4	5	6	7	8	9
POP.W Rn	R:W NEXT	Internal operation, 1 state	R:W EA						
POP.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
PUSH.W Rn	R:W NEXT	Internal operation, 1 state	W:W EA						
PUSH.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
ROTL.B Rd	R:W NEXT								
ROTL.B #2,Rd	R:W NEXT								
ROTL.W Rd	R:W NEXT								
ROTL.W #2,Rd	R:W NEXT								
ROTL.L ERd	R:W NEXT								
ROTL.L #2,ERd	R:W NEXT								
ROTR.B Rd	R:W NEXT								
ROTR.B #2,Rd	R:W NEXT								
ROTR.W Rd	R:W NEXT								
ROTR.W #2,Rd	R:W NEXT								
ROTR.L ERd	R:W NEXT								
ROTR.L #2,ERd	R:W NEXT								
ROTXL.B Rd	R:W NEXT								
ROTXL.B #2,Rd	R:W NEXT								
ROTXL.W Rd	R:W NEXT								
ROTXL.W #2,Rd	R:W NEXT								
ROTXL.L ERd	R:W NEXT								
ROTXL.L #2,ERd	R:W NEXT								
ROTXR.B Rd	R:W NEXT								
ROTXR.B #2,Rd	R:W NEXT								
ROTXR.W Rd	R:W NEXT								
ROTXR.W #2,Rd	R:W NEXT								
ROTXR.L ERd	R:W NEXT								
ROTXR.L #2,ERd	R:W NEXT								
RTE	R:W NEXT	R:W stack (EXR)	R:W stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*4			
RTS	R:W NEXT	R:W:M stack (H)	R:W stack (L) 1 state	Internal operation,	R:W*4				
SHAL.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
SHAL.B #2,Rd	R:W NEXT								
SHAL.W Rd	R:W NEXT								
SHAL.W #2,Rd	R:W NEXT								
SHAL.L ERd	R:W NEXT								
SHAL.L #2,ERd	R:W NEXT								
SHAR.B Rd	R:W NEXT								
SHAR.B #2,Rd	R:W NEXT								
SHAR.W Rd	R:W NEXT								
SHAR.W #2,Rd	R:W NEXT								
SHAR.L ERd	R:W NEXT								
SHAR.L #2,ERd	R:W NEXT								
SHLL.B Rd	R:W NEXT								
SHLL.B #2,Rd	R:W NEXT								
SHLL.W Rd	R:W NEXT								
SHLL.W #2,Rd	R:W NEXT								
SHLL.L ERd	R:W NEXT								
SHLL.L #2,ERd	R:W NEXT								
SHLR.B Rd	R:W NEXT								
SHLR.B #2,Rd	R:W NEXT								
SHLR.W Rd	R:W NEXT								
SHLR.W #2,Rd	R:W NEXT								
SHLR.L ERd	R:W NEXT								
SHLR.L #2,ERd	R:W NEXT								
SLEEP	R:W NEXT	Internal operation:M							
STC CCR,Rd	R:W NEXT								
STC EXR,Rd	R:W NEXT								
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC EXR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC CCR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC EXR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC CCR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC EXR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					

Instruction	1	2	3	4	5	6	7	8	9
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STM.L(ERn–ERn+1),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+2),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+3),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STMAC MACH,ERd	R:W NEXT								
STMAC MACL,ERd	R:W NEXT								
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
SUB.L ERs,ERd	R:W NEXT								
SUBS #1/2/4,ERd	R:W NEXT								
SUBX #xx:8,Rd	R:W NEXT								
SUBX Rs,Rd	R:W NEXT								
TAS @ERd*8	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA					
TRAPA #x:2	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
XOR.L ERs,ERd	R:W 2nd	R:W NEXT							
XORC #xx:8,CCR	R:W NEXT								
XORC #xx:8,EXR	R:W 2nd	R:W NEXT							

Instruction	1	2	3	4	5	6	7	8	9
Reset exception handling	R:W VEC	R:W VEC+2	Internal operation, 1 state	R:W*5					
Interrupt exception handling	R:W*6	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7

Notes: *1 EAs is the contents of ER5. EAd is the contents of ER6.
*2 EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution of the instruction. n is the initial value of R4L or R4. If n = 0, these bus cycles are not executed.
*3 Repeated two times to save or restore two registers, three times for three registers, or four times for four registers.
*4 Start address after return.
*5 Start address of the program.
*6 Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or software standby mode the read operation is replaced by an internal operation.
*7 Start address of the interrupt-handling routine.
*8 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

A.6 Condition Code Modification

This section indicates the effect of each CPU instruction on the condition code. The notation used in the table is defined below.

m = { 31 for longword operands; 15 for word operands; 7 for byte operands }

Si	The i-th bit of the source operand
Di	The i-th bit of the destination operand
Ri	The i-th bit of the result
Dn	The specified bit in the destination operand
—	Not affected
↕	Modified according to the result of the instruction (see definition)
0	Always cleared to 0
1	Always set to 1
*	Undetermined (no guaranteed value)
Z'	Z flag before instruction execution
C'	C flag before instruction execution

Table A-7 Condition Code Modification

Instruction	H	N	Z	V	C	Definition
ADD	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
AND	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
ANDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
BAND	—	—	—	—	↕	C = C' · Dn
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BIAND	—	—	—	—	↕	C = C' · $\overline{Dn}$
BILD	—	—	—	—	↕	C = $\overline{Dn}$
BIOR	—	—	—	—	↕	C = C' + $\overline{Dn}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{C'}$ · $\overline{Dn}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{Dn}$
BXOR	—	—	—	—	↕	C = C' · $\overline{Dn}$ + $\overline{C'}$ · Dn
CLRMAC	—	—	—	—	—	

Instruction	H	N	Z	V	C	Definition
CMP	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
DAA	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
INC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC	—	—	—	—	—	
MAC	—	—	—	—	—	

Instruction	H	N	Z	V	C	Definition
MOV	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MOVFPE						Can not be used in H8S/2633 Series
MOVTPE						
MULXS	—	↕	↕	—	—	N = R2m
						Z = $\overline{R2m} \cdot \overline{R2m-1} \cdot \cdots\cdots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4
						N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						V = Dm · Rm
						C = Dm + Rm
NOP	—	—	—	—	—	
NOT	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
OR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
POP	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
PUSH	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ROTL	—	↕	↕	0	↕	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTR	—	↕	↕	0	↕	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						C = D0 (1-bit shift) or C = D1 (2-bit shift)

Instruction	H	N	Z	V	C	Definition
ROTXL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTXR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
RTE	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
SHAL	—	↕	↕	↕	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}}$ (1-bit shift) V = $\overline{Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}}$ (2-bit shift) C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHAR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SHLL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHLR	—	0	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC	—	↕	↕	↕	—	N = 1 if MAC instruction resulted in negative value in MAC register Z = 1 if MAC instruction resulted in zero value in MAC register V = 1 if MAC instruction resulted in overflow

Instruction	H	N	Z	V	C	Definition
SUB	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
SUBS	—	—	—	—	—	
SUBX	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS*	—	↕	↕	0	—	N = Dm
						Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.

Note: * This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

Appendix B Internal I/O Register

B.1 Addresses

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FDAC	DADR2									D/A2, D/A3	8
H'FDAD	DADR3										
H'FDAE	DACR23	DAOE1	DAOE0	DAE	—	—	—	—	—		
H'FDB0	IrCR	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	SCI0, IrDA	8
H'FDB4	SCRX	—	IICX1	IICX0	IICE	FLSHE	—	—	—	IIC	8
H'FDB5	DDCSWR	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0	IIC	8
H'FDB8	DADRAH0/ DACR0	DA13/ TEST	DA12/ PWME	DA11/—	DA10/—	DA9/ OEB	DA8/ OEA	DA7/OS	DA6/CKS	PWM0	8
H'FDB9	DADRAL0	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—		
H'FDBA	DADRBH0/ DACNTH0	DA13/	DA12/	DA11/	DA10/	DA9/	DA8/	DA7/	DA6/		
H'FDBB	DADRBL0/ DACNTL0	DA5/	DA4/	DA3/	DA2/	DA1/	DA0/	CFS/	REGS		
H'FDBC	DADRAH1/ DACR1	DA13/ TEST	DA12/ PWME	DA11/—	DA10/—	DA9/OEB	DA8/OEA	DA7/OS	DA6/CKS	PWM1	8
H'FDBD	DADRAL1	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—		
H'FDBE	DADRBH1/ DACNTH1	DA13/	DA12/	DA11/	DA10/	DA9/	DA8/	DA7/	DA6/		
H'FDBF	DADRBL1/ DACNTL1	DA5/	DA4/	DA3/	DA2/	DA1/	DA0/	CFS/	REGS		
H'FDC0	TCR2	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR2, TMR3	16
H'FDC1	TCR3	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0		
H'FDC2	TCSR2	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FDC3	TCSR3	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FDC4	TCORA2										
H'FDC5	TCORA3										
H'FDC6	TCORB2										
H'FDC7	TCORB3										
H'FDC8	TCNT2										
H'FDC9	TCNT3										
H'FDD0	SMR3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI3, Smart card interface	8
	SMR3	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
H'FDD1	BRR3										
H'FDD2	SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FDD3	TDR3										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FDD4	SSR3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI3, Smart card interface	8
	SSR3	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FDD5	RDR3										
H'FDD6	SCMR3	—	—	—	—	SDIR	SINV	—	SMIF		
H'FDD8	SMR4	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI4, Smart card interface	8
	SMR4	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
H'FDD9	BRR4										
H'FDDA	SCR4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FDDB	TDR4										
H'FDDC	SSR4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR4	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FDDD	RDR4										
H'FDDE	SCMR4	—	—	—	—	SDIR	SINV	—	SMIF		
H'FDE4	SBYCR	SSBY	STS2	SYS1	STS0	OPE	—	—	—	System	8
H'FDE5	SYSCR	MACS	—	INTM1	INTM0	NMIEG	MRESE	—	RAME		
H'FDE6	SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0		
H'FDE7	MDCR	—	—	—	—	—	MDS2	MDS1	MDS0		
H'FDE8	MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0		
H'FDE9	MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0		
H'FDEA	MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0		
H'FDEB	PFCR	CSS07	CSS36	BUZZE	LCASS	AE3	AE2	AE1	AE0		
H'FDEC	LPWRCR	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0		
H'FE00	BARA	—	—	—	—	—	—	—	—	PBC	8
H'FE01		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16		
H'FE02		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8		
H'FE03		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0		
H'FE04	BARB	—	—	—	—	—	—	—	—		
H'FE05		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16		
H'FE06		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8		
H'FE07		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0		
H'FE08	BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA		
H'FE09	BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRB0	CSELB1	CSELB0	BIEB		
H'FE12	ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Interrupt controller	8
H'FE13	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FE14	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E		
H'FE15	ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FE16	DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC	8
H'FE17	DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0		
H'FE18	DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0		
H'FE19	DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0		
H'FE1A	DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0		
H'FE1B	DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0		
H'FE1E	DTCERI	DTCEI7	DTCEI6	DTCEI5	DTCEI4	DTCEI3	DTCEI2	DTCEI1	DTCEI0		
H'FE1F	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FE26	PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG	8
H'FE27	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV		
H'FE28	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8		
H'FE29	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0		
H'FE2A	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8		
H'FE2B	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
H'FE2C	NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8		
H'FE2D	NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0		
H'FE2E	NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8		
H'FE2F	NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0		
H'FE30	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	Port	8
H'FE32	P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
H'FE36	P7DDR	P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR		
H'FE39	PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR		
H'FE3A	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR		
H'FE3B	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR		
H'FE3C	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
H'FE3D	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR		
H'FE3E	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR		
H'FE3F	PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR		
H'FE40	PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR		
H'FE41	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR		
H'FE42	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR		
H'FE43	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
H'FE44	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR		
H'FE46	P3ODR	P37ODR	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR		
H'FE47	PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR		
H'FE48	PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR		
H'FE49	PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FE80	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU3	16
H'FE81	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FE82	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FE83	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FE84	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FE85	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FE86	TCNT3										
H'FE87											
H'FE88	TGR3A										
H'FE89											
H'FE8A	TGR3B										
H'FE8B											
H'FE8C	TGR3C										
H'FE8D											
H'FE8E	TGR3D										
H'FE8F											
H'FE90	TCR4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU4	16
H'FE91	TMDR4	—	—	—	—	MD3	MD2	MD1	MD0		
H'FE92	TIOR4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FE94	TIER4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FE95	TSR4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FE96	TCNT4										
H'FE97											
H'FE98	TGR4A										
H'FE99											
H'FE9A	TGR4B										
H'FE9B											
H'FEA0	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU5	16
H'FEA1	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0		
H'FEA2	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FEA4	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FEA5	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FEA6	TCNT5										
H'FEA7											
H'FEA8	TGR5A										
H'FEA9											
H'FEAA	TGR5B										
H'FEAB											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FEB0	TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU	16
H'FEB1	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0		
H'FEC0	IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	Interrupt controller	8
H'FEC1	IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC2	IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC3	IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC4	IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC5	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC6	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC7	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC8	IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC9	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECA	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECB	IPRL	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECE	IPRO	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus controller	8
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0		
H'FED2	WCRH	W71	W70	W61	W60	W51	W50	W41	W40		
H'FED3	WCRL	W31	W30	W21	W20	W11	W10	W01	W00		
H'FED4	BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMST0		
H'FED5	BCRL	BRLE	BREQOE	—	OES	DDS	RCTS	WDBE	WAITE		
H'FED6	MCR	TPC	BE	RCDM	CW2	MXC1	MXC0	RLW1	RLW0		
H'FED7	DRAMCR	RFSHE	CBRM	RMODE	CMF	CMIE	CKS2	CKS1	CKS0		
H'FED8	RTCNT										
H'FED9	RTCOR										
H'FEDB	RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH	8
H'FEE0	MAR0AH	—	—	—	—	—	—	—	—	DMAC	16
H'FEE1											
H'FEE2	MAR0AL										
H'FEE3											
H'FEE4	IOAR0A										
H'FEE5											
H'FEE6	ETCR0A										
H'FEE7											
H'FEE8	MAR0BH	—	—	—	—	—	—	—	—		
H'FEE9											
H'FEEA	MAR0BL										
H'FEEB											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FEEC	IOAR0B									DMAC	16
H'FEED											
H'FEEE	ETCR0B										
H'FEEF											
H'FEF0	MAR1AH	—	—	—	—	—	—	—	—		
H'FEF1											
H'FEF2	MAR1AL										
H'FEF3											
H'FEF4	IOAR1A										
H'FEF5											
H'FEF6	ETCR1A										
H'FEF7											
H'FEF8	MAR1BH	—	—	—	—	—	—	—	—		
H'FEF9											
H'FEFA	MAR1BL										
H'FEFB											
H'FEFC	IOAR1B										
H'FEFD											
H'FEFE	ETCR1B										
H'FEFF											
H'FF00	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	Port	8
H'FF01	—	—	—	—	—	—	—	—	—		
H'FF02	P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
H'FF04	—	—	—	—	—	—	—	—	—		
H'FF05	—	—	—	—	—	—	—	—	—		
H'FF06	P7DR	P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR		
H'FF07	—	—	—	—	—	—	—	—	—		
H'FF09	PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR		
H'FF0A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR		
H'FF0B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
H'FF0C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
H'FF0D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR		
H'FF0E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR		
H'FF0F	PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR		
H'FF10	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU0	16
H'FF11	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FF12	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	TPU0	16
H'FF13	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FF14	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FF15	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FF16	TCNT0										
H'FF17											
H'FF18	TGR0A										
H'FF19											
H'FF1A	TGR0B										
H'FF1B											
H'FF1C	TGR0C										
H'FF1D											
H'FF1E	TGR0D										
H'FF1F											
H'FF20	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU1	16
H'FF21	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF22	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF24	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF25	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF26	TCNT1										
H'FF27											
H'FF28	TGR1A										
H'FF29											
H'FF2A	TGR1B										
H'FF2B											
H'FF30	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU2	16
H'FF31	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF32	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF34	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF35	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF36	TCNT2										
H'FF37											
H'FF38	TGR2A										
H'FF39											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FF3A	TGR2B									TPU2	16
H'FF3B											
H'FF60	DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	DMAC	8
H'FF61	DMATCR	—	—	TEE1	TEE0	—	—	—	—		
H'FF62	DMACR0A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		16
H'FF63	DMACR0B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF64	DMACR1A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF65	DMACR1B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF66	DMABCRH	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A		
H'FF67	DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A		
H'FF68	TCR0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR0, TMR1	16
H'FF69	TCR1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0		
H'FF6A	TCSR0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0		
H'FF6B	TCSR1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FF6C	TCORA0										
H'FF6D	TCORA1										
H'FF6E	TCORB0										
H'FF6F	TCORB1										
H'FF70	TCNT0										
H'FF71	TCNT1										
H'FF74 (write)	TCSR0/ TCNT0	OVF/	WT/$\overline{IT}$/	TME/	—/	—/	CKS2/	CKS1/	CKS0/	WDT0	16
H'FF75 (read)	TCNT0										
H'FF76 (write)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—		
H'FF77 (read)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—		
H'FF78	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0, IIC0, Smart card interface	8
	SMR0	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
	ICCR0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP		
H'FF79	BRR0										
	ICSR0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB		
H'FF7A	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF7B	TDR0										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FF7C	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI0, IIC0, Smart card interface	8
	SSR0	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FF7D	RDR0										
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR0/ SARX0	ICDR7/ SVAX6	ICDR6/ SVAX5	ICDR5/ SVAX4	ICDR4/ SVAX3	ICDR3/ SVAX2	ICDR2/ SVAX1	ICDR1/ SVAX0	ICDR0/ FSX		
H'FF7F	ICMR0/ SAR0	MLS/ SVA6	WAIT/ SVA5	CKS2/ SVA4	CKS1/ SVA3	CKS0/ SVA2	BC2/ SVA1	BC1/ SVA0	BC0/FS		
H'FF80	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1, IIC1, Smart card interface	
	SMR1	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
	ICCR1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP		
H'FF81	BRR1										
	ICSR1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB		
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF83	TDR1										
H'FF84	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR1	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FF85	RDR1										
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR1/ SARX1	ICDR7/ SVARX6	ICDR6/ SVARX5	ICDR5/ SVARX4	ICDR4/ SVARX3	ICDR3/ SVARX2	ICDR2/ SVARX1	ICDR1/ SVARX0	ICDR0/ FSX		
H'FF87	ICMR1/ SAR1	MLS/ SVA6	WAIT/ SVA5	CKS2/ SVA4	CKS1/ SVA3	CKS0/ SVA2	BC2/ SVA1	BC1/ SVA0	BC0/FS	IIC1	8
H'FF88	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI2, Smart card interface	8
	SMR2	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
H'FF89	BRR2										
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF8B	TDR2										
H'FF8C	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FF8D	RDR2										
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	SMIF		
H'FF90	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D	8
H'FF91	ADDRAL	AD1	AD0	—	—	—	—	—	—		
H'FF92	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width (bits)
H'FF93	ADDRBL	AD1	AD0	—	—	—	—	—	—	A/D	8
H'FF94	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FF95	ADDRCL	AD1	AD0	—	—	—	—	—	—		
H'FF96	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FF97	ADDRDL	AD1	AD0	—	—	—	—	—	—		
H'FF98	ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0		
H'FF99	ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—		
H'FFA2 (write)	TCSR1/ TCNT1	OVF/	WT/$\overline{IT}$/	TME/	PSS/	RST/ $\overline{NMI}$/	CKS2/	CKS1/	CKS0/	WDT1	16
H'FFA3 (read)	TCNT1										
H'FFA4	DADR0									D/A0, D/A1	8
H'FFA5	DADR1										
H'FFA6	DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—		
H'FFA8	FLMCR1	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1	FLASH	8
H'FFA9	FLMCR2	FLER	—	—	—	—	—	—	—		
H'FFAA	EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0		
H'FFAB	EBR2	—	—	—	—	EB11	EB10	EB9	EB8		
H'FFAC	FLPWCR	PDWND	—	—	—	—	—	—	—		
H'FFB0	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	Port	8
H'FFB2	PORT3	P37	P36	P35	P34	P33	P32	P31	P30		
H'FFB3	PORT4	P47	P46	P45	P44	P43	P42	P41	P40		
H'FFB6	PORT7	P77	P76	P75	P74	P73	P72	P71	P70		
H'FFB8	PORT9	P97	P96	P95	P94	P93	P92	P91	P90		
H'FFB9	PORTA	—	—	—	—	PA3	PA2	PA1	PA0		
H'FFBA	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0		
H'FFBB	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0		
H'FFBC	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0		
H'FFBD	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0		
H'FFBE	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0		
H'FFBF	PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0		

B.2 Functions

DADR0—D/A Data Register 0 **H'FFA4** **D/A0**
DADR1—D/A Data Register 1 **H'FFA5** **D/A1**
DADR2—D/A Data Register 2 **H'FDAC** **D/A2**
DADR3—D/A Data Register 3 **H'FDAD** **D/A3**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DACR01—D/A Control Register 01 **H'FFA6** **D/A0, 1**
DACR23—D/A Control Register 23 **H'FDAE** **D/A2, 3**

Bit	:	7	6	5	4	3	2	1	0
		DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—	—

D/A enable

DAOE1	DAOE0	DAE	Description
0	0	*	Disables channel 0, 1 (channel 2, 3) D/A conversion.
	1	0	Enables channel 0 (channel 2) D/A conversion.
			Disables channel 1 (channel 3) D/A conversion.
		1	Enables channel 0, 1 (channel 2, 3) D/A conversion.
1	0	0	Disables channel 0 (channel 2) D/A conversion.
			Enables channel 1 (channel 3)D/A conversion.
		1	Enables channel 0, 1 (channel 2, 3) D/A conversion.
	1	*	Enables channel 0, 1 (channel 2, 3) D/A conversion.

* : Don't care

D/A output enable 0

0	Disables analog output DA0 (DA2).
1	Enables channel 0 D/A conversion. Also enables analog output DA0 (DA2).

D/A output enable 1

0	Disables analog output DA1 (DA3).
1	Enables channel 1 D/A conversion. Also enables analog output DA1 (DA3).

Bit	7	6	5	4	3	2	1	0
	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	—	—	—	—

IrDA clock select 2 to 0

Bit 6 IrCKS2	Bit 5 IrCKS1	Bit 4 IrCKS0	Description
0	0	0	B × 3/16 (3/16ths of bit rate)
		1	ø/2
	1	0	ø/4
		1	ø/8
1	0	0	ø/16
		1	ø/32
	1	0	ø/64
		1	ø/128

IrDA enable

0	TxD0/IrTxD and RxD0/IrRxD pins function as TxD0 and RxD0.
1	TxD0/IrTxD and RxD0/IrRxD pins function as IrTx0 and IrRxD.

SCRX—Serial Control Register X H'FDB4 IIC

Bit	7	6	5	4	3	2	1	0
	—	IICX1	IICX0	IICE	FLSHE	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Flash memory control register enable

I^2C master enable

0	Disables CPU access of I^2C bus interface data register and control register.
1	Enables CPU access of I^2C bus interface data register and control register.

I^2C transfer rate select 1, 0

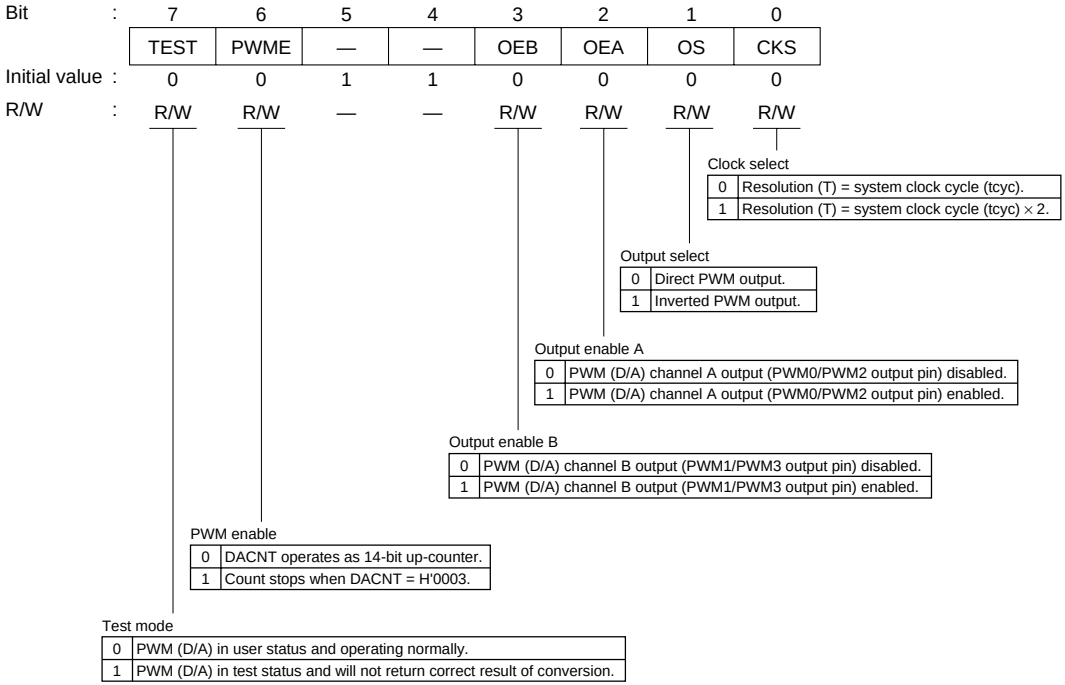
Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	CLR3	CLR2	CLR1	CLR0
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	W*2	W*2	W*2	W*2

Reserved bit

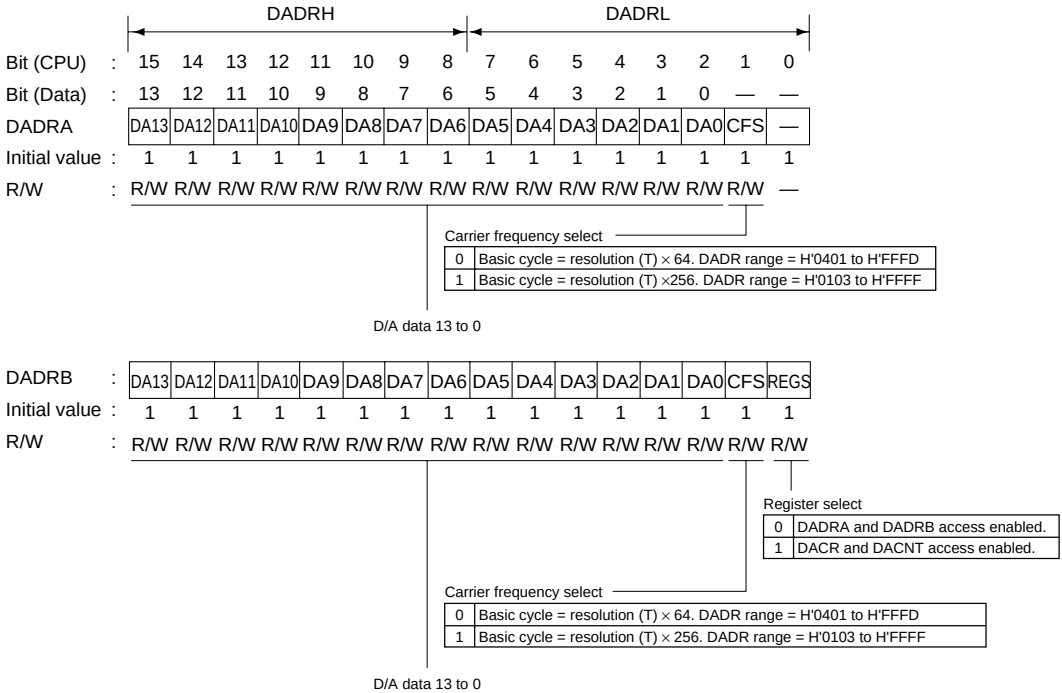
IIC clear 3 to 0

CLR3	CLR2	CLR1	CLR0	
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latch cleared
1	—	—	—	Invalid setting

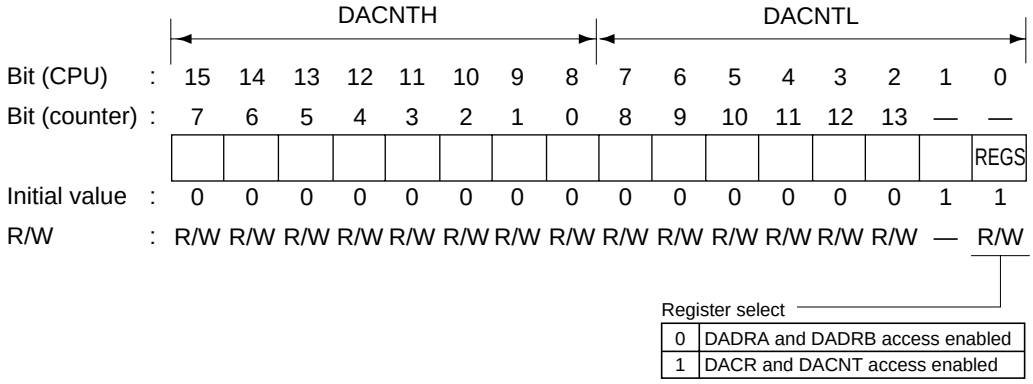
Note: *1 Should always be written with 0.
*2 Always read as 1.


Bit : 7 6 5 4 3 2 1 0
TEST PWME — — OEB OEA OS CKS
Initial value : 0 0 1 1 0 0 0 0
R/W : R/W R/W — — R/W R/W R/W R/W
Clock select
0 Resolution (T) = system clock cycle (tcyc).
1 Resolution (T) = system clock cycle (tcyc) × 2.
Output select
0 Direct PWM output.
1 Inverted PWM output.
Output enable A
0 PWM (D/A) channel A output (PWM0/PWM2 output pin) disabled.
1 PWM (D/A) channel A output (PWM0/PWM2 output pin) enabled.
Output enable B
0 PWM (D/A) channel B output (PWM1/PWM3 output pin) disabled.
1 PWM (D/A) channel B output (PWM1/PWM3 output pin) enabled.
PWM enable
0 DACNT operates as 14-bit up-counter.
1 Count stops when DACNT = H'0003.
Test mode
0 PWM (D/A) in user status and operating normally.
1 PWM (D/A) in test status and will not return correct result of conversion.

DADRAH0—PWM (D/A) Data Register AH0	**H'FDB8**	**PWM0**
DADRAL0—PWM (D/A) Data Register AL0	**H'FDB9**	**PWM0**
DADRBH0—PWM (D/A) Data Register BH0	**H'FDBA**	**PWM0**
DADRBL0—PWM (D/A) Data Register BL0	**H'FDBB**	**PWM0**
DADRAH1—PWM (D/A) Data Register AH1	**H'FDBC**	**PWM1**
DADRAL1—PWM (D/A) Data Register AL1	**H'FDBD**	**PWM1**
DADRBH1—PWM (D/A) Data Register BH1	**H'FDBE**	**PWM1**
DADRBL1—PWM (D/A) Data Register BL1	**H'FDBF**	**PWM1**



DACNTH0—PWM (D/A) Counter H0	**H'FDBA**	**PWM0**
DACNTL0—PWM (D/A) Counter L0	**H'FDBB**	**PWM0**
DACNTH1—PWM (D/A) Counter H1	**H'FDBE**	**PWM1**
DACNTL1—PWM (D/A) Counter L1	**H'FDBF**	**PWM1**



TCR0—Timer Control Register 0 **H'FF68** **TMR0**
TCR1—Timer Control Register 1 **H'FF69** **TMR1**
TCR2—Timer Control Register 2 **H'FDC0** **TMR2**
TCR3—Timer Control Register 3 **H'FDC1** **TMR3**

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Timer overflow interrupt enable

0	OVF interrupt request (OVI) disabled
1	OVF interrupt request (OVI) enabled

Compare match interrupt enable A

0	CMFA interrupt request (CMIA) disabled
1	CMFA interrupt request (CMIA) enabled

Compare match interrupt enable B

0	CMFB interrupt request (CMIB) disabled
1	CMFB interrupt request (CMIB) enabled

Counter clear 1, 0

CCLR1	CCLR0	
0	0	Clearing disabled
	1	Cleared by compare match A
1	0	Cleared by compare match B
	1	Cleared by rising edge of external reset input

Clock select 2 to 0

CKS2	CKS1	CKS0	
0	0	0	Clock input disabled
		1	Internal clock: Counting on falling edge of ø/8
	1	0	Internal clock: Counting on falling edge of ø/64
		1	Internal clock: Counting on falling edge of ø/8192
1	0	0	Channel 0: Counting on TCNT1 overflow signal * Channel 1: Counting on TCNT0 compare match A * Channel 2: Counting on TCNT3 overflow signal * Channel 3: Counting on TCNT2 compare match A *
		1	External clock: Counting on rising edge
	1	0	External clock: Counting on falling edge
		1	External clock: Counting on both rising and falling edges

Note: * No countup clock is generated if the channel 0 (channel 2) clock input is the TCNT1 (TCNT3) overflow signal, and that the channel 1 (channel 3) clock input is the TCNT0 (TCNT2) compare match signal. Do not, therefore, attempt to make such a setting.

TCSR0

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

TCSR1, TCSR3

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	1	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

TCSR2

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Bit 7: Compare match flag B

0	[Clearing] (1) Reading CMFB then writing 0 to CMFB when CMFB=1 (2) When DTC is started by CMIB interrupt and DTC MRB DISEL bit is 0
1	[Setting] When TCNT=TCORB

Bit 6: Compare match flag A

0	[Clearing] (1) Reading CMFA then writing 0 to CMFA when CMFA=1 (2) When DTC is started by CMIA interrupt and DTC MRB DISEL bit is 0
1	[Setting] When TCNT=TCORA

Bit 5: Timer overflow flag

0	[Clearing] Reading OVF then writing 0 to OVF when OVF=1
1	[Setting] When TCNT changes from H'FF to H'00

Bit 4: A/D trigger enable

0	A/D conversion start request by compare match A disabled
1	A/D conversion start request by compare match A enabled

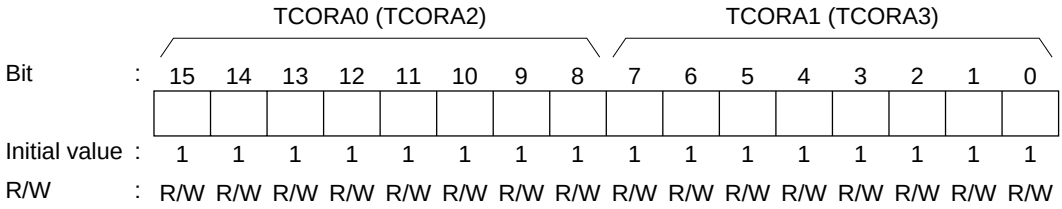
Bits 3 to 0: Output select 3 to 0

OS3	OS2	
0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Inverted output each compare match B (toggle output)

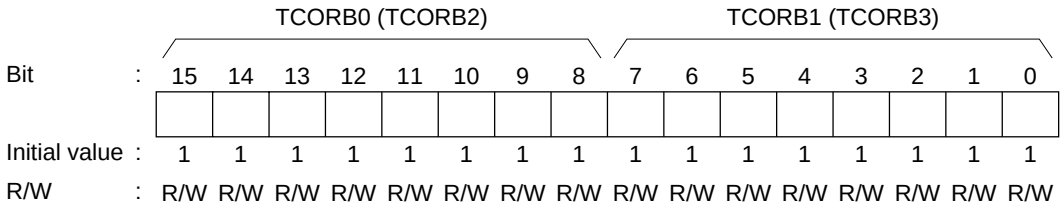
OS1	OS0	
0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Inverted output each compare match A (toggle output)

Note: * Only 0 can be written to bits 7 to 5 (to clear these flags).

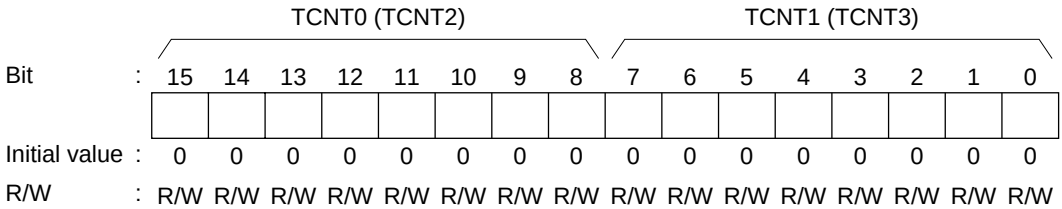
TCORA0—Time Constant Register A0	**H'FF6C**	**TMR0**
TCORA1—Time Constant Register A1	**H'FF6D**	**TMR1**
TCORA2—Time Constant Register A2	**H'FDC4**	**TMR2**
TCORA3—Time Constant Register A3	**H'FDC5**	**TMR3**



TCORB0—Time Constant Register B0	**H'FF6E**	**TMR0**
TCORB1—Time Constant Register B1	**H'FF6F**	**TMR1**
TCORB2—Time Constant Register B2	**H'FDC6**	**TMR2**
TCORB3—Time Constant Register B3	**H'FDC7**	**TMR3**



TCNT0—Timer Counter 0	**H'FF70**	**TMR0**
TCNT1—Timer Counter 1	**H'FF71**	**TMR1**
TCNT2—Timer Counter 2	**H'FDC8**	**TMR2**
TCNT3—Timer Counter 3	**H'FDC9**	**TMR3**



SMR0—Serial Mode Register 0	**H'FF78**	**SCI0**
SMR1—Serial Mode Register 1	**H'FF80**	**SCI1**
SMR2—Serial Mode Register 2	**H'FF88**	**SCI2**
SMR3—Serial Mode Register 3	**H'FDD0**	**SCI3**
SMR4—Serial Mode Register 4	**H'FDD8**	**SCI4**

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 1 and 0

CKS1	CKS0	Description
0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Multiprocessor mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop bit length

0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: *1 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.
*2 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

Communication mode

0	Asynchronous mode
1	Clocked synchronous mode

Bit	7	6	5	4	3	2	1	0
	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Basic clock pulse 1, 0

BCP1	BCP0	
0	0	32 clock
	1	64 clock
1	0	372 clock
	1	256 clock

Block transfer mode

0	Operation of normal smart card interface mode (1) Error signal output, detection, and automatic resending of data; (2) TXI interrupt generated by TEND flag; (3) TEND flag set 12.5etu after start of transmission (after 11.0etu in GSM mode).
1	Operation in block transfer mode (1) No error signal output, detection, or automatic resending of data; (2) TXI interrupt generated by TDRE flag; (3) TEND flag set 11.5etu after start of transmission (after 11.0etu in GSM mode).

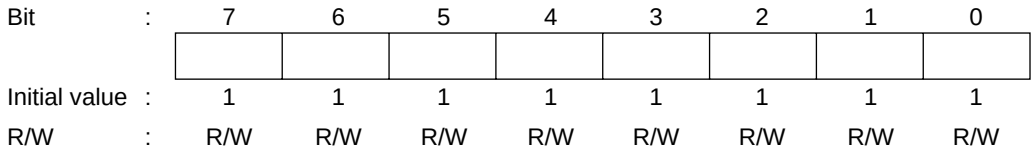
GSM Mode

0	Operation in normal smart card interface mode (1)TEND flag set 12.5etu (11.5etu in block transfer mode) after start of first bit; (2)ON/OFF control only of clock output.
1	Operation in GSM mode smart card interface mode (1)TEND flag set 11.0etu after start of first bit; (2)In addition to ON/OFF control of clock output, High/Low control also enabled (set by SCR).

Note: etu: Elementary Time Unit. The time to send one bit.

Note: * Set bit 5 to 1 when using the smart card interface.

BRR0—Bit Rate Register 0	**H'FF79**	**SCI0**
BRR1—Bit Rate Register 1	**H'FF81**	**SCI1**
BRR2—Bit Rate Register 2	**H'FF89**	**SCI2**
BRR3—Bit Rate Register 3	**H'FDD1**	**SCI3**
BRR4—Bit Rate Register 4	**H'FDD9**	**SCI4**



Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

SCR0—Serial Control Register 0 **H'FF7A** **SCI0**
SCR1—Serial Control Register 1 **H'FF82** **SCI1**
SCR2—Serial Control Register 2 **H'FF8A** **SCI2**
SCR3—Serial Control Register 3 **H'FDD2** **SCI3**
SCR4—Serial Control Register 4 **H'FDDA** **SCI4**

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable 1, 0

Bit 1	Bit 0	Description	
CKE1	CKE0		
0	0	Async mode	Internal clock/SCK pin set as I/O port*1
		Clock sync mode	Internal clock/SCK pin set for sync clock output*1
	1	Async mode	Internal clock/SCK pin set for clock output*2
		Clock sync mode	Internal clock/SCK pin set for sync clock output
1	0	Async mode	External clock/SCK pin set for clock input*3
		Clock sync mode	External clock/SCK pin set for sync clock input
	1	Async mode	External clock/SCK pin set for clock input*3
		Clock sync mode	External clock/SCK pin set for sync clock input

Notes: *1 Initial value
*2 Clock output at same frequency as bit rate
*3 Clock input at 16 times frequency of bit rate

Transmit end interrupt enable

0	Transmit end interrupt (TEI) requests disabled*
1	Transmit end interrupt (TEI) requests enabled*

Note: * To cancel a TEI, clear SSR TDRE flag to 0 after reading TDRE=1, then either clear the TEND flag to 0 or clear the TEIE bit to 0.

Multiprocessor interrupt enable

0	Multiprocessor interrupt disabled (normal receive operations) [Clearing] (1) Clear the MPIE bit to 0; (2) When data MPB=1 is received.
1	Multiprocessor interrupt enabled* Until data is received that the multiprocessor bit = 1, receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and SSR RDRF, FER, and ORER flags cannot be set.

Note: * On reception of receive data that includes MPB=0, the receive data is not sent from the RSR to the RDR, and, on detection of receive errors, the SSR RDRF, FER and ORER flags are not set. On reception of receive data that includes MPB=1, the SSR MPB bit is set to 1 and the MPIE bit is automatically cleared to 0. If an RXI or ERI interrupt request occurs (when the SCR TIE or RIE bit is set to 1), the FER and ORER flags can be set.

Receive enable

0	Disable receive operation.*1
1	Enable receive operation.*2

Notes: *1 Clearing the RE bit has no effect on the RDRF, FER, PER, or ORER flags.
*2 Serial receiving starts on detection of the start bit when in async mode, or on detection of sync clock input in clock sync mode.
Before setting the RE bit to 1, be sure to set the SMR to decide the receive format.

Transmit enable

0	Disable transmit operation.*1
1	Enable transmit operation.*2

Notes: *1 The SSR TDRE flag is set to 1 (fixed).
*2 Transmission starts when, in this state, transmit data is written to TDR and the SSR TDRE flag is cleared to 0. Before setting the TE bit to 1, be sure to set the SMR to decide the transmit format.

Receive interrupt enable

0	Disable receive data full interrupt (RXI) requests and receive error interrupt (ERI) requests.*
1	Enable receive data full interrupt (RXI) requests and receive error interrupt (ERI) requests.

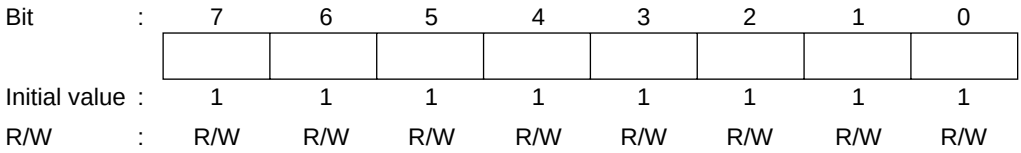
Note: * To cancel RXI and ERI interrupt requests, either clear the RDRF or FER, PER, or ORER flags after reading "1", or clear the RIE bit to 0.

Transmit interrupt enable

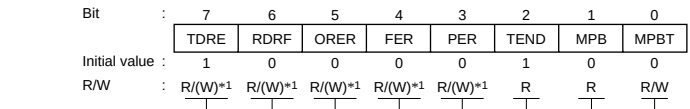
0	Disable transmit data empty interrupt (TXI) requests.
1	Enable transmit data empty interrupt (TXI) requests.

Note: To clear TXI interrupt requests, clear the TDRE flag to 0 after reading "1", or clear the TIE bit to 0.

TDR0—Transmit Data Register 0	**H'FF7B**	**SCI0**
TDR1—Transmit Data Register 1	**H'FF83**	**SCI1**
TDR2—Transmit Data Register 2	**H'FF8B**	**SCI2**
TDR3—Transmit Data Register 3	**H'FDD3**	**SCI3**
TDR4—Transmit Data Register 4	**H'FDDB**	**SCI4**



Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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



Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	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

Multiprocessor bit transfer

0	Transfer data "multiprocessor bit = 0".
1	Transfer data "multiprocessor bit = 1".

Multiprocessor bit

0	[Clearing]* When data "multiprocessor bit = 0" is received.
1	[Setting] When data "multiprocessor bit = 1" is received.

Note: * The existing status is continued when, in multi-processor format, the SCR RE bit is cleared to 0.

Transmit end

0	[Clearing] (1) Writing 0 to TDRE flag after reading TDRE=1; (2) When data is written to TDR by DMAC or DTC by TXI interrupt request.
1	[Setting] (1) When SCR TE bit=0; (2) When TDRE=1 at transfer of last bit of any byte of serial transmit character.

Parity error

0	[Clearing]*1 Writing 0 to PER after reading PER=1;
1	[Setting] When receiving, when the number of 1s in receive data plus parity bit does not match the even or odd parity specified in the SMR O/$\overline{E}$ bit.*2

Notes: *1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
*2 If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Framing error

0	[Clearing] *1 Writing 0 to FER after reading FER=1.
1	[Setting] When SCI checks if the stop bit at the end of receive data is 1 on completion of receiving, the stop bit is found to be 0.*2

Notes: *1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
*2 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Overrun error

0	[Clearing]*1 Writing 0 to ORER after reading ORER=1.
1	[Setting] On completion of next serial receive operation when RDRF=1.*2

Notes: *1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
*2 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Receive data register full

0	[Clearing] (1) Writing 0 to RDRF after reading RDRF=1. (2) After reading RDR data by DMAC or DTC by RXI interrupt request.
1	[Setting] When receive data is sent from RSR to RDR on normal completion of serial receive operation.

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Transmit data register empty

0	[Clearing] (1) Writing 0 to TDRE after reading TDRE=1; (2) When data written to TDR by DMAC or DTC by TXI interrupt request;
1	[Setting] (1) When SCR TE bit=0; (2) When data is sent from TDR to TSR and data can be written to TDR.

Note: Only 0 can be written to these bits (to clear these flags).

RDR0—Receive Data Register 0	**H'FF7D**	**SCI0**
RDR1—Receive Data Register 1	**H'FF85**	**SCI1**
RDR2—Receive Data Register 2	**H'FF8D**	**SCI2**
RDR3—Receive Data Register 3	**H'FDD5**	**SCI3**
RDR4—Receive Data Register 4	**H'FDDD**	**SCI4**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

SCMR0—Smart Card Mode Register 0	**H'FF7E**	**SCI0**
SCMR1—Smart Card Mode Register 1	**H'FF86**	**SCI1**
SCMR2—Smart Card Mode Register 2	**H'FF8E**	**SCI2**
SCMR3—Smart Card Mode Register 3	**H'FDD6**	**SCI3**
SCMR4—Smart Card Mode Register 4	**H'FDDE**	**SCI4**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

Smart card interface mode select

0	Operates as normal SCI (smart card interface function disabled)
1	Enables smart card interface function.

Smart card data invert

0	TDR contents are transmitted without modification Receive data is stored in RDR without modification
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart card data transfer direction

0	Sends TDR contents LSB first; Receive data stored in RDR as LSB first.
1	Sends TDR contents MSB first; Receive data stored in RDR as MSB first.

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	0	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—

Output port enable

0	In software standby mode, watch mode, and during direct transfer, the address bus and bus control signal are in the high-impedance state.
1	In software standby mode, watch mode, and during direct transfer, the address bus and bus control signal remain in the output state.

Standby timer select 2 to 0

STS2	STS1	STS0	
0	0	0	Hold time: 8192 states
		1	Hold time: 16384 states
	1	0	Hold time: 32768 states
		1	Hold time: 65536 states
1	0	0	Hold time: 131072 states
		1	Hold time: 262144 states
	1	0	Reserved
		1	Hold time: 16 states

Software standby

0	When the SLEEP command is executed in high-speed or medium-speed modes, the operation enters sleep mode. When the SLEEP command is executed in sub-active mode, the operation enters sub-sleep mode.
1	When the SLEEP command is executed in high-speed and medium-speed modes, operation enters software standby mode, sub-active mode, and watch mode. When the SLEEP command is executed in sub-active mode, operation enters watch mode and high-speed mode.

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

NMI edge select

0	Interrupt request issued on falling edge of NMI input.
1	Interrupt request issued on rising edge of NMI input.

Interrupt control mode 1, 0

INTM1	INTM0	Interrupt control mode	
0	0	0	Interrupt controlled by bit 1
	1	—	Do not set.
1	0	2	Interrupt controlled by bits 12 to 10 and IPR.
	1	—	Do not set.

MAC saturation

0	Non-saturating calculation for MAC instruction
1	Saturating calculation for MAC instruction

Manual reset select bit

0	Manual reset disabled. Pins P74/TMO2/$\overline{\text{MRES}}$ can be used as P74/TMO2 I/O pins.
1	Manual reset enabled. Pins P74/TMO2/$\overline{\text{MRES}}$ can be used as $\overline{\text{MRES}}$ input pins.

RAM Enable

0	Internal RAM disabled.
1	Internal RAM enabled.

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

System clock select 2 to 0

SCK2	SCK1	SCK0	
0	0	0	Bus master set to high-speed mode.
		1	Medium-speed clock: ø/2
	1	0	Medium-speed clock: ø//4
		1	Medium-speed clock: ø8
1	0	0	Medium-speed clock: ø/16
		1	Medium-speed clock: ø/32
	1	—	—

Frequency multiplier switching mode select

0	Specified multiplier valid after transferring to software standby mode, watch mode, and sub-active mode.
1	Specified multiplier valid immediately after setting value in STC bit.

ø clock output disable

PSTOP	High-speed mode, Medium-speed mode Sub-active mode	Sleep mode Sub-Sleep mode	Software standby mode Watch mode direct transition	Hardware standby mode
0	ø output	ø output	High level (fixed)	High impedance
1	High level (fixed)	High level (fixed)	High level (fixed)	High impedance

MDCR—Mode Control Register **H'FDE7** **System**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	R/W	—	—	—	—	R	R	R

Mode select 2 to 0

Note: * Determined by pins MD2 to MD0.

MSTPCRA—Module Stop Control Register A **H'FDE8** **System**

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	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

Module stop

0	DMAC module stop mode is cleared
1	DMAC module stop mode is set

MSTPCRB—Module Stop Control Register B **H'FDE9** **System**

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	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

Module stop

0	IIC channel 1 module stop mode canceled.
1	IIC channel 1 module stop mode enabled

Module stop

0	IIC channel 0 module stop mode canceled.
1	Channel 0 module stop mode enabled

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	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

Module stop

0	PC brake controller module stop mode canceled.
1	PC brake controller module stop mode enabled.

Bit	:	7	6	5	4	3	2	1	0
		CSS07	CSS36	BUZZE	LCASS	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCAS output pin select bit

0	LCAS signal output from PF2.
1	LCAS signal output from PF6.

BUZZ output enable

0	Functions as PF1 input pin.
1	Functions as BUZZ output pin.

$\overline{CS3}$/$\overline{CS6}$ Select

0	Selects $\overline{CS3}$.
1	Selects $\overline{CS6}$.

$\overline{CS0}$/$\overline{CS7}$ Select

0	Selects $\overline{CS0}$.
1	Selects $\overline{CS7}$.

Address output enable 3 to 0

AE3	AE2	AE1	AE0	
0	0	0	0	A8 to A23 address output disabled.
0	0	0	1	A8 address output enabled. A9 to A23 address output disabled.
0	0	1	0	A8 and A9 address output enabled. A10 to A23 address output disabled.
0	0	1	1	A8 to A10 address output enabled. A11 to A23 address output disabled.
0	1	0	0	A8 to A11 address output enabled. A12 to A23 address output disabled.
0	1	0	1	A8 to A12 address output enabled. A13 to A23 address output disabled.
0	1	1	0	A8 to A13 address output enabled. A14 to A23 address output disabled.
0	1	1	1	A8 to A14 address output enabled. A15 to A23 address output disabled.
1	0	0	0	A8 to A15 address output enabled. A16 to A23 address output disabled.
1	0	0	1	A8 to A16 address output enabled. A17 to A23 address output disabled.
1	0	1	0	A8 to A17 address output enabled. A18 to A23 address output disabled.
1	0	1	1	A8 to A18 address output enabled. A19 to A23 address output disabled.
1	1	0	0	A8 to A19 address output enabled. A20 to A23 address output disabled.
1	1	0	1	A8 to A20 address output enabled. A21 to A23 address output disabled.
1	1	1	0	A8 to A21 address output enabled. A22 and A23 address output disabled.
1	1	1	1	A8 to A23 address output enabled.

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

Bit	7	6	5	4	3	2	1	0
	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Frequency multiplier

STC1	STC0	
0	0	× 1 (initial value)
	1	× 2
1	0	× 4
	1	Do not set.

Note: A system clock frequency multiplied by the multiplication factor (STC1 and STC0) should not exceed the maximum operating frequency defined in section 25 Electrical Characteristics.

Oscillator circuit feedback resistor control bit

0	Feedback resistor ON when main clock operating; OFF when not operation.
1	Feedback resistor OFF.

Subclock enable

0	Subclock generation enabled.
1	Subclock generation disabled.

Noise elimination sampling frequency select

0	Sampling uses ø/32 clock.
1	Sampling uses ø/4 clock.

Low-speed ON flag

0	• When the SLEEP command is executed in high-speed mode or medium-speed mode, operation transfers to sleep mode, software standby mode, or watch mode.* • When the SLEEP command is executed in sub-active mode, operation transfers to watch mode, or directly to high-speed mode. • Operation transfers to high-speed mode after watch mode is canceled.
1	• When the SLEEP command is executed in high-speed mode, operation transfers to watch mode or sub-active mode. • When the SLEEP command is executed in sub-active mode, operation transfers to sub-sleep mode or watch mode. • Operation transfers to sub-active mode immediately watch mode is canceled.

Note: * Always select high-speed mode when transferring to watch mode or sub-active mode.

Direct transfer ON flag

0	• When the SLEEP command is executed in high-speed mode or medium-speed mode, operation transfers to sleep mode, software standby mode, or watch mode. • When the SLEEP command is executed in sub-active mode, operation transfers to sub-sleep mode or watch mode.
1	• When the SLEEP command is executed in high-speed mode or medium-speed mode, operation transfers directly to sub-active mode, or transfers to sleep mode or software standby mode. • When the SLEEP command is executed in sub-active mode, operation transfers directly to high-speed mode or transfers to sub-sleep mode.

Note: * Always select high-speed mode when transferring to watch mode or sub-active mode.

Bit	:	31	…	24	23	22	21	20	19	18	17	16	…	7	6	5	4	3	2	1	0
		—	…	—	BAA 23	BAA 22	BAA 21	BAA 20	BAA 19	BAA 18	BAA 17	BAA 16	…	BAA 7	BAA 6	BAA 5	BAA 4	BAA 3	BAA 2	BAA 1	BAA 0
Initial value	:	Unde-fined	…	Unde-fined	0	0	0	0	0	0	0	0	…	0	0	0	0	0	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

Break address 23 to 0

Note: The bit configuration of BARB is the same as that of BARA.

Bit	7	6	5	4	3	2	1	0
	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA
Initial value	0	0	0	0	0	0	0	0
R/W	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CPU cycle/DTC cycle select A

0	When the CPU is the bus master, PC break performed.
1	When the CPU or DTC is the bus master, PC break performed.

Condition match flag A

0	[Clearing] Writing 0 to CMFA after reading CMFA=1.
1	[Setting] When channel A conditions are true.

Break address mask register A2 to A0

BAMRA 2	BAMRA 1	BAMRA 0	
0	0	0	All bits, without masking BARA, included in break condition.
0	0	1	BAA0 (LSB) masked and not included in break condition.
0	1	0	BAA1 and BAA0 (low 2 bits) masked and not included in break condition.
0	1	1	BAA2 to BAA0 (low 3 bits) masked and not included in break condition.
1	0	0	BAA3 to BAA0 (low 4 bits) masked and not included in break condition.
1	0	1	BAA7 to BAA0 (low 8 bits) masked and not included in break condition.
1	1	0	BAA11 to BAA0 (low 12 bits) masked and not included in break condition.
1	1	1	BAA15 to BAA0 (low 16 bits) masked and not included in break condition.

Break condition select

CSELA1	CSELA0	
0	0	Sets instruction fetch as break condition.
0	1	Sets data read cycle as break condition.
1	0	Sets data write cycle as break condition.
1	1	Sets data read/write cycle as break condition.

Break interrupt enable

0	Disables PC break interrupt.
1	Enables PC break interrupt.

Notes: The bit configuration of BCRB is the same as that of BCRA.
* Only 0 can be written to these bits (to clear these flags).

ISCRH

Bit	15	14	13	12	11	10	9	8
	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRL

Bit	7	6	5	4	3	2	1	0
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 sense control A, B to IRQ0 sense control A,

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	
0	0	Interrupt request issued when $\overline{IRQ7}$ to $\overline{IRQ0}$ input level low.
	1	Interrupt request issued on falling edge of $\overline{IRQ7}$ to $\overline{IRQ0}$ input.
1	0	Interrupt request issued on rising edge of $\overline{IRQ7}$ to $\overline{IRQ0}$ input.
	1	Interrupt request issued on both falling and rising edge of $\overline{IRQ7}$ to $\overline{IRQ0}$ input.

IER—IRQ Enable Register **H'FE14** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 to IRA0 enable

0	Disables IRQn interrupt.
1	Enables IRQn interrupt.

(n= 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

IRQ7 to IRQ0 flag

0	[Clearing] (1) Writing 0 to flag IRQnF after reading IRQnF=1; (2) When interrupt exception processing is executed when set for LOW-level detection (IRQnSCB=IRQnSCA=0) and, in addition, the $\overline{\text{IRQn}}$ input level is HIGH; (3) When IRQn interrupt exception processing is executed when set for rising edge or falling edge or both rising edge and falling edge detection (IRQnSCB=1 and IRQnSCA=1); (4) When the DTC starts due to IRQn interrupt and the DTC MRB DISEL bit is 0.
1	[Setting] (1) When the $\overline{\text{IRQn}}$ input level changes to LOW when set for LOW level detection (IRQnSCB=IRQnSCA=0); (2) When a falling edge occurs at the $\overline{\text{IRQn}}$ input when set for falling edge detection (IRQnSCB=0, IRQnSCA=1); (3) When a rising edge occurs at the $\overline{\text{IRQn}}$ input when set for rising edge detection (IRQnSCB=1, IRQnSCA=0); (4) When either a falling edge or rising edge occurs at the $\overline{\text{IRQn}}$ input when set for both falling edge and rising edge detection (IRQnSCB=IRQnSCA=1).

(n= 7 to 0)

Note: * Only 0 can be written to these bits (to clear these flags).

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTC start enable

DTCEn	
0	DTC startup by interrupt disabled [Clearing] • When data transmission ends with the DISEL bit =1. • On completion of the specified number of transmissions.
1	DTC startup by interrupt enabled [Retention] When DISEL=0 and the specified number of transmissions has not completed.

(n= 7 to 0)

Bit	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/(W)*1	R/(W)*2	R/(W)*2	R/(W)*2	R/(W)*2	R/(W)*2	R/(W)*2	R/(W)*2

DTC software startup enable

0	DTC software startup disabled [Clearing] • When DISEL=0 and the specified number of transmissions has not completed. • When 0 is written after a software startup data transmit end interrupt (SWDTEND) request is sent to the CPU.
1	DTC software startup enabled [Retention] • When DISEL=1 and data transmission ends; • On completion of the specified number of transmissions; • During data transmission by software startup.

DTC software startup vector 6 to 0

Notes: *1 Only 1 can be written to the SWDTE bit.

*2 DTVEC6 to DTVEC0 can be written to when SWDTE=0.

Bit	:	7	6	5	4	3	2	1	0
		G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	:	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

Group 2 compare match select 1, 0

G2CMS1	G2CMS0	Pulse output group 2 output trigger
0	0	TPU channel 0 compare match
	1	TPU channel 1 compare match
1	0	TPU channel 2 compare match
	1	TPU channel 3 compare match

Group 3 compare match select 1, 0

G3CMS1	G3CMS0	Pulse output group 3 output trigger
0	0	TPU channel 0 compare match
	1	TPU channel 1 compare match
1	0	TPU channel 2 compare match
	1	TPU channel 3 compare match

Group 1 compare match select 1, 0

G1CMS1	G1CMS0	Pulse output group 1 output trigger
0	0	TPU channel 0 compare match
	1	TPU channel 1 compare match
1	0	TPU channel 2 compare match
	1	TPU channel 3 compare match

Group 0 compare match select 1, 0

G0CMS1	G0CMS0	Pulse output group 0 output trigger
0	0	TPU channel 0 compare match
	1	TPU channel 1 compare match
1	0	TPU channel 2 compare match
	1	TPU channel 3 compare match

Bit	7	6	5	4	3	2	1	0
	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Group 0 inversion

0	Pulse output group 0 set for inverted output (pin output level is set LOW when PODRL=1).
1	Pulse output group 0 set for direct output (pin output level is set HIGH when PODRL=1).

Group 1 inversion

0	Pulse output group 1 set for inverted output (pin output level is set LOW when PODRL=1).
1	Pulse output group 1 set for direct output (pin output level is set HIGH when PODRL=1).

Group 2 inversion

0	Pulse output group 2 set for inverted output (pin output level is set LOW when PODRH=1).
1	Pulse output group 2 set for direct output (pin output level is set HIGH when PODRH=1).

Group 3 inversion

0	Pulse output group 3 set for inverted output (pin output level is set LOW when PODRH=1).
1	Pulse output group 3 set for direct output (pin output level is set HIGH when PODRH=1).

Group 3 non-overlap

0	Pulse output group 3 set for normal operation (output value updated on compare match A for selected TPU).
1	Pulse output group 3 set for non-overlap operation (1 output and 0 output can be output independently on compare matches A and B of selected TPU).

Group 2 non-overlap

0	Pulse output group 2 set for normal operation (output value updated on compare match A for selected TPU).
1	Pulse output group 2 set for non-overlap operation (1 output and 0 output can be output independently on compare matches A and B of selected TPU).

Group 1 non-overlap

0	Pulse output group 1 set for normal operation (output value updated on compare match A for selected TPU).
1	Pulse output group 1 set for non-overlap operation (1 output and 0 output can be output independently on compare matches A and B of selected TPU).

Group 0 non-overlap

0	Pulse output group 0 set for normal operation (output value updated on compare match A for selected TPU).
1	Pulse output group 0 set for non-overlap operation (1 output and 0 output can be output independently on compare matches A and B of selected TPU).

NDERH

Bit	:	7	6	5	4	3	2	1	0
		NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 15 to 8

NDER15 to NDER8	
0	Pulse output PO15 to PO8 disabled (transfer from NDR15-NDR8 to POD15-POD8 disabled).
1	Pulse output PO15 to PO8 enabled (transfer from NDR15-NDR8 to POD15-POD8 enabled).

NDERL

Bit	:	7	6	5	4	3	2	1	0
		NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 7 to 0

NDER7 to NDER0	
0	Pulse output PO7 to PO0 disabled (transfer from NDR7-NDR0 to POD7-POD0 disabled).
1	Pulse output PO7 to PO0 enabled (transfer from NDR7-NDR0 to POD7-POD0 enabled).

PODRH

Bit	:	7	6	5	4	3	2	1	0
		POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

PODRL

Bit	:	7	6	5	4	3	2	1	0
		POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * The bits set for pulse output by NDER are read-only bits.

Same trigger for pulse output groups.

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—	—

Different triggers for pulse output groups.

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

Note: For details see section 12.2.4, Notes on NDR Access.

Same trigger for pulse output groups.

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—

Different triggers for pulse output groups.

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	—	—	—	—

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Note: For details see section 12.2.4, Notes on NDR Access.

P1DDR—Port 1 Data Direction Register **H'FE30** **Port**

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

P3DDR—Port 3 Data Direction Register **H'FE32** **Port**

Bit	:	7	6	5	4	3	2	1	0
		P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P7DDR—Port 7 Data Direction Register **H'FE36** **Port**

Bit	:	7	6	5	4	3	2	1	0
		P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PADDR—Port A Data Direction Register **H'FE39** **Port**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	W	W	W	W

PBDDR—Port B Data Direction Register **H'FE3A** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PCDDR—Port C Data Direction Register **H'FE3B** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR—Port D Data Direction Register **H'FE3C** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PEDDR—Port E Data Direction Register **H'FE3D** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PFDDR—Port F Data Direction Register **H'FE3E** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Mode 4 to 6									
Initial value	:	1	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W
Mode 7									
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PGDDR—Port G Data Direction Register H'FE3F Port

Bit	7	6	5	4	3	2	1	0
	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Mode 4 and 5								
Initial value	Undefined	Undefined	Undefined	1	0	0	0	0
R/W	—	—	—	W	W	W	W	W
Mode 6 and 7								
Initial value	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	—	—	—	W	W	W	W	W

PAPCR—Port A Pull-Up MOS Control Register H'FE40 Port

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

PBPCR—Port B Pull-Up MOS Control Register H'FE41 Port

Bit	7	6	5	4	3	2	1	0
	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCPCR—Port C Pull-Up MOS Control Register H'FE42 Port

Bit	7	6	5	4	3	2	1	0
	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDPCR—Port D Pull-Up MOS Control Register H'FE43 Port

Bit	7	6	5	4	3	2	1	0
	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEPCR—Port E Pull-Up MOS Control Register H'FE44 Port

Bit	7	6	5	4	3	2	1	0
	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3ODR—Port 3 Open-Drain Control Register H'FE46 Port

Bit	7	6	5	4	3	2	1	0
	P37ODR	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PAODR—Port A Open Drain Control Register H'FE47 Port

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value :	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W :	—	—	—	—	R/W	R/W	R/W	R/W

PBODR—Port B Open Drain Control Register H'FE48 Port

Bit	:	7	6	5	4	3	2	1	0
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCODR—Port C Open Drain Control Register H'FE49 Port

Bit	:	7	6	5	4	3	2	1	0
		PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TCR0
Channel 3: TCR3

Bit :	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler 2, 1, 0

TCR0

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	External clock: counts on TCLKD pin input

TCR3

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	Internal clock: counts on ø/1024
	1	0	Internal clock: counts on ø/256
		1	Internal clock: counts on ø/4096

Clock edge 1, 0

CKEG1	CKEG0	
0	0	Counts on rising edge.
	1	Counts on falling edge.
1	—	Counts on both edges.

Note: * Internal clock edge selection is valid only when the input clock is ø/4 or slower. This setting is ignored when the input clock is ø/1 or an overflow or underflow in another channel is selected.

Counter clear 2, 1, 0

CCLR2	CCLR1	CCLR0	
0	0	0	TCNT clearing disabled.
		1	TCNT cleared at TGRA compare match/input capture.
	1	0	TCNT cleared at TGRB compare match/input capture.
		1	TCNT cleared when other channel counters with synchronized clearing or synchronized operation are cleared.*1
1	0	0	TCNT clearing disabled.
		1	TCNT cleared at TGRC compare match/input capture.*2
	1	0	TCNT cleared at TGRD compare match/input capture.*2
		1	TCNT cleared when other channel counters with synchronized clearing or synchronized operation are cleared. *1

Note: *1 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
*2 When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Channel 0: TMDR0

Channel 3: TMDR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Buffer operation B

0	Normal TGRB operation.
1	Buffer operation of TGRB and TGRD.

Buffer operation A

0	Normal TGRA operation.
1	Buffer operation of TGRA and TGRC.

Mode 3 to 0

MD3*1	MD2*2	MD1	MD0	
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase calculation mode 1
			1	Phase calculation mode 2
		1	0	Phase calculation mode 3
			1	Phase calculation mode 4
1	*	*	*	—

* : Don't care

Notes: *1 MD3 is a reserved bit. Only write 0 to this bit.
*2 Phase calculation mode cannot be set for channels 0 and 3. Only write 0 to MD2.

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3A I/O Control

0	0	0	0	TGR3A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

*: Don't care

TGR3B I/O Control

0	0	0	0	TGR3B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR4A I/O Control

0	0	0	0	TGR4A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

*: Don't care

TGR4B I/O Control

0	0	0	0	TGR4B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/input capture

*: Don't care

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR5A I/O Control

0	0	0	0	TGR5A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGR5B I/O Control

0	0	0	0	TGR5B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0A I/O Control

0	0	0	0	TGR0A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down

*: Don't care

TGR0B I/O Control

0	0	0	0	TGR0B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR1A I/O Control

0	0	0	0	TGR1A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/ input capture

*: Don't care

TGR1B I/O Control

0	0	0	0	TGR1B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don't care

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR2A I/O Control

0	0	0	0	TGR2A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGR2B I/O Control

0	0	0	0	TGR2B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3C I/O Control

0	0	0	0	TGR3C is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

*: Don't care

Note: *1 When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

TGR3D I/O Control

0	0	0	0	TGR3D is output compare register*2	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0C I/O Control

0	0	0	0	TGR0C is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

TGR0D I/O Control

0	0	0	0	TGR0D is output compare register*2	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Channel 0: TIER0

Channel 3: TIER3

Bit	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Overflow interrupt enable

0	TCFV interrupt request (TCIV) disabled.
1	TCFV interrupt request (TCIV) enabled.

A/D conversion start request enable

0	A/D conversion start request generation disabled.
1	A/D conversion start request generation enabled.

TGR interrupt enable D

0	TGFD bit interrupt request (TGID) disabled.
1	TGFD bit interrupt request (TGID) enabled.

TGR interrupt enable C

0	TGFC bit interrupt request (TGIC) disabled.
1	TGFC bit interrupt request (TGIC) enabled.

TGR interrupt enable B

0	TGFB bit interrupt request (TGIB) disabled
1	TGFB bit interrupt request (TGIB) enabled

TGR interrupt enable A

0	TGFA bit interrupt request (TGIA) disabled.
1	TGFA bit interrupt request (TGIA) enabled.

Channel 0: TSR0

Channel 3: TSR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Overflow flag

0	[Clearing] Writing 0 to TCFV after reading TCFV=1.
1	[Setting] When the TCNT value overflows (H'FFFF → H'0000).

Input capture/output compare flag D

0	[Clearing] (1) When the DTC is started by a TGID interrupt and the DTC MRB DISEL bit is 0; (2) Writing 0 to TGFD after reading TGFD=1.
1	[Setting] (1) When TGRD is functioning as the output compare register and TCNT=TGRD; (2) When TGRD is functioning as the input capture register and the value of TCNT is sent to TGRD by the input capture signal.

Input capture/output compare flag C

0	[Clearing] (1) When the DTC is started by a TGIC interrupt and the DTC MRB DISEL bit is 0; (2) Writing 0 to TGFC after reading TGFC=1.
1	[Setting] (1) When TGRC is functioning as the output compare register and TCNT=TGRC; (2) When TGRC is functioning as the input capture register and the value of TCNT is sent to TGRC by the input capture signal.

Input capture/output compare flag B

0	[Clearing] (1) When the DTC is started by a TGIB interrupt and the DTC MRB DISEL bit is 0; (2) Writing 0 to TGFB after reading TGFB=1.
1	[Setting] (1) When TGRB is functioning as the output compare register and TCNT=TGRB; (2) When TGRB is functioning as the input capture register and the value of TCNT is sent to TGRB by the input capture signal.

Input capture/output compare flag A

0	[Clearing] (1) When the DTC is started by a TGIA interrupt and the DTC MRB DISEL bit is 0; (2) When the DMAC is started by a TGIA interrupt and the D MAC DMABCR DTA bit is 1; (3) Writing 0 to TGFA after reading TGFA=1.
1	[Setting] (1) When TGRA is functioning as the output compare register and TCNT=TGRA; (2) When TGRA is functioning as the input capture register and the value of TCNT is sent to TGRA by the input capture signal.

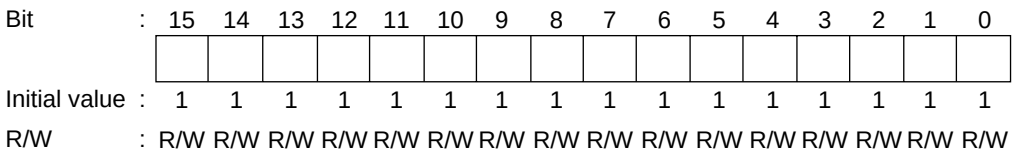
Note: * Only 0 can be written to these bits (to clear these flags).

TCNT0—Timer Counter 0	**H'FF16**	**TPU0**
TCNT0—Timer Counter 1	**H'FF26**	**TPU1**
TCNT0—Timer Counter 2	**H'FF36**	**TPU2**
TCNT0—Timer Counter 3	**H'FE86**	**TPU3**
TCNT0—Timer Counter 4	**H'FE96**	**TPU4**
TCNT0—Timer Counter 5	**H'FEA6**	**TPU5**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	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: * This register can be used as an up/down counter only in phase calculation mode (and when counting overflows and underflows in other channels in phase calculation mode) In all other cases, this register functions as an up-counter.

TGR0A—Timer General Register 0A	**H'FF18**	**TPU0**
TGR0B—Timer General Register 0B	**H'FF1A**	**TPU0**
TGR0C—Timer General Register 0C	**H'FF1C**	**TPU0**
TGR0D—Timer General Register 0D	**H'FF1E**	**TPU0**
TGR1A—Timer General Register 1A	**H'FF28**	**TPU1**
TGR1B—Timer General Register 1B	**H'FF2A**	**TPU1**
TGR2A—Timer General Register 2A	**H'FF38**	**TPU2**
TGR2B—Timer General Register 2B	**H'FF3A**	**TPU2**
TGR3A—Timer General Register 3A	**H'FE88**	**TPU3**
TGR3B—Timer General Register 3B	**H'FE8A**	**TPU3**
TGR3C—Timer General Register 3C	**H'FE8C**	**TPU3**
TGR3D—Timer General Register 3D	**H'FE8E**	**TPU3**
TGR4A—Timer General Register 4A	**H'FE98**	**TPU4**
TGR4B—Timer General Register 4B	**H'FE9A**	**TPU4**
TGR5A—Timer General Register 5A	**H'FEA8**	**TPU5**
TGR5B—Timer General Register 5B	**H'FEAA**	**TPU5**



Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	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

Channel 1: TCR1
Channel 2: TCR2
Channel 4: TCR4
Channel 5: TCR5

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler 2, 1, 0

TCR1

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	Internal clock: counts on ø/256
		1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

TCR2

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	Internal clock: counts on ø/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

TCR4

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKC pin input
	1	0	Internal clock: counts on ø/1024
		1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

TCR5

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKC pin input
	1	0	Internal clock: counts on ø/256
		1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

Clock edge 1, 0

CKEG1	CKEG0	
0	0	Counts on rising edge.
	1	Counts on falling edge.
1	—	Counts on both edges.

Note: Internal clock edge selection is valid only when the input clock is ø/4 or slower. This setting is ignored when the input clock is ø/1 or an overflow or underflow in another channel is selected.

Counter clear 2, 1, 0

Reserve*2	CCLR1	CCLR0	
0	0	0	TCNT clearing disabled.
		1	TCNT cleared at TGRA compare match/input capture.
	1	0	TCNT cleared at TGRB compare match/input capture.
		1	TCNT cleared when other channel counters with synchronized clearing or synchronized operation are cleared.*1

Note: *1 Sync operation is selected by setting 1 in the TSYR SYNC bit.
*2 Bit 7 of channels 1, 2, 4, and 5 is reserved. This bit always returns 0 when read, and cannot be written to.

TMDR1—Timer Mode Register 1	**H'FF21**	**TPU1**
TMDR2—Timer Mode Register 2	**H'FF31**	**TPU2**
TMDR4—Timer Mode Register 4	**H'FE91**	**TPU4**
TMDR5—Timer Mode Register 5	**H'FEA1**	**TPU5**

Channel 1: TMDR1

Channel 2: TMDR2

Channel 4: TMDR4

Channel 5: TMDR5

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode 3 to 0

MD3*1	MD2*2	MD1	MD0	
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase calculation mode 1
			1	Phase calculation mode 2
		1	0	Phase calculation mode 3
			1	Phase calculation mode 4
1	*	*	*	—

* : Don't care

Note: *1 MD3 is a reserved bit. Only write 0 to this bit.
*2 Phase calculation mode cannot be set for channels 0 and 3. Only write 0 to MD2.

TIER1—Timer Interrupt Enable Register 1	**H'FF24**	**TPU1**
TIER2—Timer Interrupt Enable Register 2	**H'FF34**	**TPU2**
TIER4—Timer Interrupt Enable Register 4	**H'FE94**	**TPU4**
TIER5—Timer Interrupt Enable Register 5	**H'FEA4**	**TPU5**

Channel 1: TIER1

Channel 2: TIER2

Channel 4: TIER4

Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W	R/W

Underflow interrupt enable

0	TCFU interrupt request (TCIU) disabled.
1	TCFU interrupt request (TCIU) enabled.

A/D conversion start request enable

0	A/D conversion start request generation disabled.
1	A/D conversion start request generation enabled.

Overflow interrupt enable

0	TCFV interrupt request (TCIV) disabled.
1	TCFV interrupt request (TCIV) enabled.

TGR interrupt enable B

0	TGFB bit interrupt request (TGIB) disabled.
1	TGFB bit interrupt request (TGIB) enabled.

TGR interrupt enable A

0	TGFA bit interrupt request (TGIA) disabled.
1	TGFA bit interrupt request (TGIA) enabled.

Channel 1: TSR1

Channel 2: TSR2

Channel 4: TSR4

Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Underflow flag

0	[Clearing] Writing 0 to TCFU after reading TCFU=1.
1	[Setting] When the TCNT value underflows (H'0000 → H'FFFF).

Count direction flag

0	TCNT counts down.
1	TCNT counts up.

Overflow flag

0	[Clearing] Writing 0 to TCFV after reading TCFV=1.
1	[Setting] When the TCNT value overflows (H'FFFF → H'0000).

Input capture/output compare flag B

0	[Clearing] (1) When the DTC is started by a TGIB interrupt and the DTC MRB DISEL bit is 0; (2) Writing 0 to TGFB after reading TGFB=1.
1	[Setting] (1) When TGRB is functioning as the output compare register and TCNT= TGRB; (2) When TGRB is functioning as the input capture register and the value of TCNT is sent to TGRB by the input capture signal.

Input capture/output compare flag A

0	[Clearing] (1) When the DTC is started by a TGIA interrupt and the DTC MRB DISEL bit is 0; (2) When the DMAC is started by a TGIA interrupt and the DMAC DMABCR DTA bit is 1; (3) Writing 0 to TGFA after reading TGFA=1.
1	[Setting] (1) When TGRA is functioning as the output compare register and TCNT= TGRA; (2) When TGRA is functioning as the input capture register and the value of TCNT is sent to TGRA by the input capture signal.

Note: * Only 0 can be written to these bits (to clear these flags).

TSTR—Timer Start Register **H'FEB0** **TPU Common**

Bit	7	6	5	4	3	2	1	0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Counter start 5 to 0

0	TCNTn counting operation disabled.
1	TCNTn counting operation enabled.

(n= 5 to 0)

Note: When the TIOC pin is operating as an output pin, writing 0 to a CST bit disables counting. The TIOC pins output compare output level is maintained. When a CST bit is 0, the output level of the pin is updated to the set initial output value by writing to TIOR.

TSYR—Timer Synchro Register **H'FEB1** **TPU Common**

Bit	7	6	5	4	3	2	1	0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Timer sync 5 to 0

0	TCNTn operate independently (TCNTs are preset and cleared independently of other channels)
1	TCNTn operate in sync mode. Synchronized TCNT presetting and clearing enabled.

(n= 5 to 0)

Note: *1 The SYNC bit of a minimum of two channels must be set to 1 in order to select sync operation.
*2 To enable sync clearing, in addition to the SYNC bits, the TCR CCLR2 to CCLR0 bits must be set for the TCNT clearing factors.

IPRA—Interrupt Priority Register A	**H'FEC0**
IPRB—Interrupt Priority Register B	**H'FEC1**
IPRC—Interrupt Priority Register C	**H'FEC2**
IPRD—Interrupt Priority Register D	**H'FEC3**
IPRE—Interrupt Priority Register E	**H'FEC4**
IPRF—Interrupt Priority Register F	**H'FEC5**
IPRG—Interrupt Priority Register G	**H'FEC6**
IPRH—Interrupt Priority Register H	**H'FEC7**
IPRI—Interrupt Priority Register I	**H'FEC8**
IPRJ—Interrupt Priority Register J	**H'FEC9**
IPRK—Interrupt Priority Register K	**H'FECA**
IPRL—Interrupt Priority Register L	**H'FECB**
IPRO—Interrupt Priority Register O	**H'FECE**

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Interrupt factors vs IPR

Register	Bit	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2	IRQ4
	IRQ3	IRQ5
IPRC	IRQ6 IRQ7	DTC
IPRD	Watchdog timer 0	Refresh timer
IPRE	PC brake	ADC Watchdog timer 1
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	ITPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRL	8-bit timer 2, 3	IIC (optional)
IPRO	SCI channel 3	SCI channel 4

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Mode 5 to 7	:								
Initial value	:	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
Mode 4									
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 bus width control

0	Sets area n to 16-bit access.
1	Sets area n to 8-bit access.

(n= 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	:	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

Area 7 to 0 access state control

0	Area n set as 2-state access area. Insertion of wait states in area n external area access is disabled.
1	External area access of area n set as 3-state access area. Insertion of wait states in area n external area access is enabled.

(n= 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	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

Area 6 wait control 1, 0

W61	W60	
0	0	No program wait inserted when accessing external area of area 6.
	1	1 program wait state inserted when accessing external area of area 6.
1	0	2 program wait states inserted when accessing external area of area 6.
	1	3 program wait states inserted when accessing external area of area 6.

Area 7 wait control 1, 0

W71	W70	
0	0	No program wait inserted when accessing external area of area 7.
	1	1 program wait state inserted when accessing external area of area 7.
1	0	2 program wait states inserted when accessing external area of area 7.
	1	3 program wait states inserted when accessing external area of area 7.

Area 5 wait control 1, 0

W51	W50	
0	0	No program wait inserted when accessing external area of area 5.
	1	1 program wait state inserted when accessing external area of area 5.
1	0	2 program wait states inserted when accessing external area of area 5.
	1	3 program wait states inserted when accessing external area of area 5.

Area 4 wait control 1, 0

W41	W40	
0	0	No program wait inserted when accessing external area of area 4.
	1	1 program wait state inserted when accessing external area of area 4.
1	0	2 program wait states inserted when accessing external area of area 4.
	1	3 program wait states inserted when accessing external area of area 4.

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	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

Area 2 wait control

W21	W20	
0	0	No program wait inserted when accessing external area of area 2.
	1	1 program wait state inserted when accessing external area of area 2.
1	0	2 program wait states inserted when accessing external area of area 2.
	1	3 program wait states inserted when accessing external area of area 2.

Area 3 wait control

W31	W30	
0	0	No program wait inserted when accessing external area of area 3.
	1	1 program wait state inserted when accessing external area of area 3.
1	0	2 program wait states inserted when accessing external area of area 3.
	1	3 program wait states inserted when accessing external area of area 3.

Area 1 wait control

W11	W10	
0	0	No program wait inserted when accessing external area of area 1.
	1	1 program wait state inserted when accessing external area of area 1.
1	0	2 program wait states inserted when accessing external area of area 1.
	1	3 program wait states inserted when accessing external area of area 1.

Area 0 wait control

W01	W00	
0	0	No program wait inserted when accessing external area of area 0.
	1	1 program wait state inserted when accessing external area of area 0.
1	0	2 program wait states inserted when accessing external area of area 0.
	1	3 program wait states inserted when accessing external area of area 0.

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMTS0
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Burst cycle select 1

0	Burst cycle = 1 state.
1	Burst cycle = 2 states.

Burst ROM enable

0	Area 0 is basic bus interface. (Initial value)
1	Area 0 is burst ROM interface.

Idle cycle insertion 0

0	No idle cycle is inserted when an external read cycle follows an external write cycle.
1	An idle cycle is inserted when an external read cycle follows an external write cycle. (Initial value)

Idle cycle insertion 1

0	No idle cycle is inserted when an external read cycle follows an external read cycle of another area.
1	An idle cycle is inserted when an external read cycle follows an external read cycle of another area. (Initial value)

Burst cycle select 0

0	Burst access = 4 words max.
1	Burst access = 8 words max.

RAM type select

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	0	Normal area			
		1	Normal area			DRAM area
	1	0	Normal area		DRAM area	
		1	DRAM area			
1	1	1	Contiguous DRAM area			

Note: When all areas selected in the DRAM area are set for 8-bit access, the PF2 pin can be used as an I/O port or $\overline{\text{BREQO}}$ or $\overline{\text{WAIT}}$. When set for contiguous DRAM the bus widths for areas 2 to 5 and the number of access states (number of programmable waits) must be set to the same values. Do not attempt to set combinations other than those shown in the table.

Bit	7	6	5	4	3	2	1	0
	BRLE	BREQOE	—	OES	DDS	RCTS	WDBE	WAITE
Initial value	0	0	0	0	1	0	0	0
R/W	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W

OE select

0	$\overline{CS3}$ pin used as port or as $\overline{CS3}$ signal output.
1	When only area 2 is set as DRAM, or when areas 2 to 5 are set as contiguous DRAM space, the $\overline{CS3}$ pin is used as the $\overline{OE}$ pin.

BREQO pin enable

0	$\overline{BREQO}$ output disabled. $\overline{BREQO}$ can be used as an I/O port.
1	$\overline{BREQO}$ output enabled.

Bus release enable

0	Release of external bus privileges disabled. $\overline{BREQ}$, $\overline{BACK}$, and $\overline{BREQO}$ can be used as I/O ports.
1	Release of external bus privileges enabled.

DACK timing select

0	When performing DMAC single address transmission to the DRAM space, always perform full access. The $\overline{DACK}$ signal level changes to LOW from T_r or T_1 cycle.
1	Burst access is also available when performing DMAC single address transmission to the DRAM space. The $\overline{DACK}$ signal level changes to LOW from T_{C1} or T_2 cycle.

Read CAS timing select

0	$\overline{CAS}$ signal output timing is the same when reading and writing.
1	When reading, the $\overline{CAS}$ signal is asserted one half cycle faster than when writing.

Write data buffer enable

0	Do not use write data buffer function.
1	Use write data buffer function.

WAIT pin enable

0	Wait input via $\overline{WAIT}$ pin disabled. The $\overline{WAIT}$ pin can be used as an I/O port.
1	Wait input via $\overline{WAIT}$ pin enabled.

Bit	7	6	5	4	3	2	1	0
	TPC	BE	RCDM	CW2	MXC1	MXC0	RLW1	RLW0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bit

RAS down mode

0	DRAM interface: RAS up mode selected.
1	DRAM interface: RAS down mode selected.

Burst access enable

0	Burst access disabled (permanently full access).
1	DRAM space accessed in high-speed page mode.

TP cycle control

0	One precharge cycle state inserted.
1	Two precharge cycle states inserted.

Multiplex shift count 1, 0

MXC1	MXC0	
0	0	8-bit shift (1) When set for 8-bit access space: Row addresses A23 to A8 are targets of comparison. (2) When set for 16-bit access space: Row addresses A23 to A9 are targets of comparison.
	1	9-bit shift (1) When set for 8-bit access space: Row addresses A23 to A9 are targets of comparison. (2) When set for 16-bit access space: Row addresses A23 to A10 are targets of comparison.
1	0	10-bit shift (1) When set for 8-bit access space: Row addresses A23 to A10 are targets of comparison. (2) When set for 16-bit access space: Row addresses A23 to A11 are targets of comparison.
	1	—

Refresh cycle wait control 1, 0

RLW1	RLW0	
0	0	Do not insert wait state.
	1	Insert 1 wait state.
1	0	Insert 2 wait states.
	1	Insert 3 wait states.

Bit	:	7	6	5	4	3	2	1	0
		RFSHE	CBRM	RMODE	CMF	CMIE	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Refresh mode

0	Do not perform self-refresh in software standby mode.
1	Perform self-refresh in software standby mode.

CBR refresh mode

0	External access enabled at CAS-before-RAS refresh.
1	External access disabled at CAS-before-RAS refresh.

Refresh control

0	Do not perform refresh control.
1	Perform refresh control.

Compare match flag

0	[Clearing] Writing 0 to CMF flag after reading CMF=1.
1	[Setting] When RTCNT=RTCOR.

Compare match interrupt enable

0	CMF flag interrupt request (CMI) disabled.
1	CMF flag interrupt request (CMI) enabled.

Refresh counter clock select

CKS2	CKS1	CKS0	
0	0	0	No counting operation
		1	Counting on ø/2
	1	0	Counting on ø/8
		1	Counting on ø/32
1	0	0	Counting on ø/128
		1	Counting on ø/512
	1	0	Counting on ø/2048
		1	Counting on ø/4096

RTCNT—Refresh Timer Counter **H'FED8** **Bus Controller**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCOR—Refresh Time Constant Register **H'FED9** **Bus Controller**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

RAMER—RAM Emulation Register **H'FEDB** **FLASH**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Flash memory area selection

Addresses	Block Name	RAMS	RAM1	RAM1	RAM0
H'FFD000–H'FFDFFF	RAM area 4 kbytes	0	*	*	*
H'000000–H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000–H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000–H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000–H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000–H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000–H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000–H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000–H'007FFF	EB7 (4 kbytes)	1	1	1	1

* : Don't care

RAM Select

0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

MAR0AH—Memory Address Register 0AH **H'FEE0** **DMAC**
MAR0AL—Memory Address Register 0AL **H'FEE2** **DMAC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR	—	—	—	—	—	—	—	—								
Initial value	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Not used

* : Undefined

IOAR0A—I/O Address Register 0A **H'FEE4** **DMAC**
IOAR1A—I/O Address Register 1A **H'FEF4** **DMAC**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Not used

* : Undefined

ETCR0A—Transfer Count Register 0A **H'FEE6** **DMAC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR0A	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sequential mode and idle mode Normal mode: Transfer counter

Repeat mode: Holds number of transfers / Transfer counter

Block transfer mode: Holds block size / Block size counter

*: Undefined

MAR0BH—Memory Address Register 0BH **H'FEE8** **DMAC**
MAR0BL—Memory Address Register 0BL **H'FEEA** **DMAC**

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR0BH	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR0BL	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Specifies transfer destination

*: Undefined

IOAR0B—I/O Address Register 0B **H'FEEC** **DMAC**
IOAR1B—I/O Address Register 1B **H'FEFC** **DMAC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR0B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Not used

*: Undefined

ETCR0B—Transfer Count Register 0B **H'FEEE** **DMAC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR0B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sequential mode and idle mode: Transfer counter

Repeat mode: Holds number of transfers / Transfer counter

Block transfer mode: Block transfer counter

*: Undefined

Note: Not used in normal mode.

MAR1AH—Memory Address Register 1AH **H'FEF0** **DMAC**
MAR1AL—Memory Address Register 1AL **H'FEF2** **DMAC**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR1AH	—	—	—	—	—	—	—	—								
Initial value	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR1AL																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Not used

*: Undefined

ETCR1A—Transfer Count Register 1A **H'FEF6** **DMAC**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR1A																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sequential mode, Idle mode, Normal mode: Transfer counter

Repeat mode: Holds number of transfers / Transfer counter

Block transfer mode: Holds block size / Block size counter

*: Undefined

MAR1BH—Memory Address Register 1BH **H'FEF8** **DMAC**
MAR1BL—Memory Address Register 1BL **H'FEFA** **DMAC**

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR1BH	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
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
MAR1BL	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In short address mode: Specifies transfer destination/transfer source address
In full address mode: Not used

*: Undefined

ETCR1B—Transfer Count Register 1B **H'FEFE** **DMAC**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR1B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sequential mode and idle mode: Transfer counter

Repeat mode: Holds number of transfers | Transfer counter

Block transfer mode: Block transfer counter

*: Undefined

Note: Not used in normal mode.

P1DR—Port 1 Data Register **H'FF00** **Port**

Bit	7	6	5	4	3	2	1	0
	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR—Port 3 Data Register **H'FF02** **Port**

Bit	7	6	5	4	3	2	1	0
	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7DR—Port 7 Data Register **H'FF06** **Port**

Bit	7	6	5	4	3	2	1	0
	P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PADR—Port A Data Register **H'FF09** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

PBDR—Port B Data Register **H'FF0A** **Port**

Bit	7	6	5	4	3	2	1	0
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDR—Port C Data Register **H'FF0B** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDDR—Port D Data Register **H'FF0C** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEDR—Port E Data Register **H'FF0D** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFDR—Port F Data Register **H'FF0E** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PGDR—Port G Data Register **H'FF0F** **Port**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Write enable 1B

0	Disables writing to all DMACR1B bits, DMABCR bits 11, 7, and 3, and DMATCR bit 5. (Initial value)
1	Enables writing to all DMACR1B bits, DMABCR bits 11, 7, and 3, and DMATCR bit 5.

Write enable 1A

0	Disables writing to all DMACR1A bits, and DMABCR bits 10, 6, and 2. (Initial value)
1	Enables writing to all DMACR1A bits, and DMABCR bits 10, 6, and 2.

Write enable 0B

0	Disables writing to all DMACR0B bits, DMABCR bits 9, 5, and 1, and DMATCR bit 4 (Initial value)
1	Enables writing to all DMACR0B bits, DMABCR bits 9, 5, and 1, and DMATCR bit 4.

Write enable 0A

0	Disables writing to all DMACR0A bits, and DMABCR bits 8, 4, and 0. (Initial value)
1	Enables writing to all DMACR0A bits, and DMABCR bits 8, 4, and 0.

DMATCR—DMA Terminal Control Register H'FF61 DMAC

Bit	7	6	5	4	3	2	1	0
DMATCR	—	—	TEE1	TEE0	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	R/W	R/W	—	—	—	—

Transfer end pin enable 0

0	Disables $\overline{\text{TEND0}}$ pin output.
1	Enables $\overline{\text{TEND0}}$ pin output.

Transfer end pin enable 1

0	Disables $\overline{\text{TEND1}}$ pin output.
1	Enables $\overline{\text{TEND1}}$ pin output.

Full address mode

Bit	:	15	14	13	12	11	10	9	8
DMACRA	:	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block Direction/Block Enable

0	0	Transfer in normal mode
	1	Transfer in block transfer mode, destination is block area
1	0	Transfer in normal mode
	1	Transfer in block transfer mode, source is block area

Source Address Increment/Decrement

0	0	MARA is fixed
	1	MARA is incremented after a data transfer • When DTSZ = 0, MARA is incremented by 1 after a transfer • When DTSZ = 1, MARA is incremented by 2 after a transfer
1	0	MARA is fixed
	1	MARA is decremented after a data transfer • When DTSZ = 0, MARA is decremented by 1 after a transfer • When DTSZ = 1, MARA is decremented by 2 after a transfer

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

Full address mode

Bit	:	7	6	5	4	3	2	1	0
DMACRB	:	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Transfer Factor

DTF3	DTF2	DTF1	DTF0	Block Transfer Mode	Normal Mode
0	0	0	0	—	—
			1	Activated by A/D converter conversion end interrupt	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*	Activated by $\overline{\text{DREQ}}$ pin falling edge input
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt	—
			1	Activated by SCI channel 0 reception complete interrupt	—
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt	Auto-request (cycle steal)
			1	Activated by SCI channel 1 reception complete interrupt	Auto-request (burst)
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt	—
			1	Activated by TPU channel 1 compare match/input capture A interrupt	—
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt	—
			1	Activated by TPU channel 3 compare match/input capture A interrupt	—
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt	—
			1	Activated by TPU channel 5 compare match/input capture A interrupt	—
		1	0	—	—
			1	—	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

Destination Address Increment/Decrement

0	0	MARB is fixed
	1	MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 after a transfer • When DTSZ = 1, MARB is incremented by 2 after a transfer
1	0	MARB is fixed
	1	MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 after a transfer • When DTSZ = 1, MARB is decremented by 2 after a transfer

Short address mode

Bit	7	6	5	4	3	2	1	0
DMACR	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

Data Transfer Increment/Decrement

0	MAR is incremented after a data transfer • When DTSZ = 0, MAR is incremented by 1 after a transfer • When DTSZ = 1, MAR is incremented by 2 after a transfer
1	MAR is decremented after a data transfer • When DTSZ = 0, MAR is decremented by 1 after a transfer • When DTSZ = 1, MAR is decremented by 2 after a transfer

Data Transfer Factor

				Channel A	Channel B
0	0	0	0	—	
			1	Activated by A/D converter conversion end interrupt	
		1	0	—	Activated by $\overline{\text{DREQ}}$ pin falling edge input
			1	—	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt	
			1	Activated by SCI channel 0 reception complete interrupt	
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt	
			1	Activated by SCI channel 1 reception complete interrupt	
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt	
			1	Activated by TPU channel 1 compare match/input capture A interrupt	
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt	
			1	Activated by TPU channel 3 compare match/input capture A interrupt	
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt	
			1	Activated by TPU channel 5 compare match/input capture A interrupt	
		1	0	—	
			1	—	

Data Transfer Direction

0	0	Transfer with MAR as source address and IOAR as destination address
	1	Transfer with IOAR as source address and MAR as destination address
1	0	Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe
	1	Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address

Repeat Enable

0	0	Transfer in sequential mode (no transfer end interrupt)
	1	Transfer in sequential mode (with transfer end interrupt)
1	0	Transfer in repeat mode (no transfer end interrupt)
	1	Transfer in idle mode (with transfer end interrupt)

Short address mode

Bit	15	14	13	12	11	10	9	8
DMABCRH	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Single address enable 0

0	Transfer in dual address mode.
1	Transfer in single address mode.

Data transfer acknowledge 1B

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Single address enable 1

0	Transfer in dual address mode.
1	Transfer in single address mode.

Data transfer acknowledge 1A

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Full address enable 0

0	Short address mode.
1	Full address mode.

Data transfer acknowledge 0B

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Full address enable 1

0	Short address mode.
1	Full address mode.

Data transfer acknowledge 0A

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Bit	7	6	5	4	3	2	1	0
DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data transfer enable 0A

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer interrupt enable 1B

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer enable 0B

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer interrupt enable 1A

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer enable 1A

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer interrupt enable 0B

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer enable 1B

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer interrupt enable 0A

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Full address mode

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	—	—	DTA1	—	DTA0	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data transfer acknowledge 0

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Data transfer acknowledge 1

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Full address enable 0

0	Short address mode
1	Full address mode

Full address enable 1

0	Short address mode
1	Full address mode

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data transfer end interrupt enable 0A

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Data transfer end interrupt enable 1A

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Data transfer interrupt enable 0B

0	Transfer break interrupt disabled
1	Transfer break interrupt enabled

Data transfer interrupt enable 1B

0	Transfer break interrupt disabled
1	Transfer break interrupt enabled

Data transfer enable 0

0	Data transfer disabled
1	Data transfer enabled

Data transfer master enable 0

0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

Data transfer enable 1

0	Data transfer disabled
1	Data transfer enabled

Data transfer master enable 1

0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
R/W	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock select 2 to 0

WDT0 input clock select

CKS2	CKS1	CKS0	Clock	Overflow cycle (when ø= 25MHz)
0	0	0	ø/2	20.4 µs
		1	ø/64	652.8 µs
	1	0	ø/128	1.3 ms
		1	ø/512	5.2 ms
1	0	0	ø/2048	20.9 ms
		1	ø/8192	83.6 ms
	1	0	ø/32768	334.2 ms
		1	ø/131072	1.34 s

Note: The overflow cycle starts when TCNT starts counting from H'00 and ends when an overflow occurs.

Timer enable

0	Initializes TCNT to H'00 and disables the counting operation.
1	TCNT performs counting operation.

Timer mode select

0	Interval timer mode: Interval timer interrupt (WOVI) request sent to CPU when overflow occurs at TCNT.
1	Watchdog timer mode: $\overline{WDTOVF}$ signal output externally when overflow occurs at TCNT. *

Note: * See Section 15.2.3, "Reset control/status register (RSTCSR)" for details of when TCNT overflows in watchdog timer mode.

Overflow flag

0	[Clearing] When 0 is written to OVF bit after reading TCSR when OVF=1.
1	[Setting] When TCNT overflows (changes from H'FF to H'00). When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.

Notes: * Only 0 can be written to these bits (to clear these flags).

TCSR is write-protected by a password to prevent accidental overwriting. For details see section 15.2.5, Notes on Register Access.

TCNT0—Timer Counter 0 **H'FF74 (W), H'FF75 (R)** **WDT0**
TCNT1—Timer Counter 1 **H'FFA2 (W), H'FFA3 (R)** **WDT1**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TCNT is write-protected by a password to prevent accidental overwriting.
For details see section 15.2.5, Notes on Register Access.

RSTCSR—Reset Control/Status Register **H'FF76 (W), H'FF77 (R)** **WDT0**

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

Reset select

0	Power-on reset.
1	Manual reset.

Reset enable

0	No internal reset on TCNT overflow.*
1	Internal reset performed on TCNT overflow.

Note: * The LSI is not internally reset, but TCNT and TCSR in WDT are reset.

Watchdog timer overflow flag

0	[Clearing] Writing 0 to WOVF after reading TCSR when WOVF=1.
1	[Setting] When, in watchdog timer mode, TCNT overflows (H'FF→ H'00).

Notes: * Only 0 can be written to these bits (to clear these flags).
TCNT is write-protected by a password to prevent accidental overwriting.
For details see section 15.2.5, Notes on Register Access.

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Start condition/stop condition prohibit

0	Writing 0 issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1 Writing is ignored

I²C Bus interface interrupt request flag

0	Waiting for transfer, or transfer in progress
1	Interrupt requested

Note: * For details see section 18.2.5, I²C Bus Control Register.

Bus busy

0	Bus is free [Clearing condition] When a stop condition is detected
1	Bus is free [Clearing condition] When a stop condition is detected

Acknowledge bit judgement selection

0	The value of the acknowledge bit is ignored, and continuous transfer is performed
1	If the acknowledge bit is 1, continuous transfer is interrupted

Master/slave select, transmit/receive select

0	0	Slave receive mode
	1	Slave transmit mode
1	0	Master receive mode
	1	Master transmit mode

Note: * For details see section 18.2.5, I²C Bus Control Register.

I²C Bus Interface Interrupt Enable

0	Interrupts disabled
1	Interrupts enabled

I²C Bus Interface Enable

0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function I²C bus interface module internal states initialized SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus) ICMR and ICDR can be accessed

Note: * Only 0 can be written, for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
		ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Acknowledge bit

0	When receiving, 0 is output at acknowledge output timing. When transmitting, this bit shows that an acknowledge (0) has not been sent from the receiving device.
1	When receiving, 1 is output at acknowledge output timing. When transmitting, this bit shows that an acknowledge (1) has been sent from the receiving device.

General call address confirmation flag

0	General call address not confirmed [Clearing] (1) When data is written to ICDR (when sending), or when data is read from ICDR (when receiving); (2) When 0 is written after reading ADZ=1; (3) In master mode.
1	General call address confirmation [Setting] • When general call address is detected is in slave receive mode and FSX = 0 or FS = 0).

Slave address confirmation flag

0	Slave address or general call address not confirmed [Clearing] (1) When data is written to ICDR (when sending), or when data is read from ICDR (when receiving); (2) When 0 is written after reading AAS=1; (3) In master mode.
1	Slave address or general call address confirmed [Setting] • When slave address or general call address is detected in slave receive mode and FS = 0.

Arbitration lost flag

0	Secure bus. [Clearing] (1) When data is written to ICDR (when sending), or when data is read (when receiving); (2) When 0 is written after reading AL=1.
1	Bus arbitration lost [Setting] (1) When there is a mismatch between internal SDA and SDA pin at rise in SCL in master transmit mode; (2) When the internal SCL level is HIGH at the fall in SCL in master transmit mode.

2nd slave address confirmation flag

0	2nd slave address not confirmed [Clearing] (1) When 0 is written after reading AASX=1; (2) When start conditions are detected; (3) In master mode.
1	2nd slave address confirmed [Setting] • When 2nd slave address is detected in slave receive mode and FSX = 0.

I²C bus interface continuous transmit and receive interrupt request flag

0	Transmit wait state, or transmitting [Clearing] (1) When 0 written after reading IRTR=1; (2) When IRIC flag is cleared to 0.
1	Continuous transmit state [Setting] • In I²C bus interface slave mode When 1 is set in TDRE or RDRF flag when AASX=1. • In other than I²C bus interface slave mode When TDRE or RDRF flag is set to 1.

Normal end condition detection flag

0	No normal end condition [Clearing] (1) When 0 is written after reading STOP=1; (2) When IRIC flag is cleared to 0.
1	Normal end condition detected in slave mode in I²C bus format [Setting] On detection of stop condition on completion of sending frame. • No meaning when in other than slave mode in I²C bus format

Error stop condition detection flag

0	No error stop condition [Clearing] (1) When 0 written after reading ESTP=1; (2) When IRIC flag is cleared to 0.
1	• Error stop condition detected in slave mode in I²C bus format [Setting] On detection of stop condition while sending frame. • No meaning when in other than slave mode in I²C bus format

Note: * Only 0 can be written to these bits (to clear these flags).

ICDR0—I²C Bus Data Register **H'FF7E** **IIC0**
ICDR1—I²C Bus Data Register **H'FF86** **IIC1**

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
R/W	—	—	—	—	—	—	—	—

ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W

TDRE, RDRF (internal flag)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
R/W							—	—

SARX0—2nd Slave Address Register **H'FF7E** **IIC0**
SARX1—2nd Slave Address Register **H'FF86** **IIC1**

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	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
				2nd slave address				Format select X

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit counter

Bit 2	Bit 1	Bit 0	Bit/frame	
BC2	BC1	BC0	Clock sync serial format	PC bus format
0	0	0	8	9
		1	1	2
	1	0	2	3
		1	3	4
1	0	0	4	5
		1	5	6
	1	0	6	7
		1	7	8

Transmit clock select

SCRX bit 5, 6	Bit 5	Bit 4	Bit 3	Clock	Transfer rate					
IICX	CKS2	CKS1	CKS0		ø= 5 MHz	ø= 8 MHz	ø= 10 MHz	ø= 16 MHz	ø= 20 MHz	ø= 25 MHz
0	0	0	0	ø/28	179kHz	286 kHz	357 kHz	571 kHz	714 kHz	893 kHz
			1	ø/40	125kHz	200 kHz	250 kHz	400 kHz	500 kHz	625 kHz
		1	0	ø/48	104kHz	167 kHz	208 kHz	333 kHz	417 kHz	521 kHz
			1	ø/64	78.1kHz	125 kHz	156 kHz	250 kHz	313 kHz	391 kHz
	1	0	0	ø/80	62.5kHz	100 kHz	125 kHz	200 kHz	250 kHz	313 kHz
			1	ø/100	50.0kHz	80.0 kHz	100 kHz	160 kHz	200 kHz	250 kHz
		1	0	ø/112	44.6kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz	223 kHz
			1	ø/128	39.1kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
1	0	0	0	ø/56	89.3kHz	143 kHz	179 kHz	286 kHz	357 kHz	446 kHz
			1	ø/80	62.5kHz	100 kHz	125 kHz	200 kHz	250 kHz	313 kHz
		1	0	ø/96	52.1kHz	83.3 kHz	104 kHz	167 kHz	208 kHz	260 kHz
			1	ø/128	39.1kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
	1	0	0	ø/160	31.3kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz	156 kHz
			1	ø/200	25.0kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz	125 kHz
		1	0	ø/224	22.3kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz	112 kHz
			1	ø/256	19.5kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz	97.7 kHz

Wait insert bit

0	Send data followed by acknowledge bit.
1	Insert wait between data and acknowledge bit.

MSB-first/LSB-first select

0	MSB first
1	LSB first

SAR0—Slave Address Register **H'FF7F** **IIC0**
SAR1—Slave Address Register **H'FF87** **IIC1**

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Slave address

Format select

DDCSWR bit 6	SAR bit 0	SARX bit 0	Operating mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format (initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	—	—	• Must not be set.

ADDRAH—A/D Data Register AH **H'FF90** **A/D**
ADDRAL—A/D Data Register AL **H'FF91** **A/D**
ADDRBH—A/D Data Register BH **H'FF92** **A/D**
ADDRBL—A/D Data Register BL **H'FF93** **A/D**
ADDRCH—A/D Data Register CH **H'FF94** **A/D**
ADDRCL—A/D Data Register CL **H'FF95** **A/D**
ADDRDH—A/D Data Register DH **H'FF96** **A/D**
ADDRDL—A/D Data Register DL **H'FF97** **A/D**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	:	7	6	5	4	3	2	1	0
		ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel select 3

0	AN8 to AN11 set as group 0 analog input pins, and AN12 to AN15 as group 1 analog input pins.
1	AN0 to AN3 set as group 0 analog input pins, and AN4 to AN7 set as group 1 analog input pins.

Scan mode

0	Single mode
1	Scan mode

A/D start

0	A/D conversion disabled.
1	(1) Single mode: A/D conversion starts. Automatically cleared to 0 on completion of conversion on specified channel. (2) Scan mode: A/D conversion starts. The selected channel continues to be sequentially converted until this bit is cleared to 0 by a software, reset, or standby mode is selected, or module stop mode is selected.

A/D interrupt enable

0	A/D conversion end interrupt (ADI) requests disabled.
1	A/D conversion end interrupt (ADI) requests enabled.

A/D end flag

0	[Clearing] (1) Writing 0 to the ADF flag after reading ADF=1. (2) When DTC is started by an ADI interrupt and ADDR is read.
1	[Setting] (1)Single mode: On completion of A/D conversion. (2)Scan mode: On completion of conversion of all specified channels.

Channel select 2 to 0

CH3	CH2	CH1	CH0	Single mode (SCAN= 0)	Scan mode (SCAN= 1)
0	0	0	0	AN0	AN0
			1	AN1	AN0, AN1
		1	0	AN2	AN0 to AN2
			1	AN3	AN0 to AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4 to AN6
			1	AN7	AN4 to AN7
1	0	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8 to AN10
			1	AN11	AN8 to AN11
	1	0	0	AN12	AN12
			1	AN13	AN12, AN13
		1	0	AN14	AN12 to AN14
			1	AN15	AN12 to AN15

Note: * Only 0 can be written to these bits (to clear these flags).

Bit	:	7	6	5	4	3	2	1	0
		TRGS1	TRGS0	—	—	CKS1	CKS0	—	—
Initial value	:	0	0	1	1	0	0	1	1
R/W	:	R/W	R/W	—	—	R/W	R/W	—	—

Clock select 1, 0

CKS1	CKS0	Description
0	0	Conversion time= 530 states (Max.)
	1	Conversion time= 266 states (Max.)
1	0	Conversion time= 134 states (Max.)
	1	Conversion time= 68 states (Max.)

Time trigger select 1, 0

TRGS1	TRGS0	Description
0	0	Enables starting of A/D conversion by software.
	1	Enables starting of A/D conversion by TPU conversion start trigger.
1	0	Enables starting of A/D conversion by 8-bit timer conversion start trigger.
	1	Enables starting of A/D conversion by external trigger pin ($\overline{\text{ADTRG}}$).

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
R/W	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

PSS	CSK2	CSK1	CSK0	Clock	Overflow cycle * (when ø= 25MHz) (when øSUB=32.768kHz)
0	0	0	0	ø/2	20.4 µs
			1	ø/64	652.8 µs
		1	0	ø/128	1.3 ms
			1	ø/512	5.2 ms
	1	0	0	ø/2048	20.9 ms
			1	ø/8192	83.6 ms
		1	0	ø/32768	334.2 ms
			1	ø/131072	1.34 s
1	0	0	0	øSUB/2	15.6 ms
			1	øSUB/4	31.3 ms
		1	0	øSUB/8	62.5 ms
			1	øSUB/16	125 ms
	1	0	0	øSUB/32	250 ms
			1	øSUB/64	500 ms
		1	0	øSUB/128	1 s
			1	øSUB/256	2 s

Note: * The overflow cycle starts when TCNT starts counting from H'00 and ends when an overflow occurs.

Reset or NMI

0	NMI interrupt request
1	Internal reset request

Prescaler select

0	TCNT counts the divided clock output by the ø-based prescaler.
1	TCNT counts the divided clock output by the øSUB-based prescaler (PSS).

Timer enable

0	Initializes TCNT to H'00 and disables the counting operation.
1	TCNT performs counting operation.

Timer mode select

0	Interval timer mode: Interval timer interrupt (WOVI) request sent to CPU when overflow occurs at TCNT.
1	Watchdog timer mode: Reset or NMI interrupt request sent to CPU when overflow occurs at TCNT.

Overflow flag

0	[Clearing] (1) When 0 is written to TME bit; (2) When 0 is written to OVF bit after reading TCSR when OVF=1.
1	[Setting] When TCNT overflows (H'FF→ H'00). When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.

Notes: TCSR is write-protected by a password to prevent accidental overwriting. For details see section 15.2.5, Notes on Register Access.

* Only 0 can be written to these bits (to clear these flags).

Bit	7	6	5	4	3	2	1	0
	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1
Initial value	—*	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Program 1

0	Exits program mode.
1	Enters program mode. [Setting] When FWE=1, SWE1=1, and PSU1=1.

Erase 1

0	Exits erase mode.
1	Enters erase mode. [Setting] When FWE=1, SWE1=1, and ESU1=1.

Program verify 1

0	Exits program verify mode.
1	Enters program verify mode. [Setting] When FWE=1 and SWE1=1.

Erase verify 1

0	Exits erase verify mode.
1	Enters erase verify mode. [Setting] When FWE=1 and SWE1=1

Program setup bit 1

0	Exits program setup.
1	Program setup. [Setting] When FWE=1 and SWE1=1.

Erase setup bit 1

0	Exits erase setup.
1	Erase setup. [Setting] When FWE=1 and SWE1=1.

Software write enable bit 1

0	Writing disabled.
1	Writing enabled. [Setting] When FWE=1.

Flash write enable bit

0	When LOW level signal input to FWE pin (hardware protect status).
1	When HIGH level signal input to FWE pin.

Note: * Determined by the state of the FWE pin.

FLMCR2—Flash Memory Control Register 2 H'FFA9 FLASH

Bit	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R	—	—	—	—	—	—	—

Flash memory error

0	Flash memory operating normally. Flash memory protection against writing and erasing (error protection) is ignored. [Clearing] At a power-on reset and in hardware standby mode.
1	Shows that an error has occurred when writing to or erasing flash memory. Flash memory protection against writing and erasing (error protection) is enabled. [Setting] See “22.8.3 Error Protection.”

EBR1—Erase Block Register 1 H'FFAA FLASH

Bit	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EBR2—Erase Block Register 2 H'FFAB FLASH

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EB11	EB10	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FLPWCR—Flash Memory Power Control Register H'FFAC FLASH

Bit	7	6	5	4	3	2	1	0
	PDWND	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Power-down disable

0	Transition to flash memory power-down mode enabled
1	Transition to flash memory power-down mode disabled

PORT1—Port 1 Register H'FFB0 Port

Bit	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P11	P10
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Note: * Determined by status of pins P17 to P10.

PORT3—Port 3 Register H'FFB2 Port

Bit	7	6	5	4	3	2	1	0
	P37	P36	P35	P34	P33	P32	P31	P30
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Note: * Determined by status of pins P37 to P30.

PORT4—Port 4 Register **H'FFB3** **Port**

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by status of pins P47 to P40.

PORT7—Port 7 Register **H'FFB6** **Port**

Bit	:	7	6	5	4	3	2	1	0
		P77	P76	P75	P74	P73	P72	P71	P70
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by status of pins P77 to P70.

PORT9—Port 9 Register **H'FFB8** **Port**

Bit	:	7	6	5	4	3	2	1	0
		P97	P96	P95	P94	P93	P92	P91	P90
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by status of pins P97 to P90.

PORTA—Port A Register **H'FFB9** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3	PA2	PA1	PA0
Initial value	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	—	—	—	—	R	R	R	R

Note: * Determined by status of pins PA3 to PA0.

PORTB—Port B Register **H'FFBA** **Port**

Bit	7	6	5	4	3	2	1	0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Note: * Determined by status of pins PB7 to PB0.

PORTC—Port C Register **H'FFBB** **Port**

Bit	7	6	5	4	3	2	1	0
	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Note: * Determined by status of pins PC7 to PC0.

PORTD—Port D Register **H'FFBC** **Port**

Bit	7	6	5	4	3	2	1	0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
R/W	R	R	R	R	R	R	R	R

Note: * Determined by status of pins PC7 to PC0.

PORTE—Port E Register **H'FFBD** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by status of pins PD7 to PD0.

PORTF—Port F Register **H'FFBE** **Port**

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

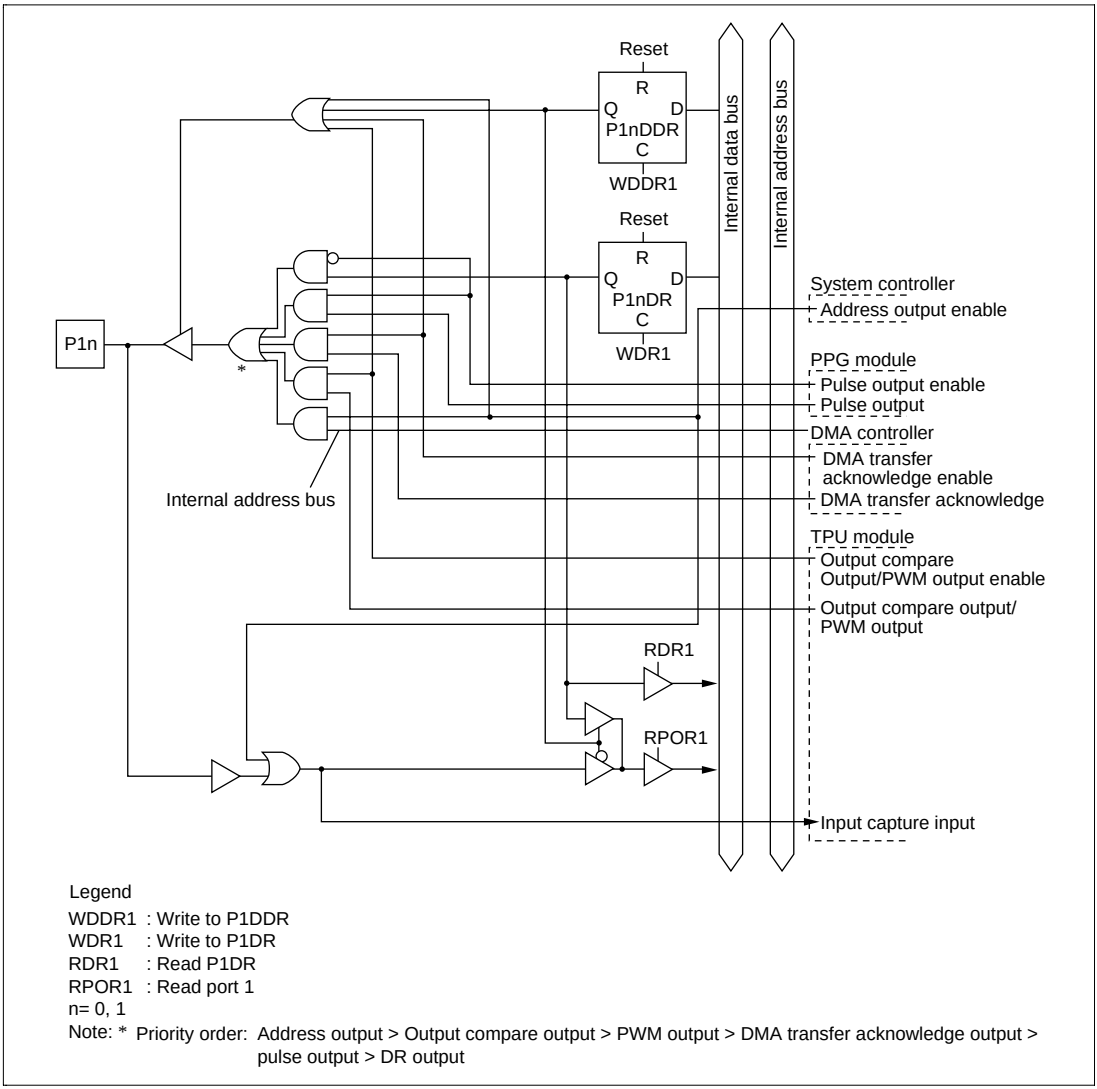
Note: * Determined by status of pins PF7 to PF0.

PORTG—Port G Register **H'FFBF** **Port**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R	R

Note: * Determined by status of pins PG4 to PG0.

C.1 Port 1 Block Diagram



Legend
WDDR1 : Write to P1DDR
WDR1 : Write to P1DR
RDR1 : Read P1DR
RPOR1 : Read port 1
n= 0, 1
Note: * Priority order: Address output > Output compare output > PWM output > DMA transfer acknowledge output > pulse output > DR output

Figure C-1 (a) Port 1 Block Diagram (Pins P10 and P11)

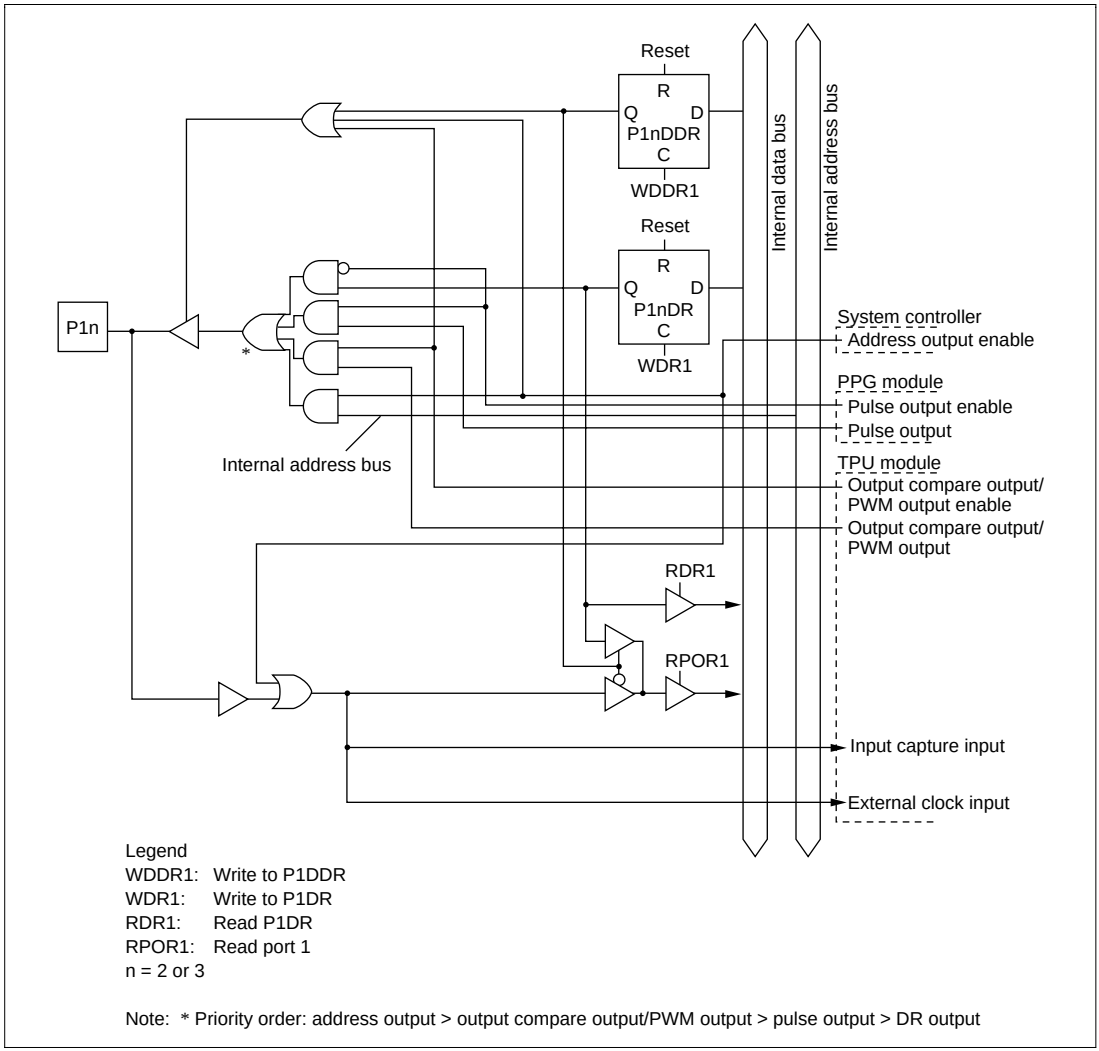


Figure C-1 (b) Port 1 Block Diagram (Pins P12 and P13)

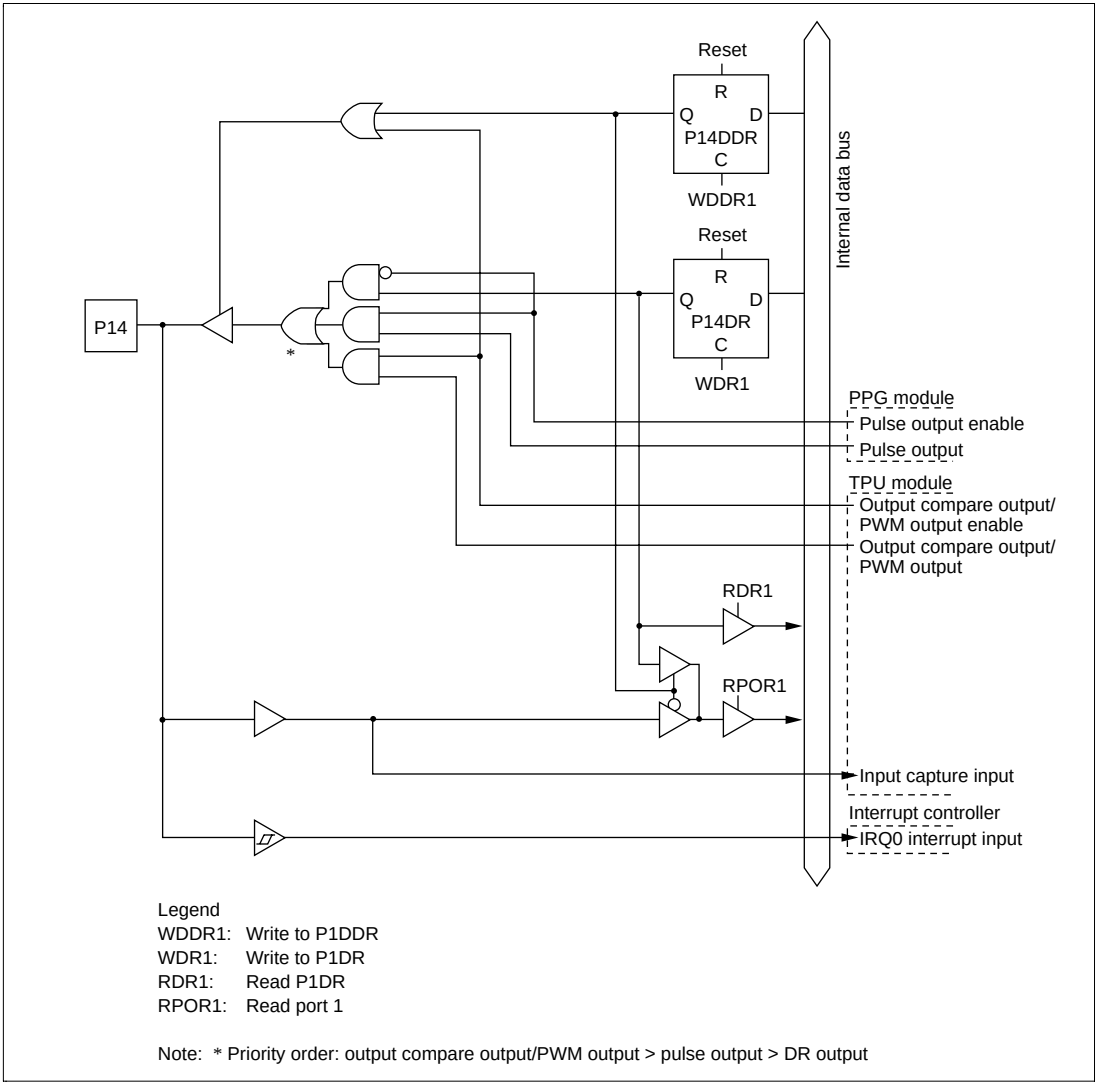


Figure C-1 (c) Port 1 Block Diagram (Pin P14)

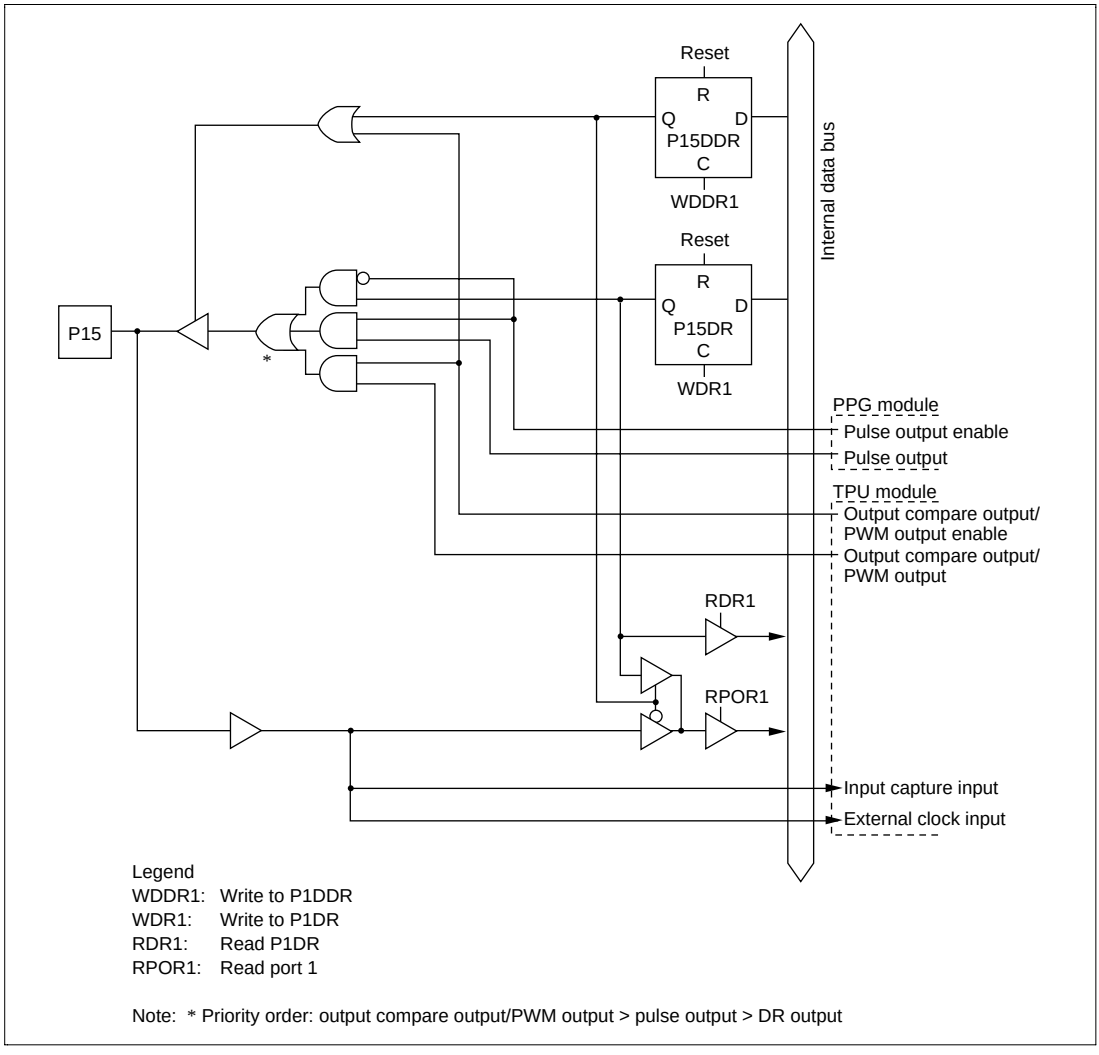


Figure C-1 (d) Port 1 Block Diagram (Pin P15)

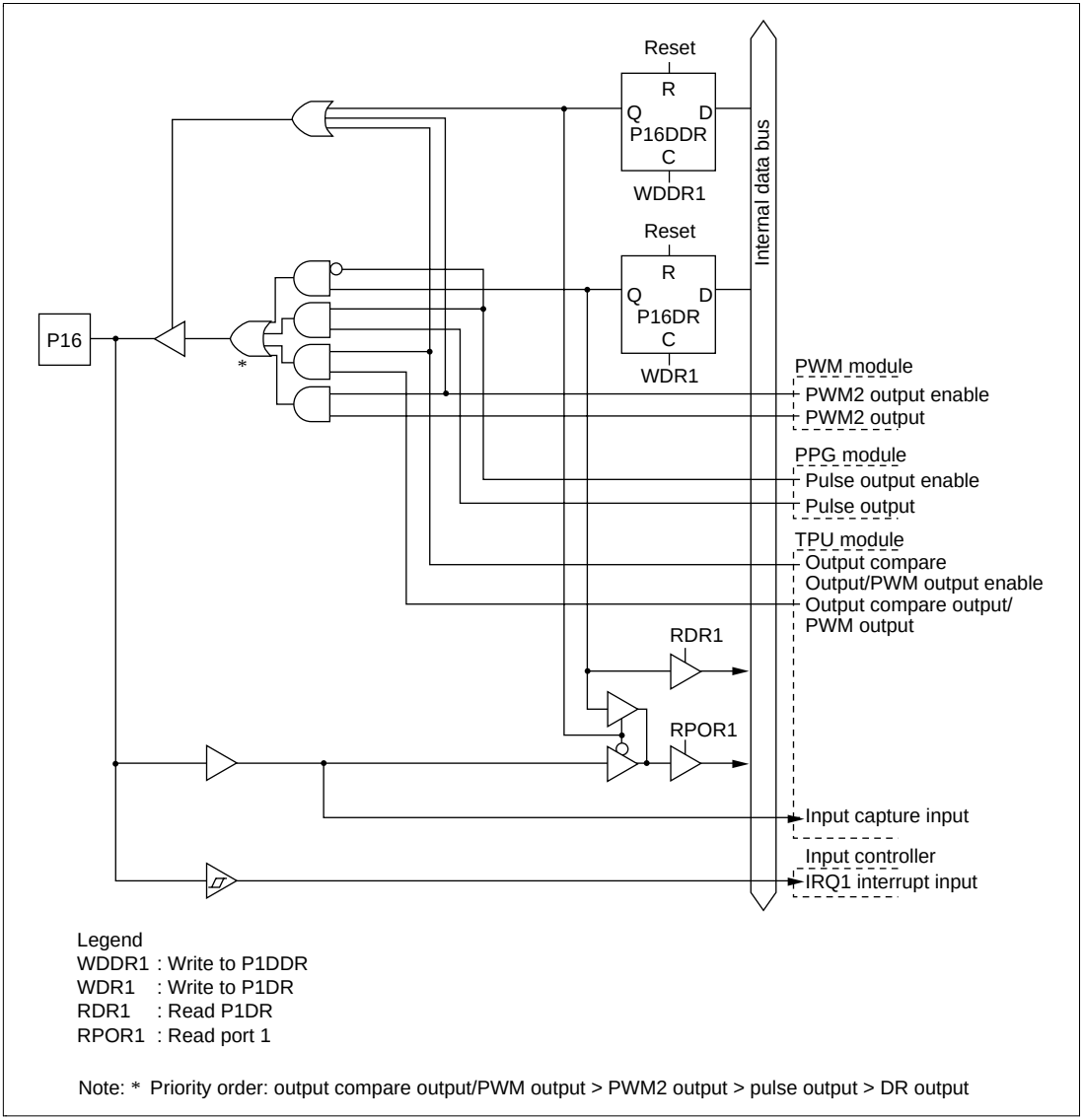


Figure C-1 (e) Port 1 Block Diagram (Pin P16)

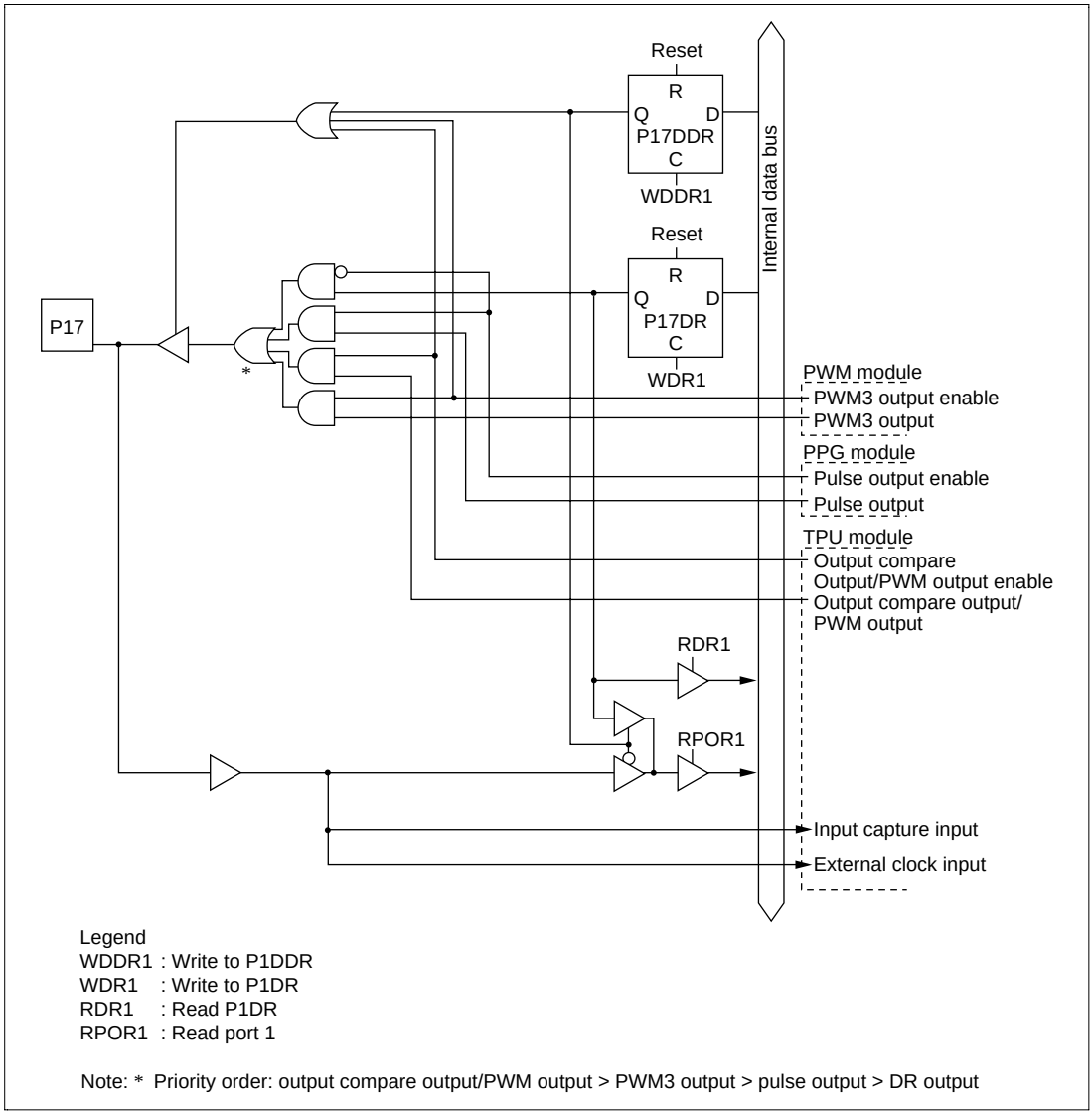


Figure C-1 (f) Port 1 Block Diagram (Pin P17)

C.2 Port 3 Block Diagram

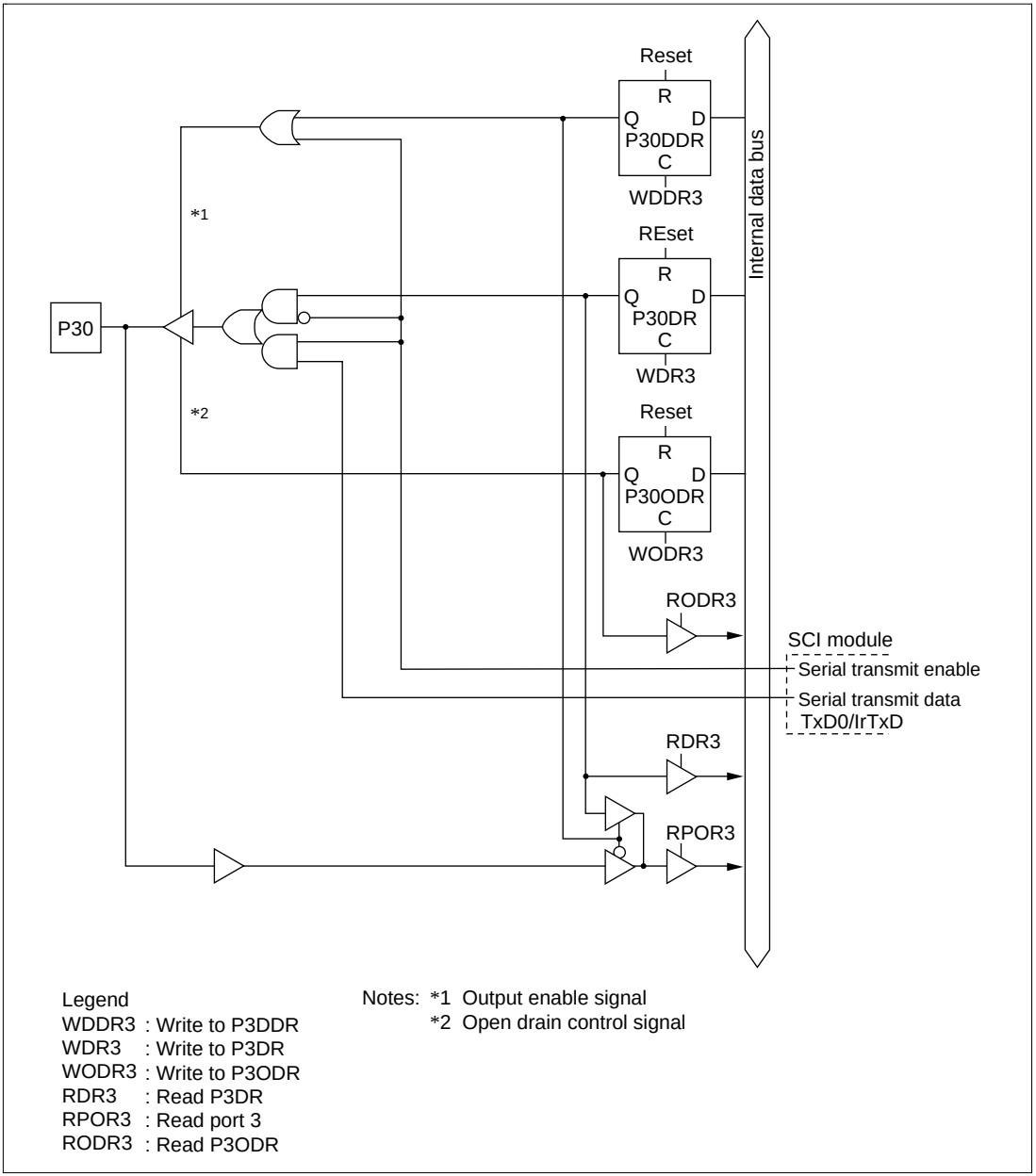


Figure C-2 (a) Port 3 Block Diagram (Pin P30)

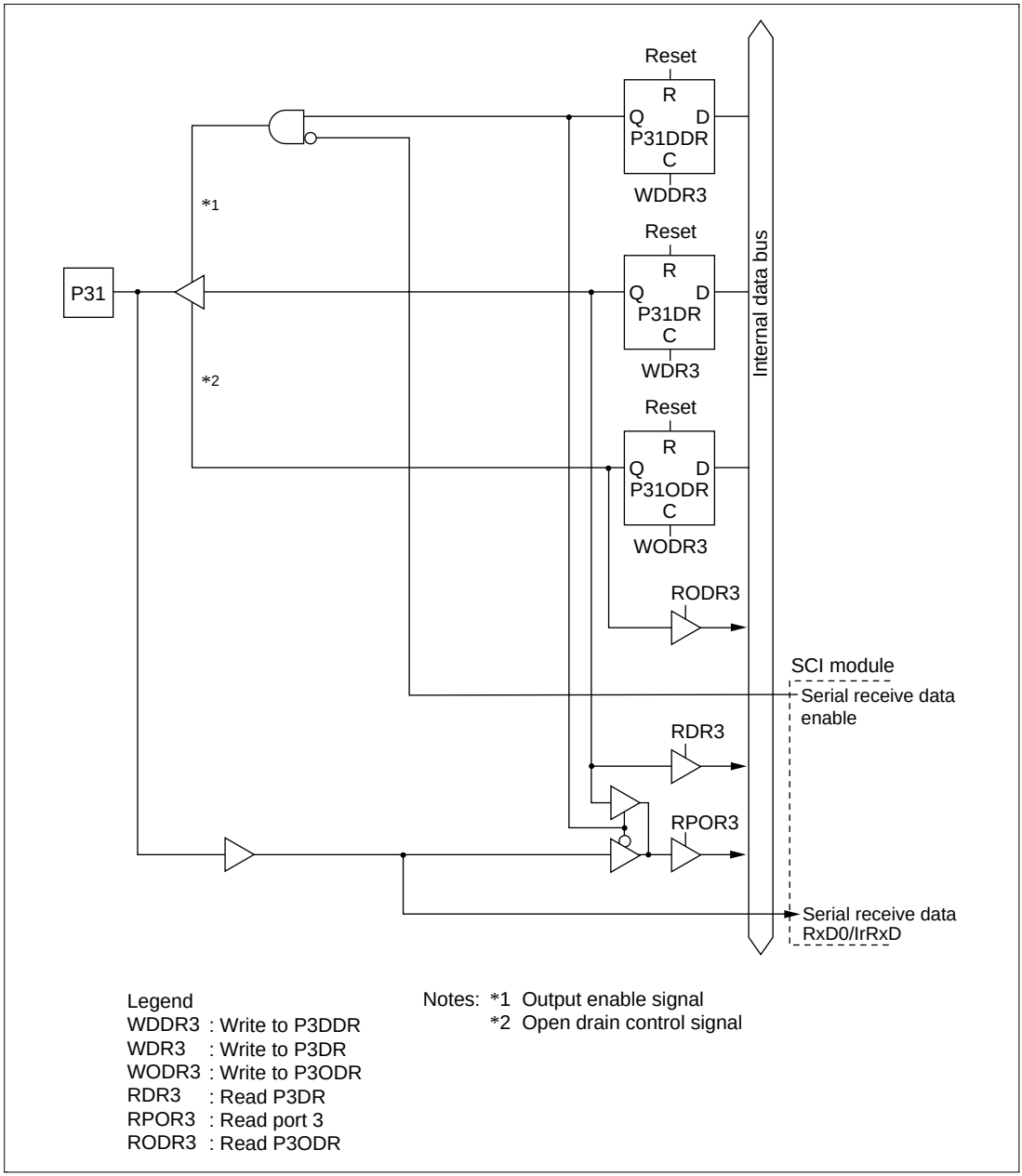


Figure C-2 (b) Port 3 Block Diagram (Pin P31)

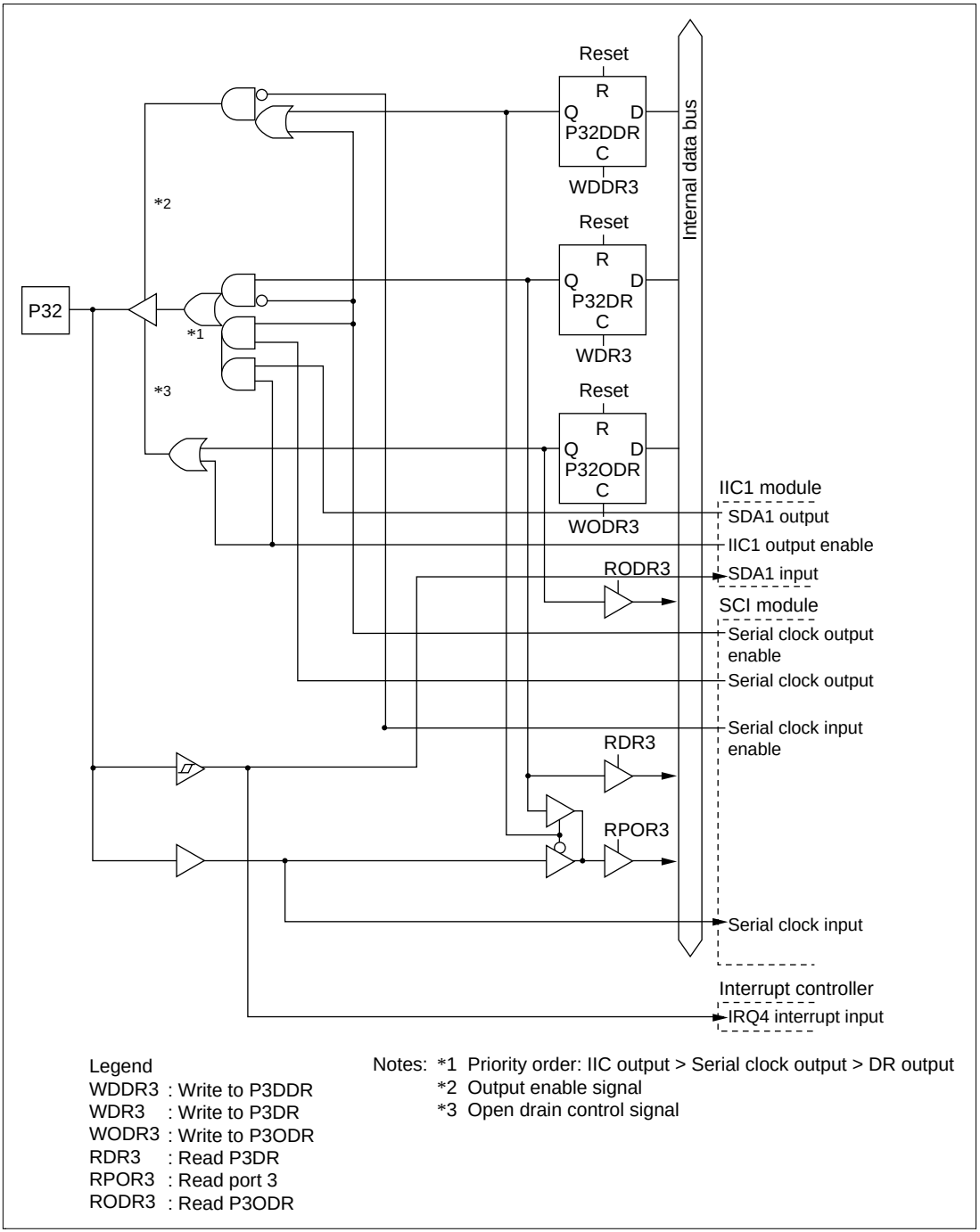


Figure C-2 (c) Port 3 Block Diagram (Pin P32)

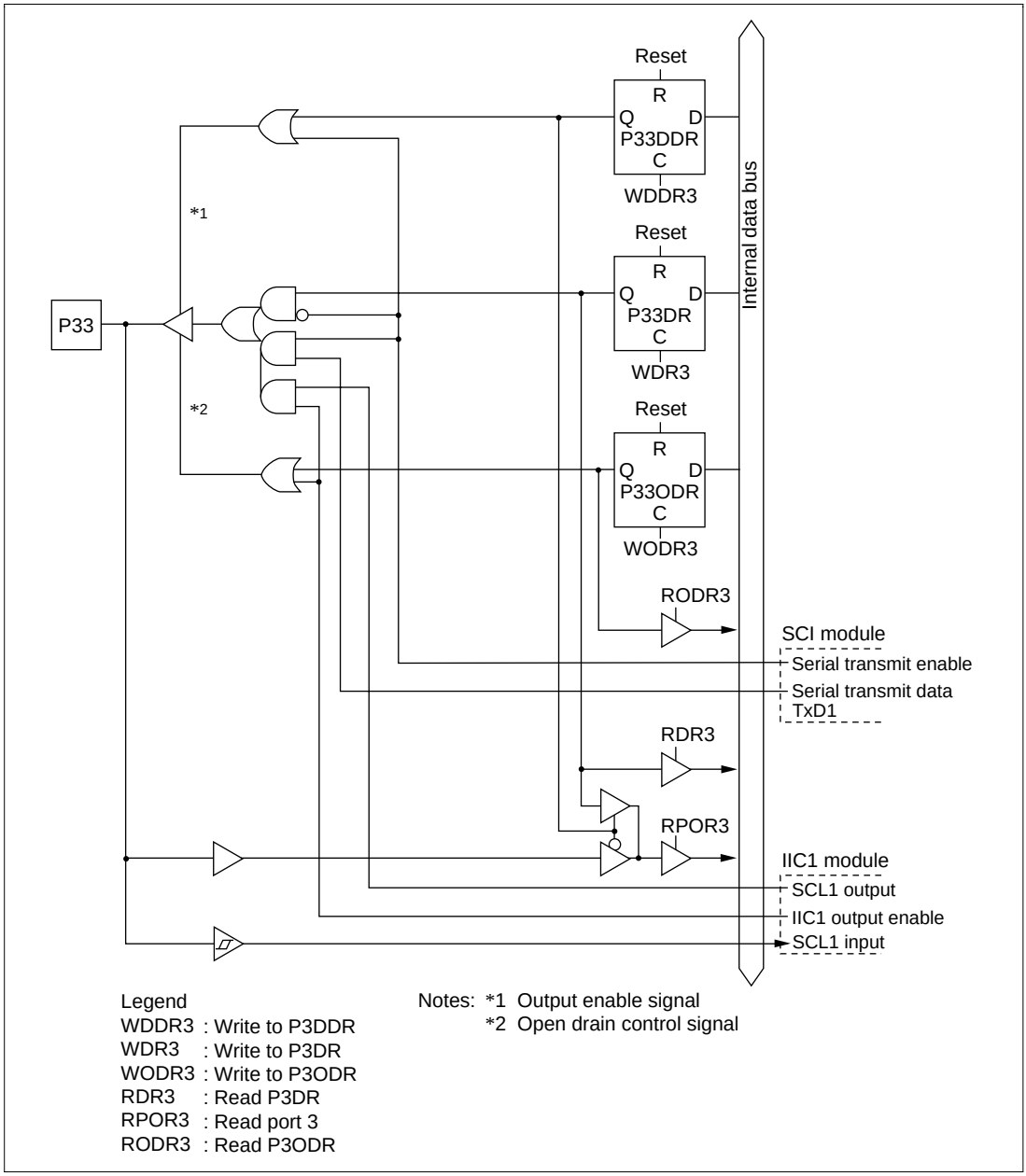


Figure C-2 (d) Port 3 Block Diagram (Pin P33)

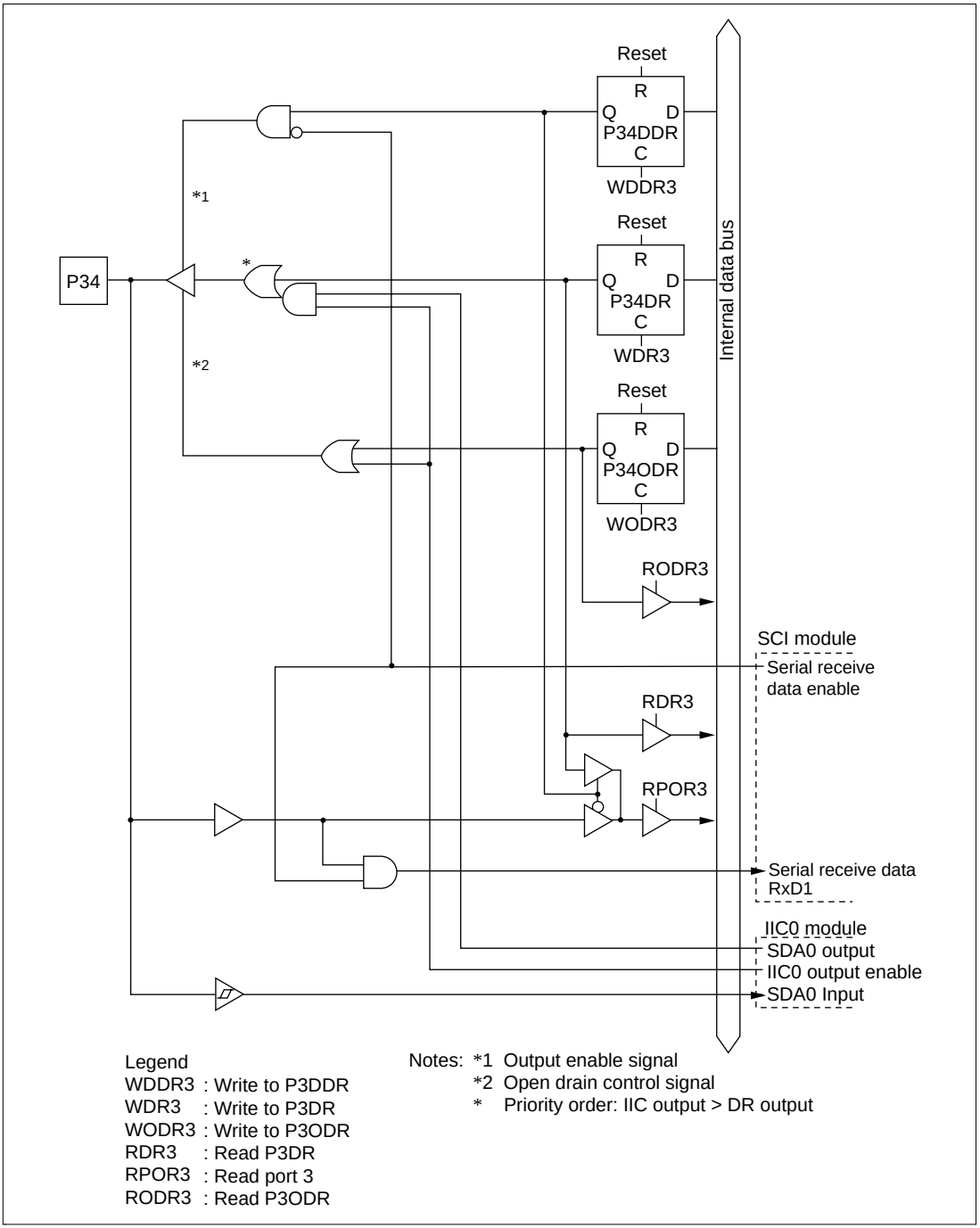
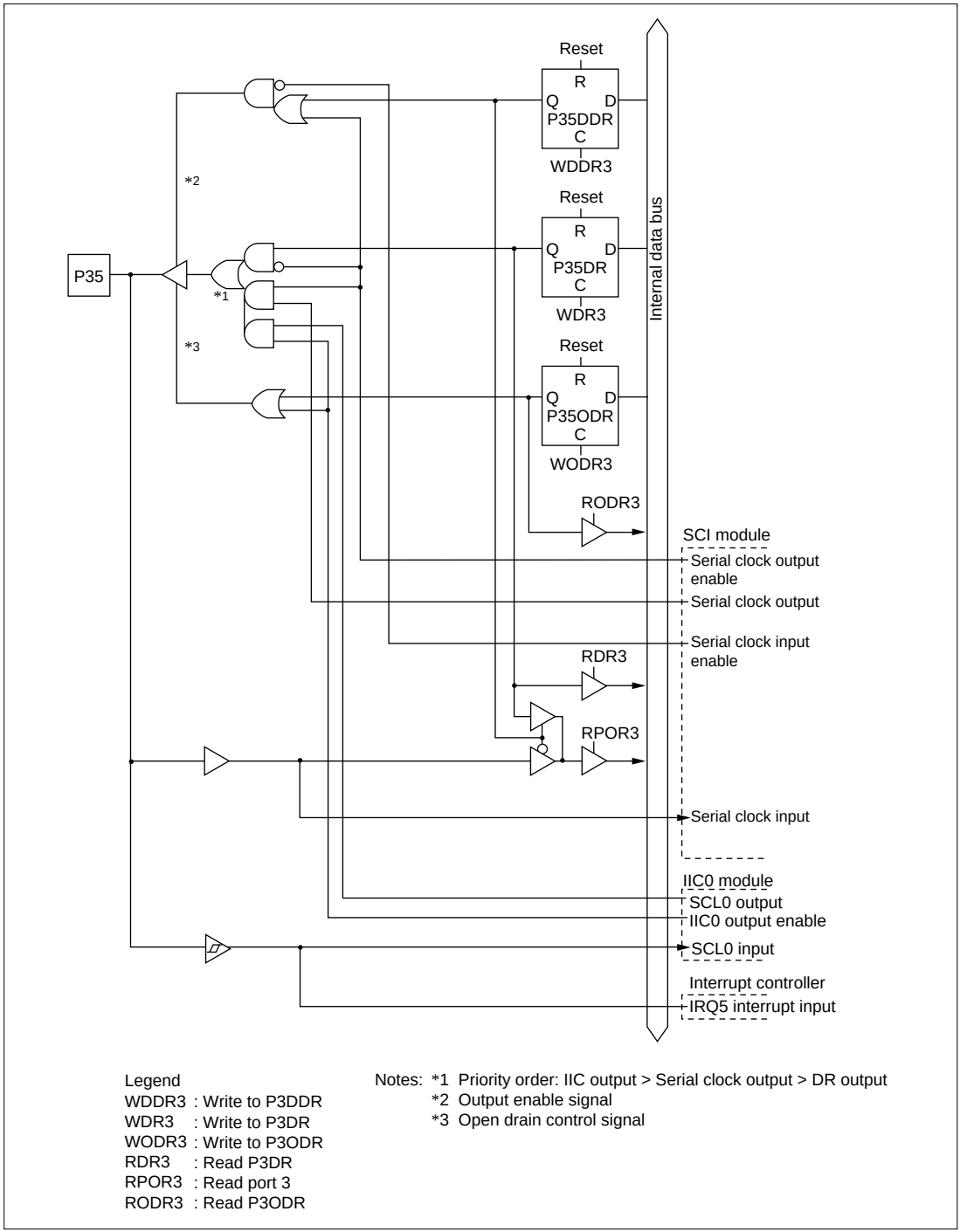


Figure C-2 (e) Port 3 Block Diagram (Pin P34)



Legend
WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR

Notes: *1 Priority order: IIC output > Serial clock output > DR output
*2 Output enable signal
*3 Open drain control signal

Figure C-2 (f) Port 3 Block Diagram (Pin P35)

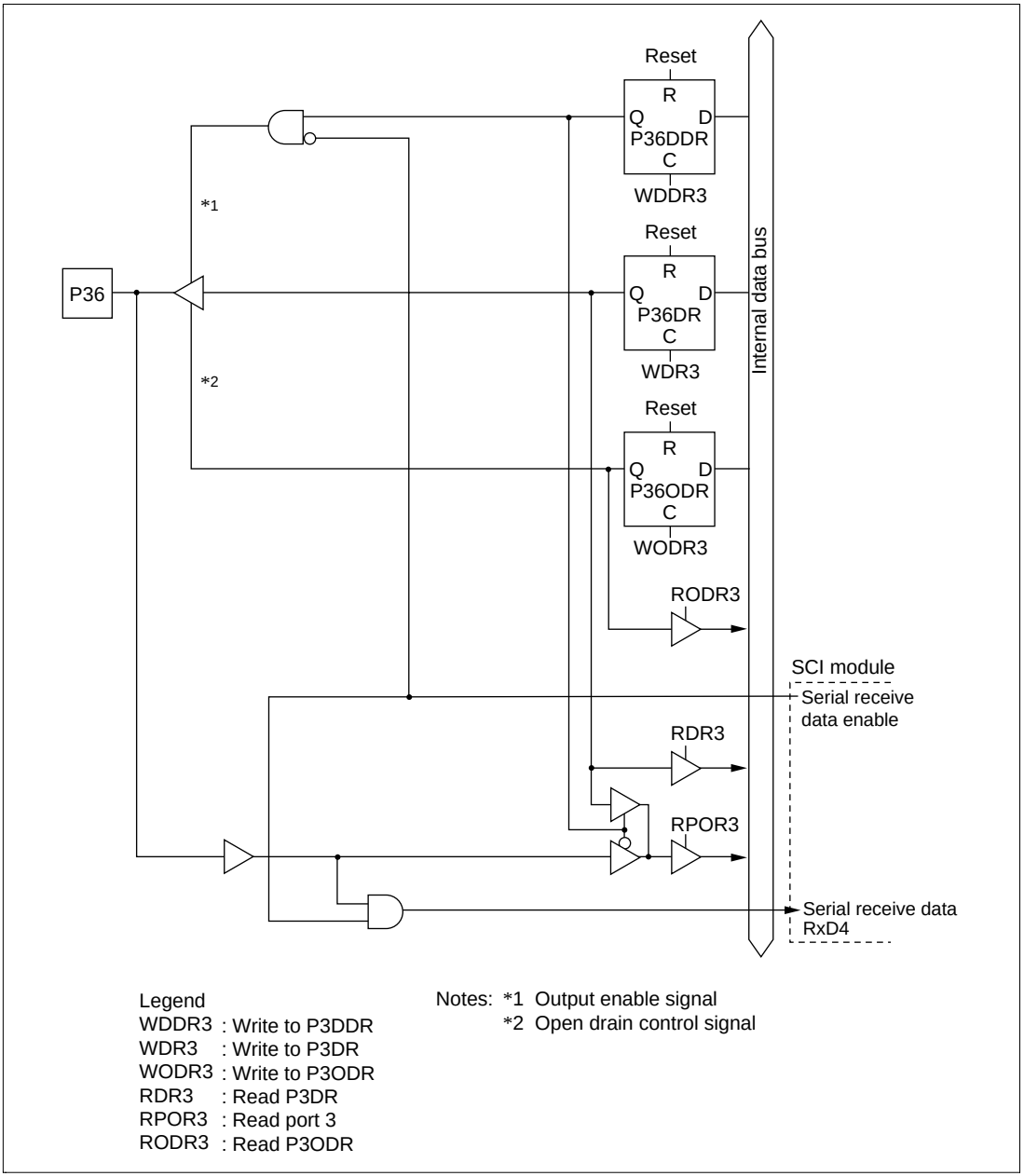


Figure C-2 (g) Port 3 Block Diagram (Pin P36)

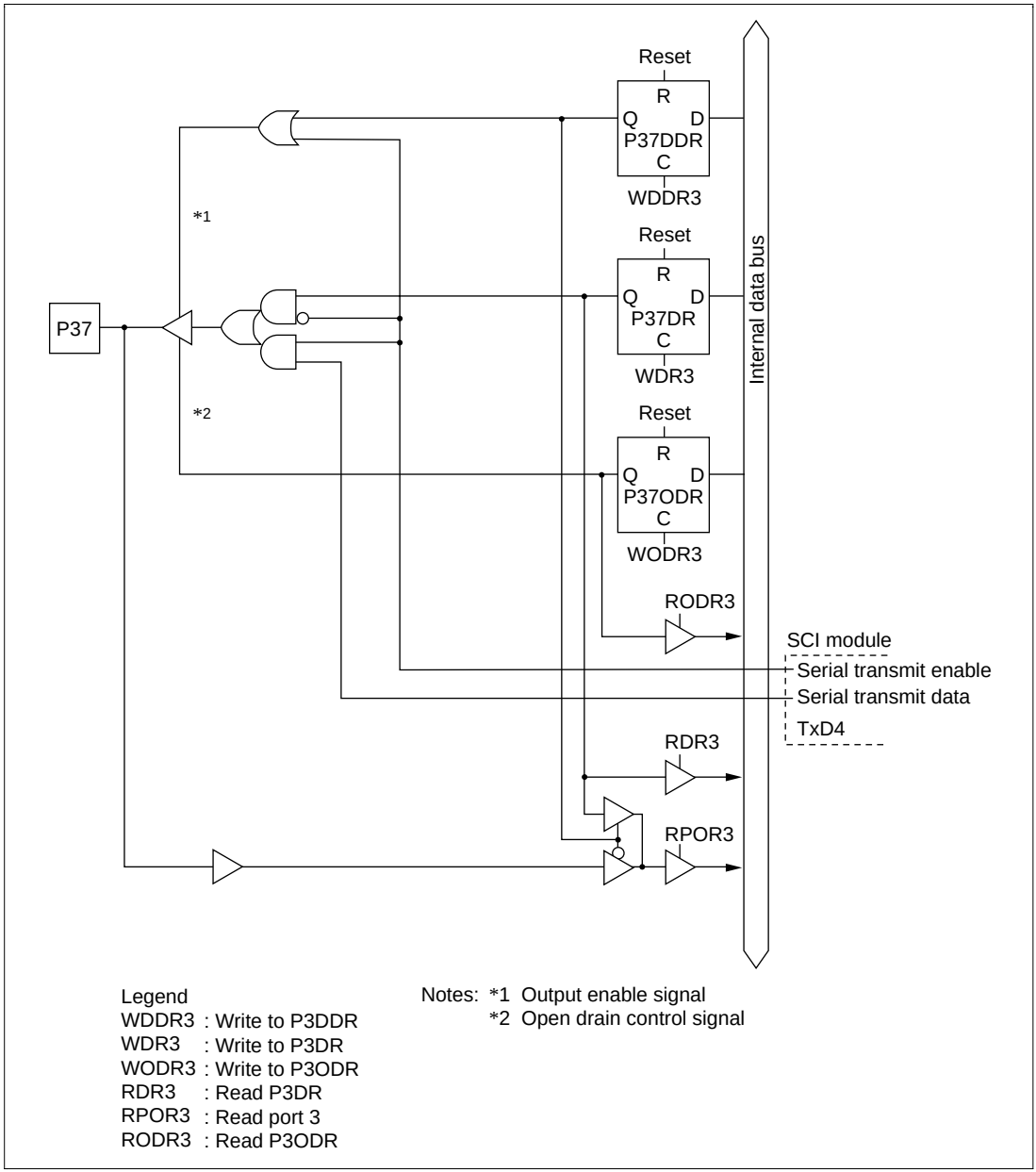


Figure C-2 (h) Port 3 Block Diagram (Pin P37)

C.3 Port 4 Block Diagram

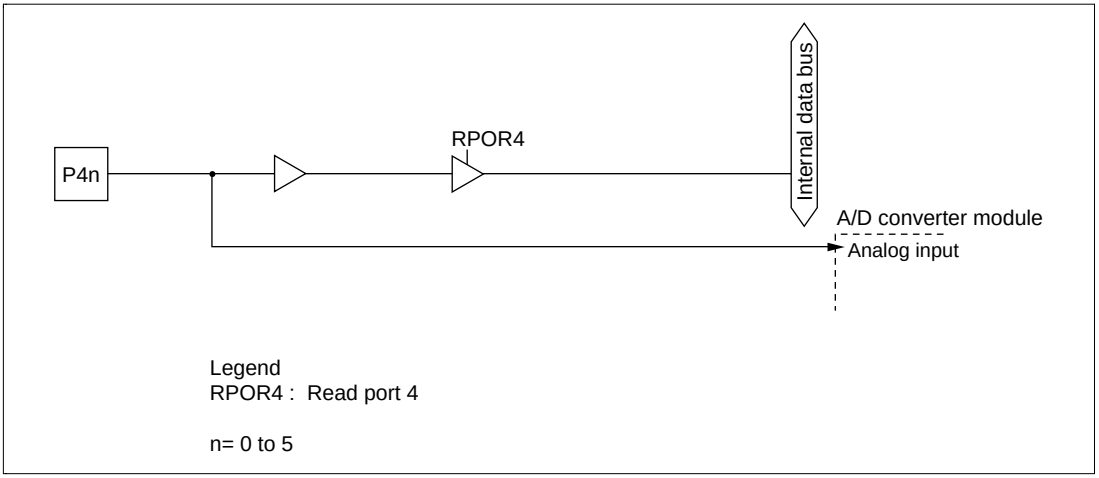


Figure C-3 (a) Port 4 Block Diagram (Pins P40 to P45)

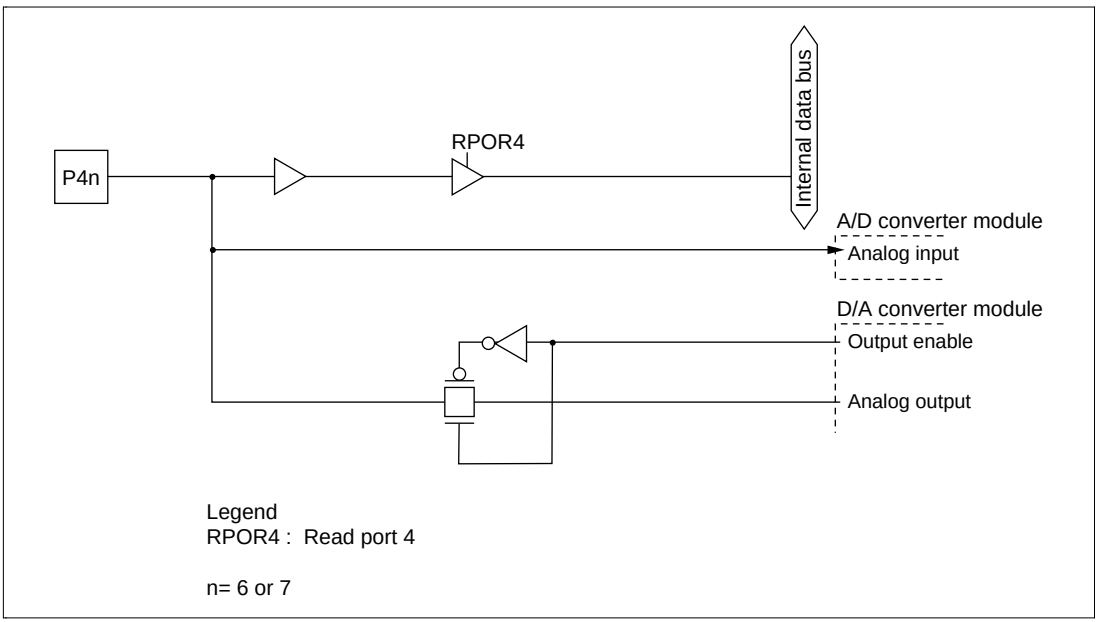


Figure C-3 (b) Port 4 Block Diagram (Pins P46 and P47)

C.4 Port 7 Block Diagram

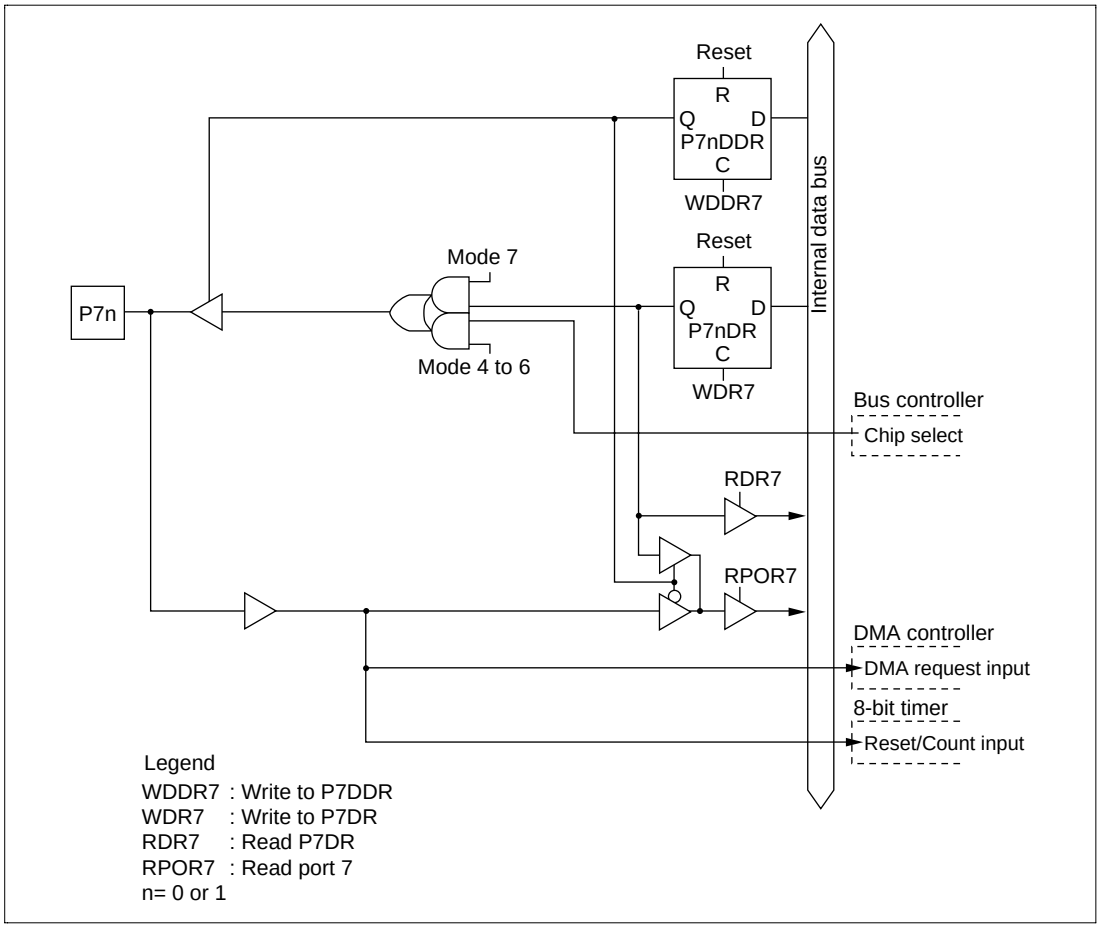


Figure C-4 (a) Port 7 Block Diagram (Pins P70 and P71)

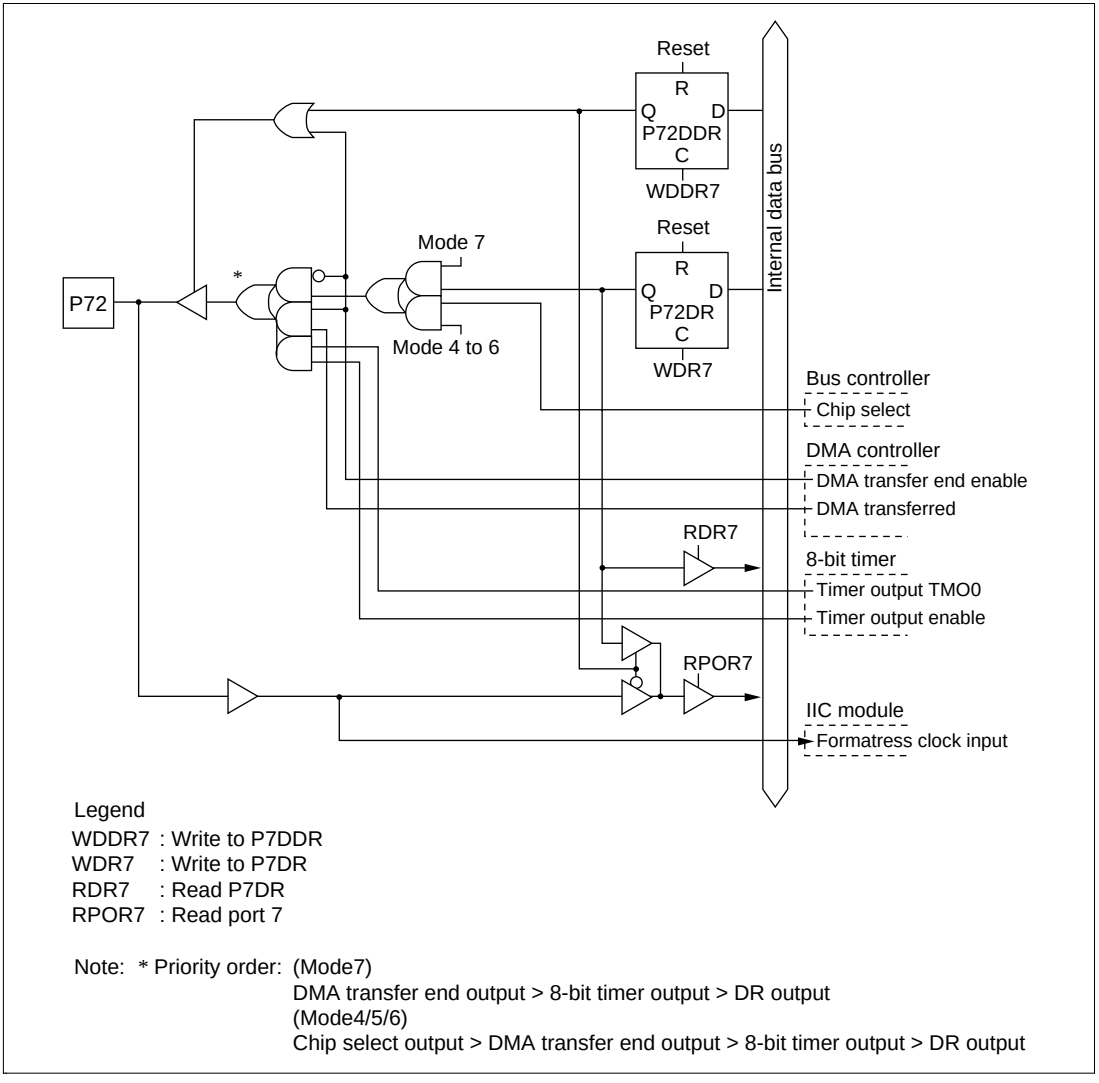
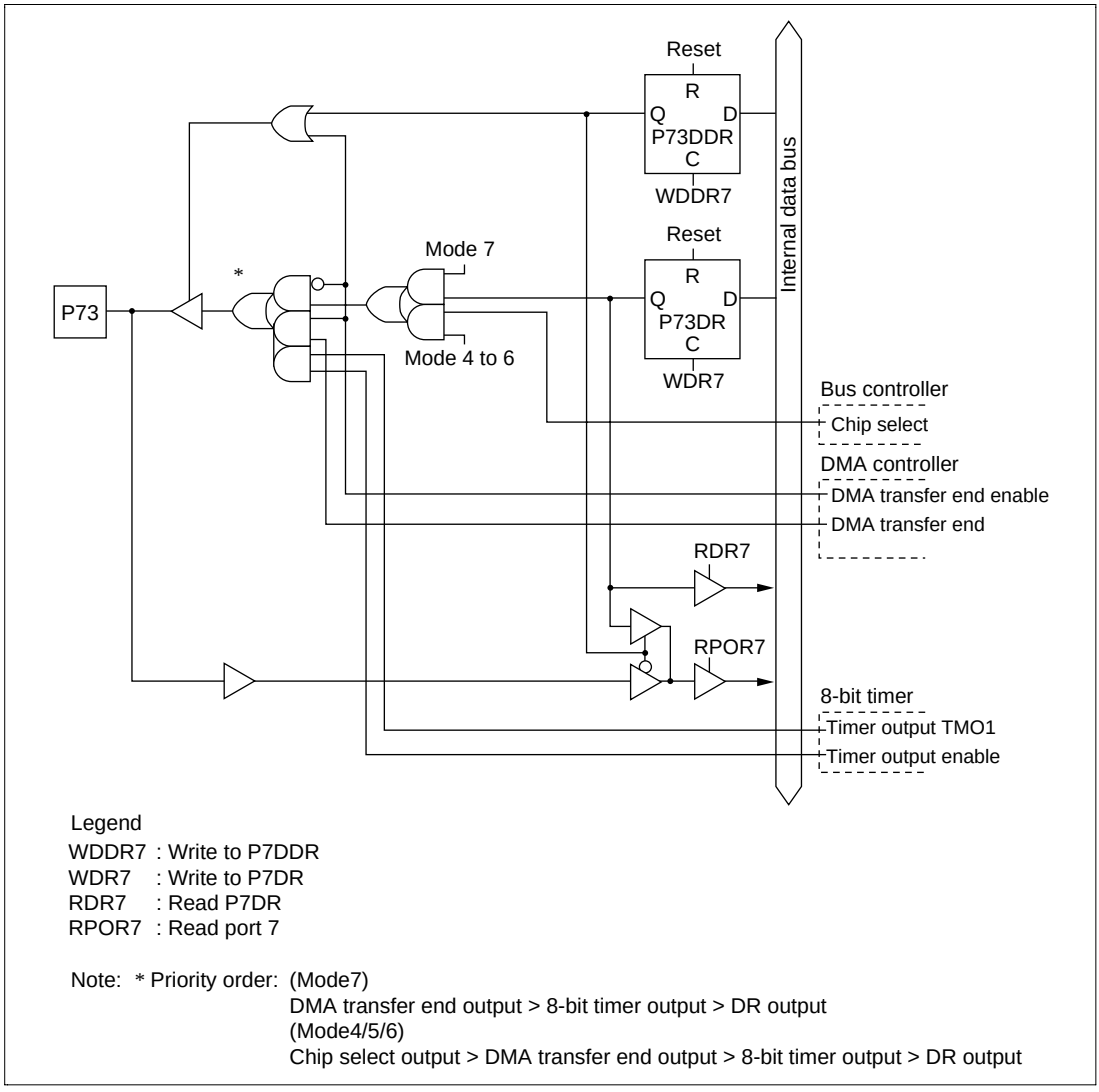


Figure C-4 (b) Port 7 Block Diagram (Pin P72)



Legend

WDDR7 : Write to P7DDR
WDR7 : Write to P7DR
RDR7 : Read P7DR
RPOR7 : Read port 7

Note: * Priority order: (Mode7)
DMA transfer end output > 8-bit timer output > DR output
(Mode4/5/6)
Chip select output > DMA transfer end output > 8-bit timer output > DR output

Figure C-4 (c) Port 7 Block Diagram (Pin P73)

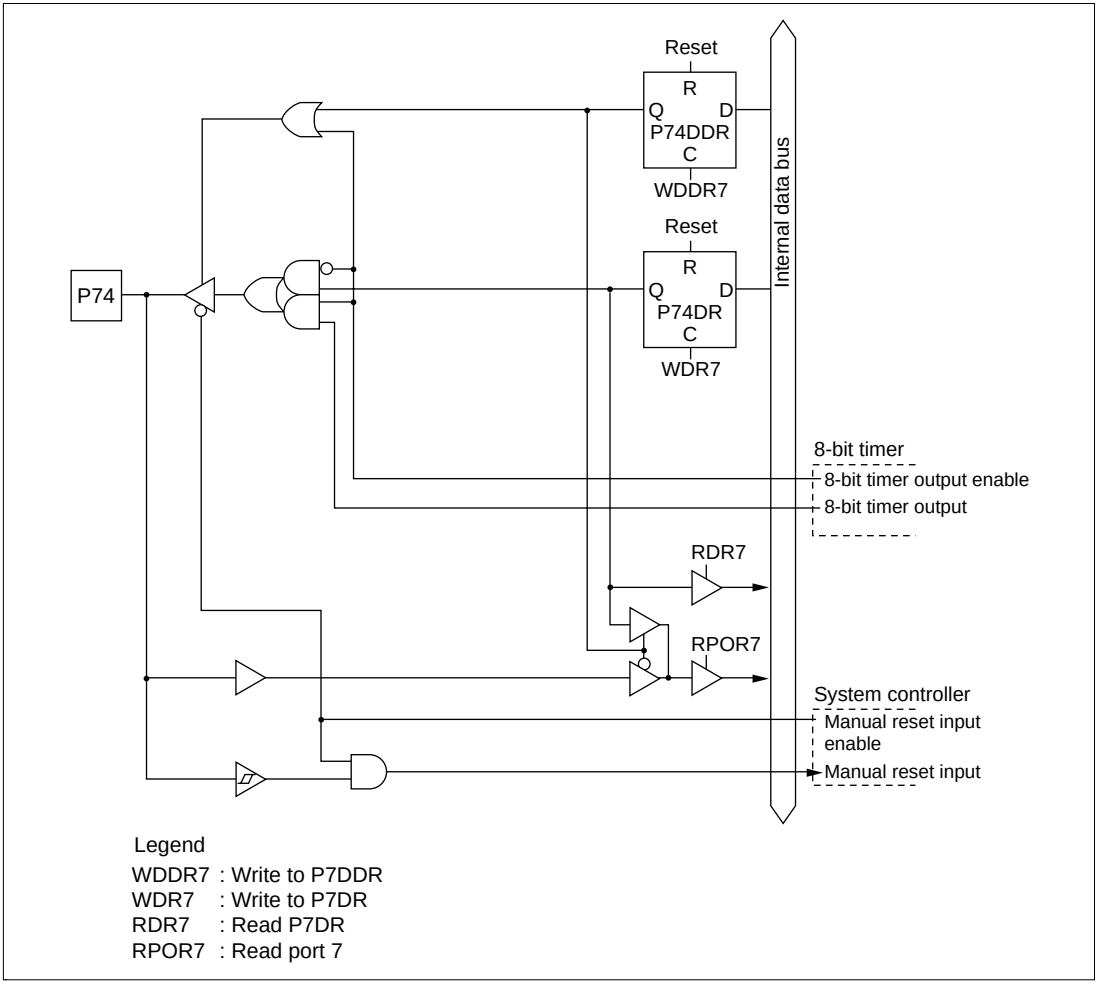


Figure C-4 (d) Port 7 Block Diagram (Pin P74)

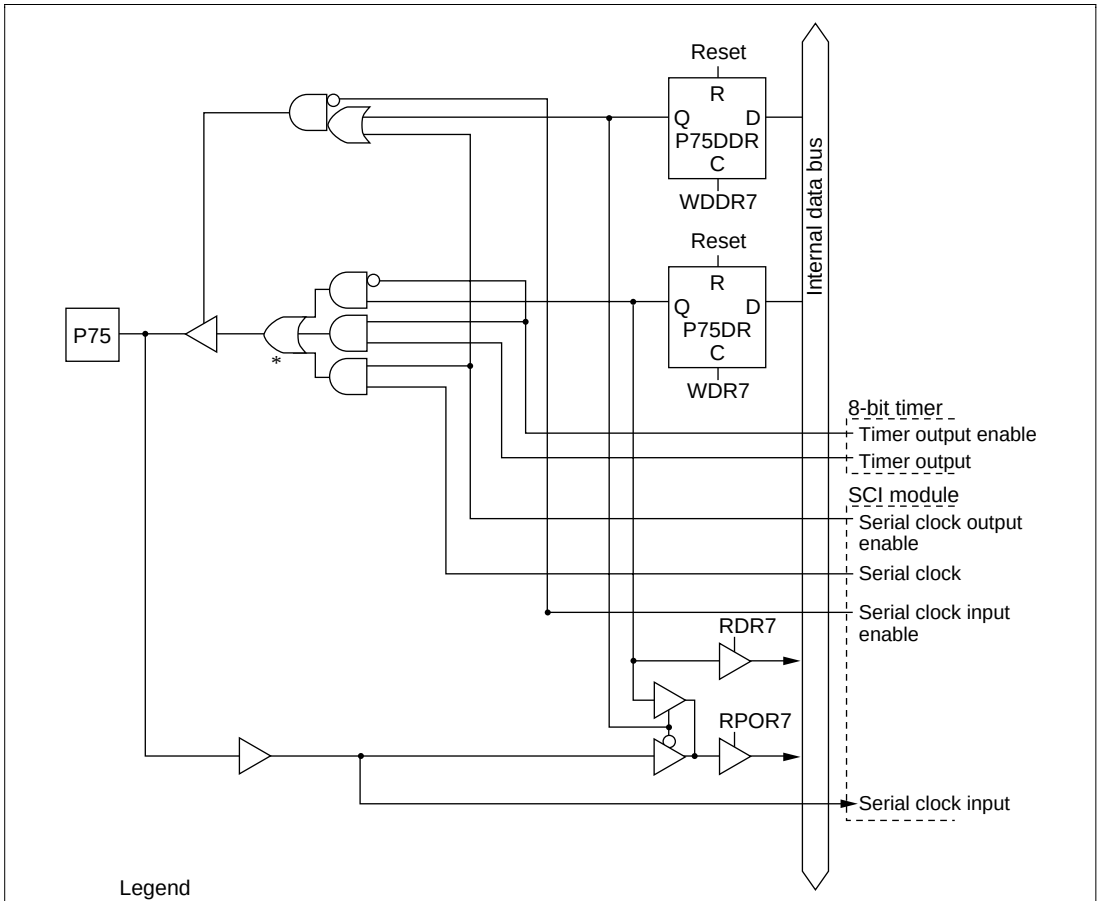


Figure C-4 (e) Port 7 Block Diagram (Pin P75)

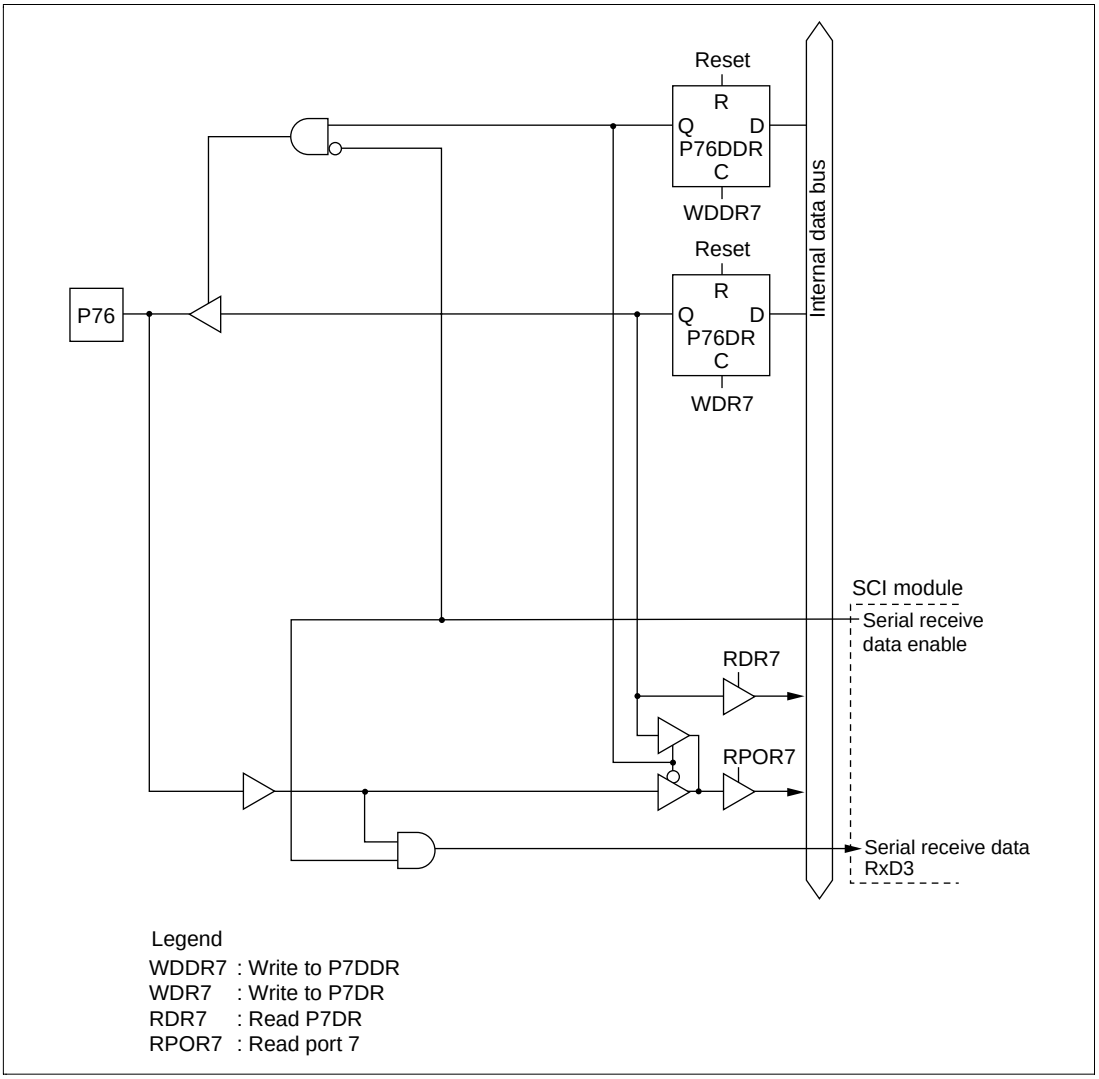


Figure C-4 (f) Port 7 Block Diagram (Pin P76)

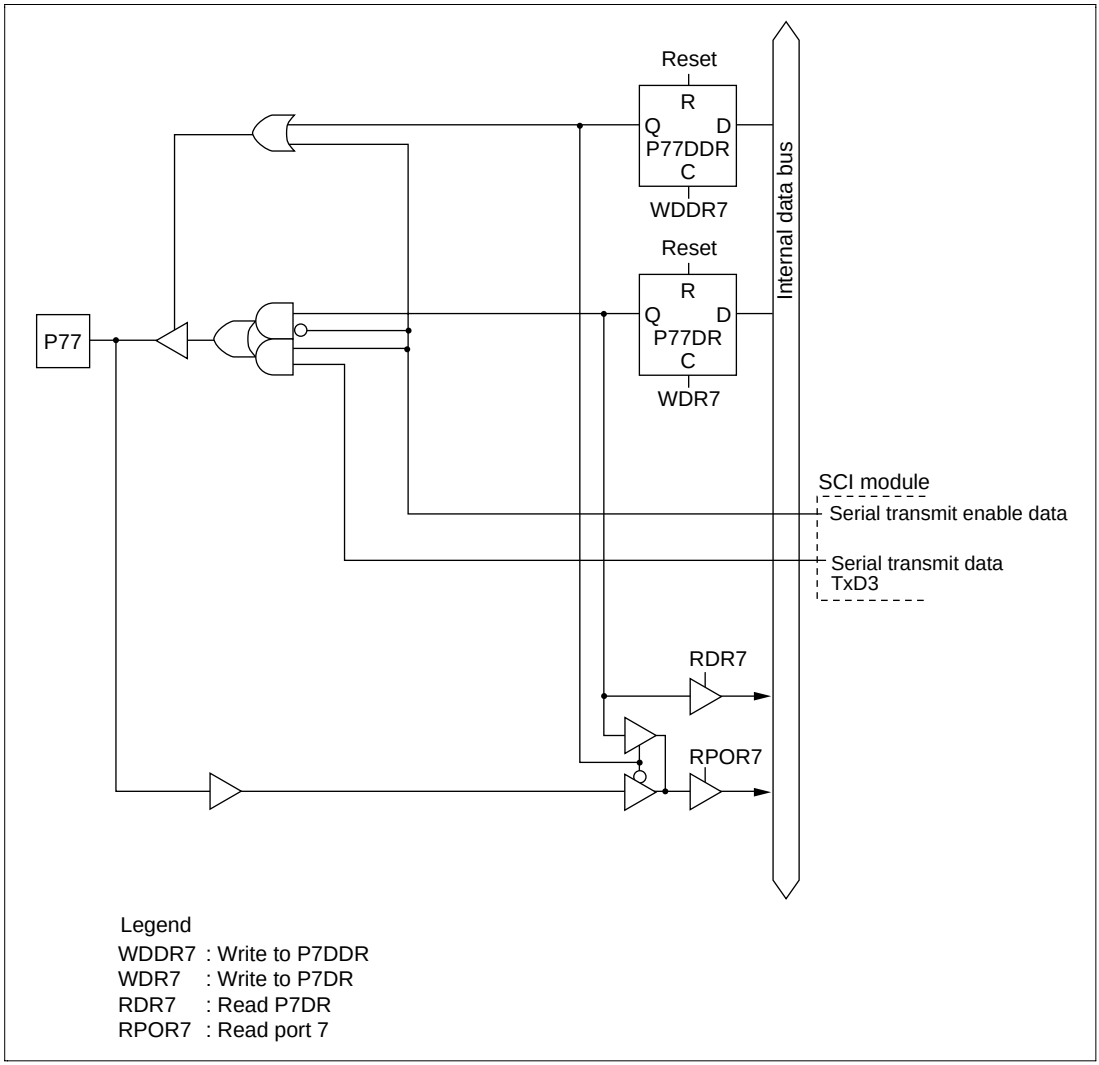


Figure C-4 (g) Port 7 Block Diagram (Pin P77)

C.5 Port 9 Block Diagram

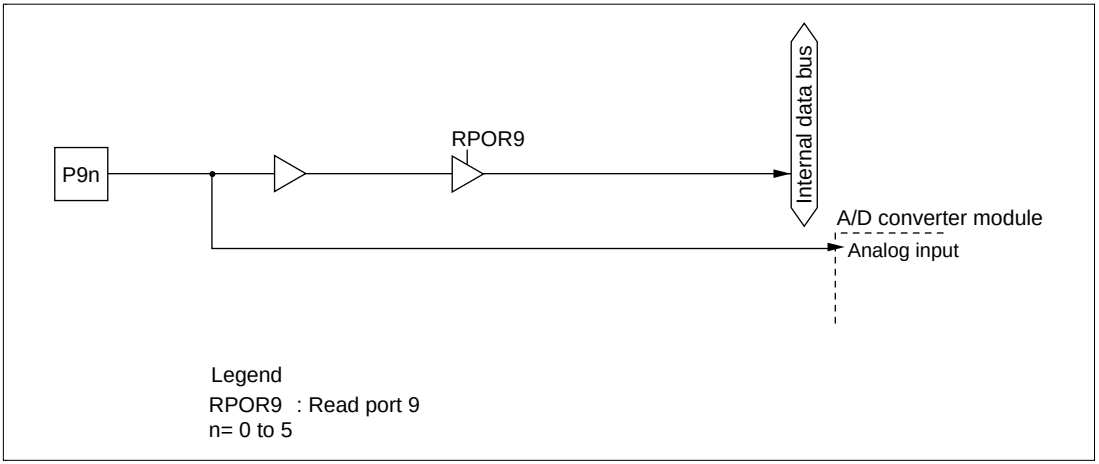


Figure C-5 (a) Port 9 Block Diagram (Pins P90 to P95)

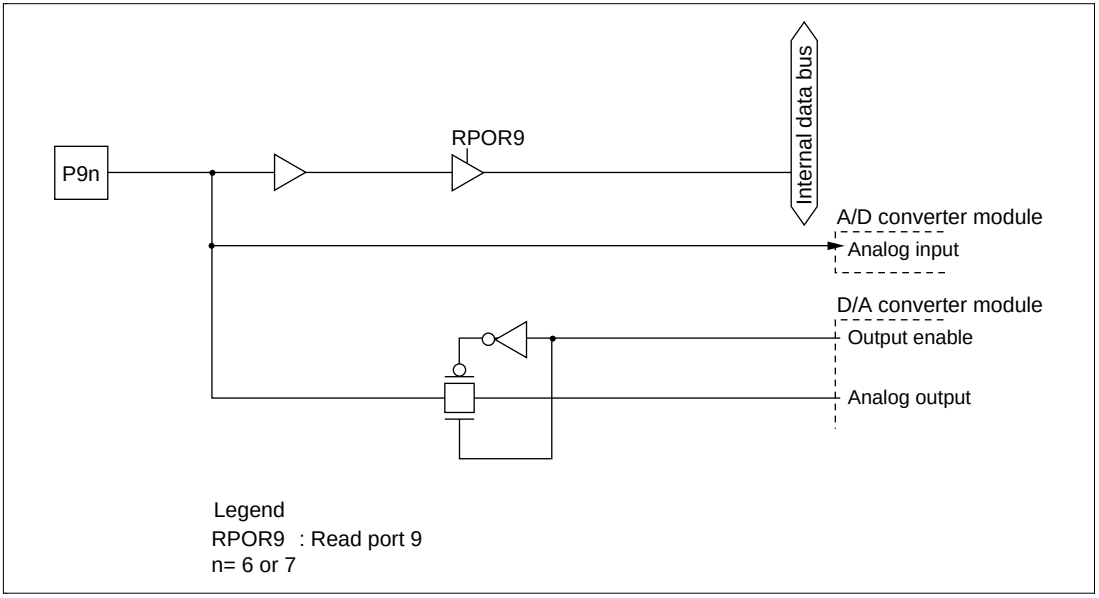


Figure C-5 (b) Port 9 Block Diagram (Pins P96 and P97)

C.6 Port A Block Diagram

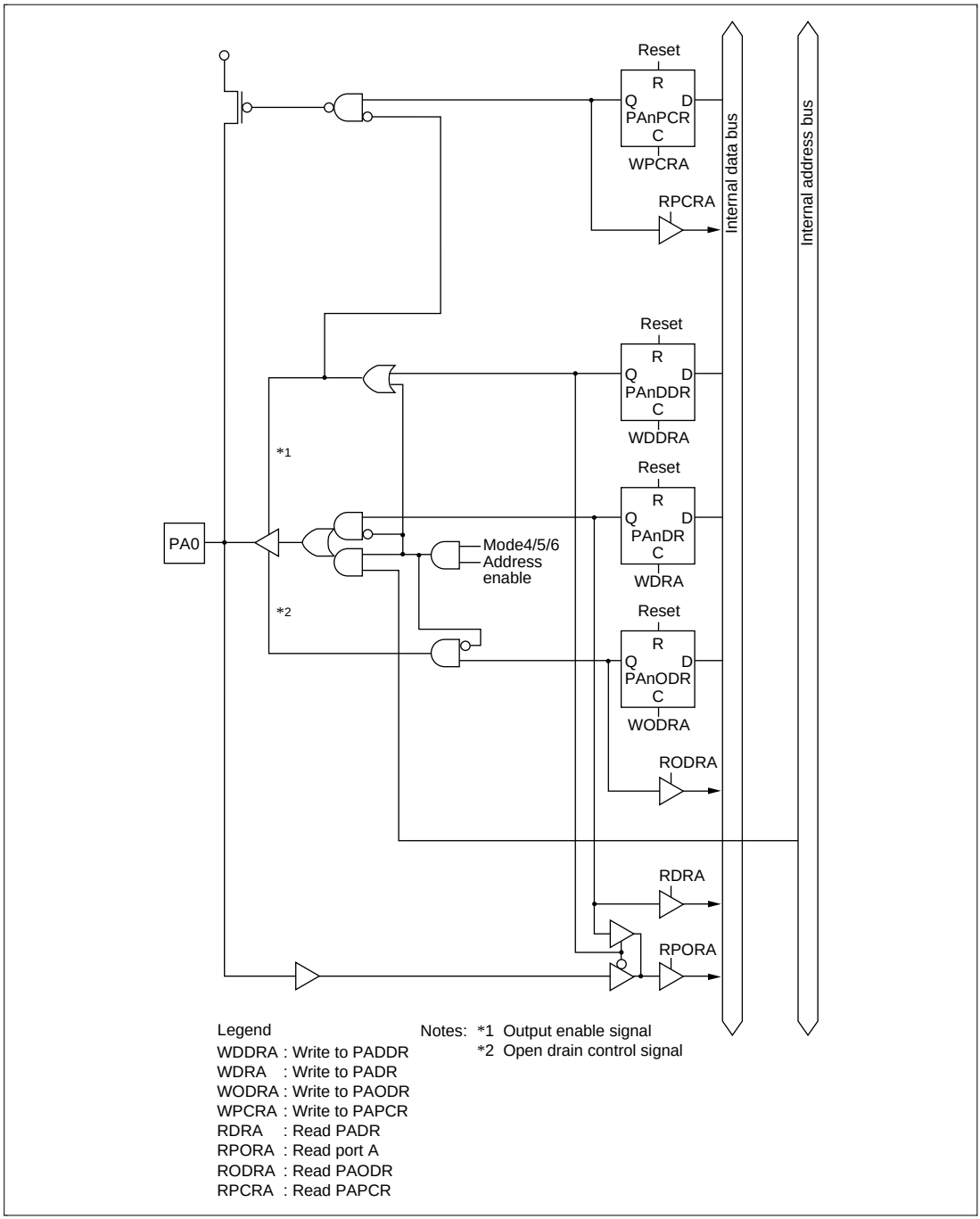


Figure C-6 (a) Port A Block Diagram (Pin PA0)

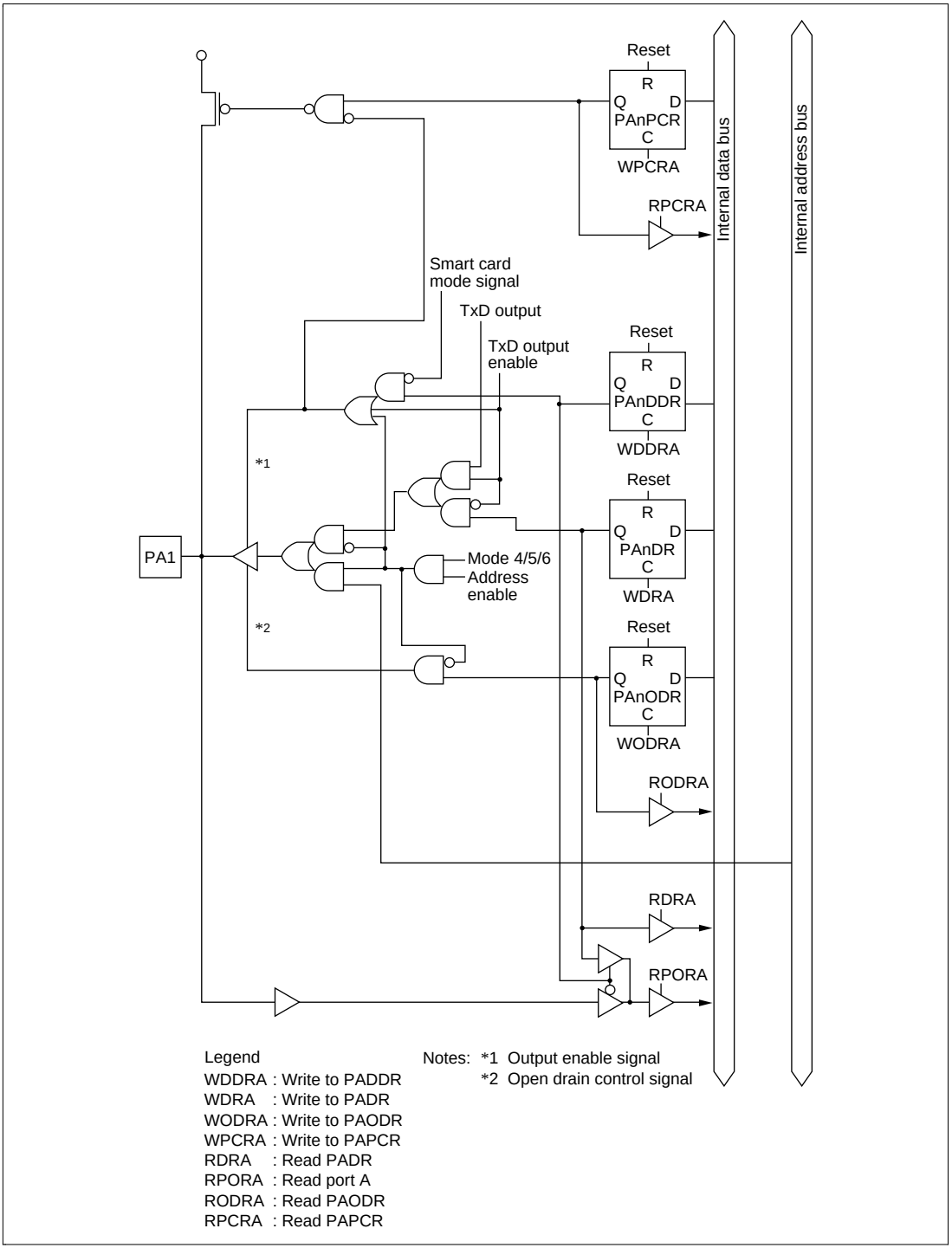


Figure C-6 (b) Port A Block Diagram (Pin PA1)

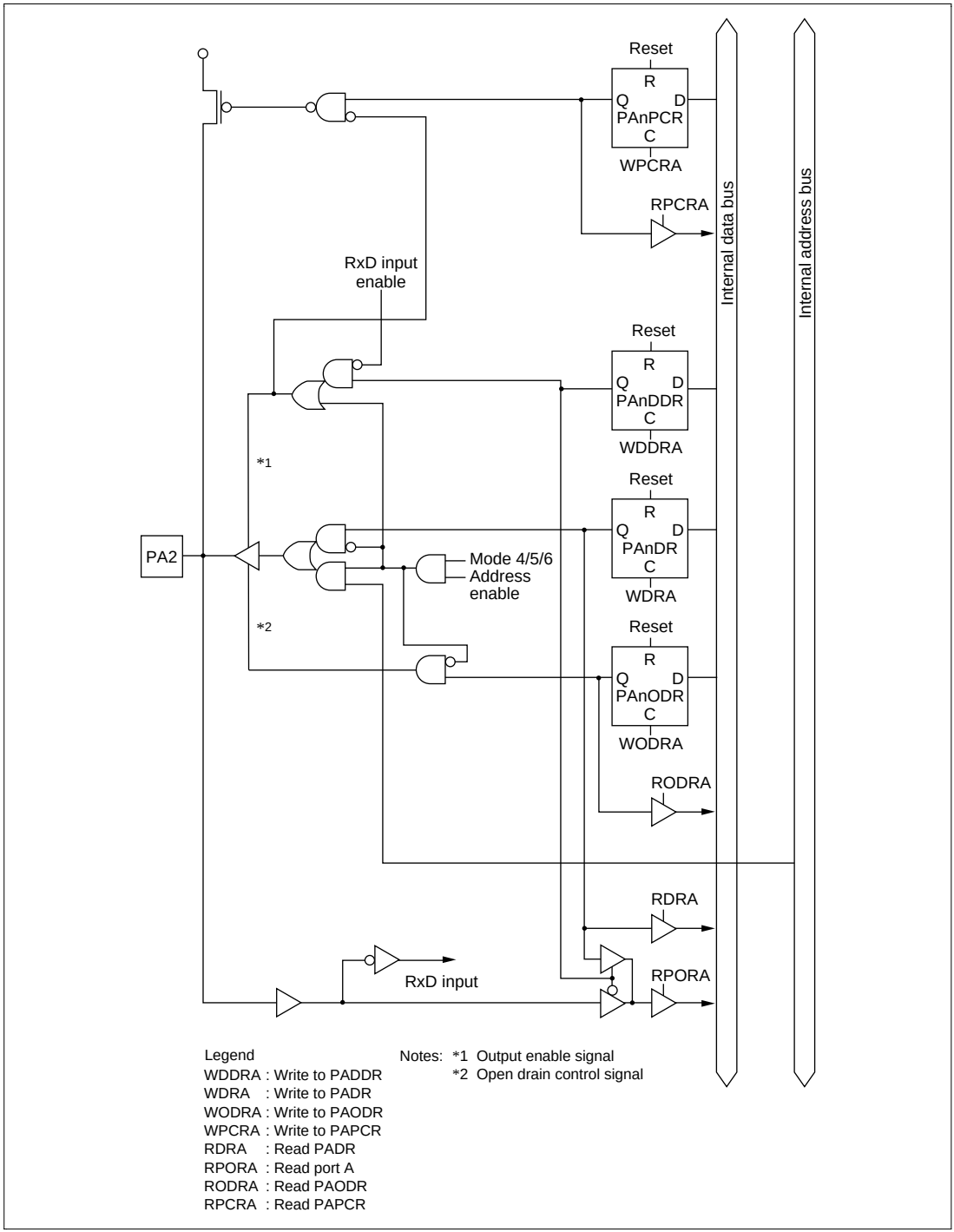
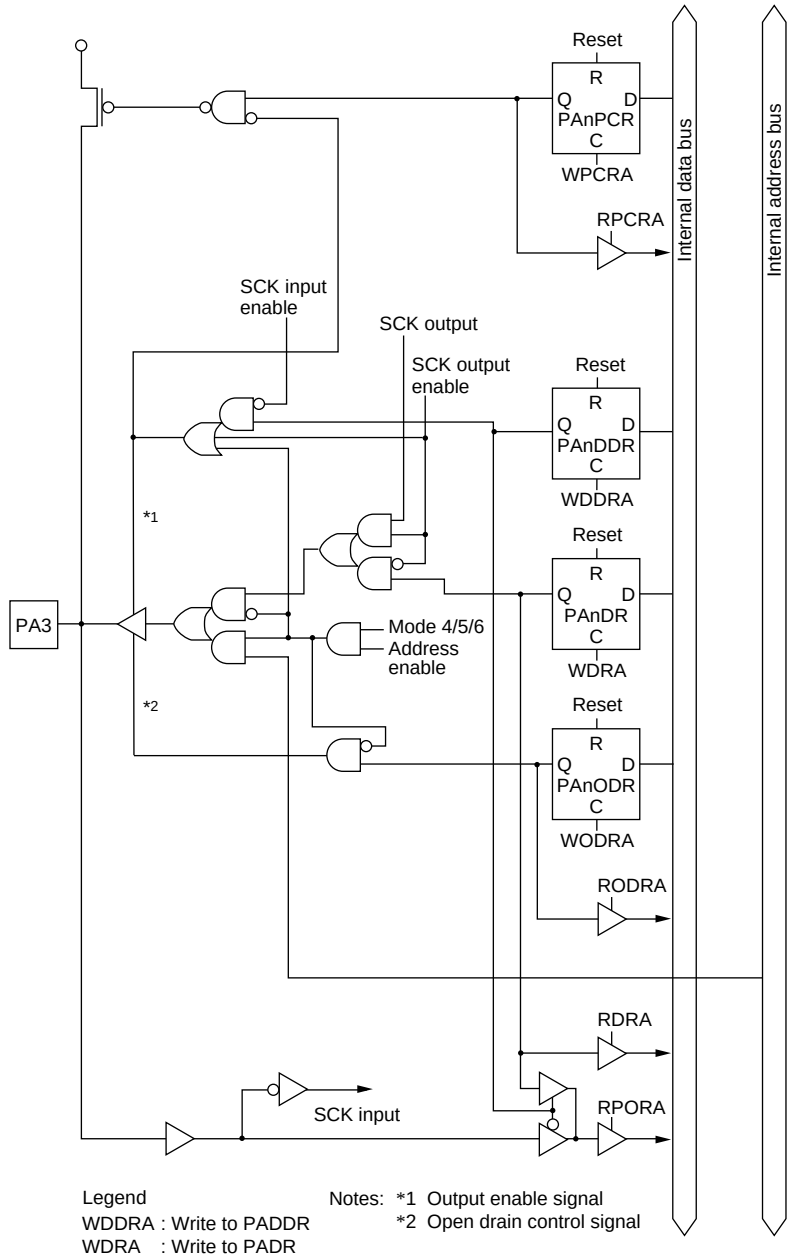


Figure C-6 (c) Port A Block Diagram (Pin PA2)



Legend
WDDRA : Write to PADDR
WDRA : Write to PADR
WODRA : Write to PAODR
WPCRA : Write to PAPCR
RDRA : Read PADR
RPORA : Read port A
RODRA : Read PAODR
RPCRA : Read PAPCR

Notes: *1 Output enable signal
*2 Open drain control signal

Figure C-6 (d) Port A Block Diagram (Pin PA3)

C.7 Port B Block Diagram

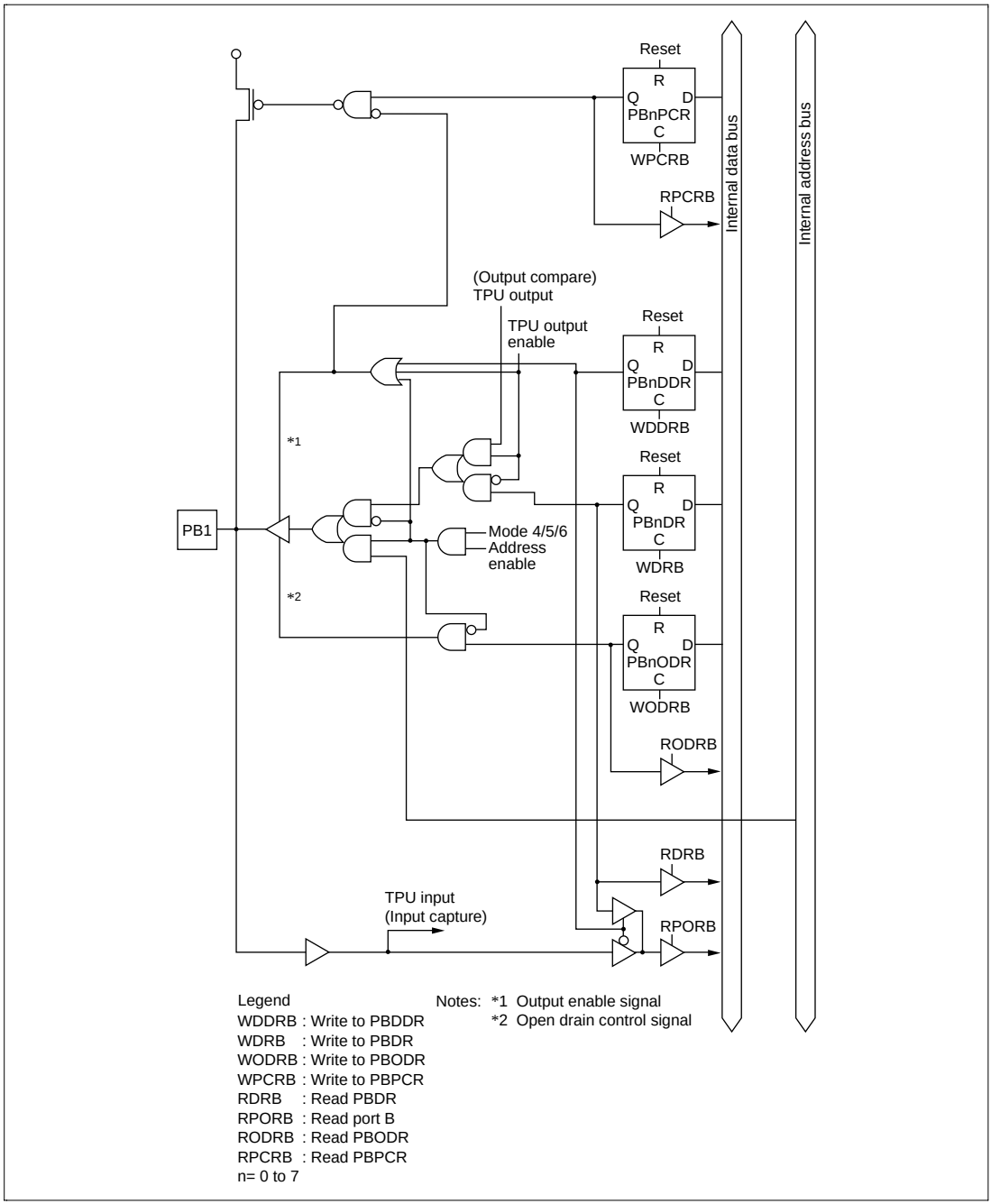


Figure C-7 Port B Block Diagram (Pins PB0 to PB7)

C.8 Port C Block Diagram

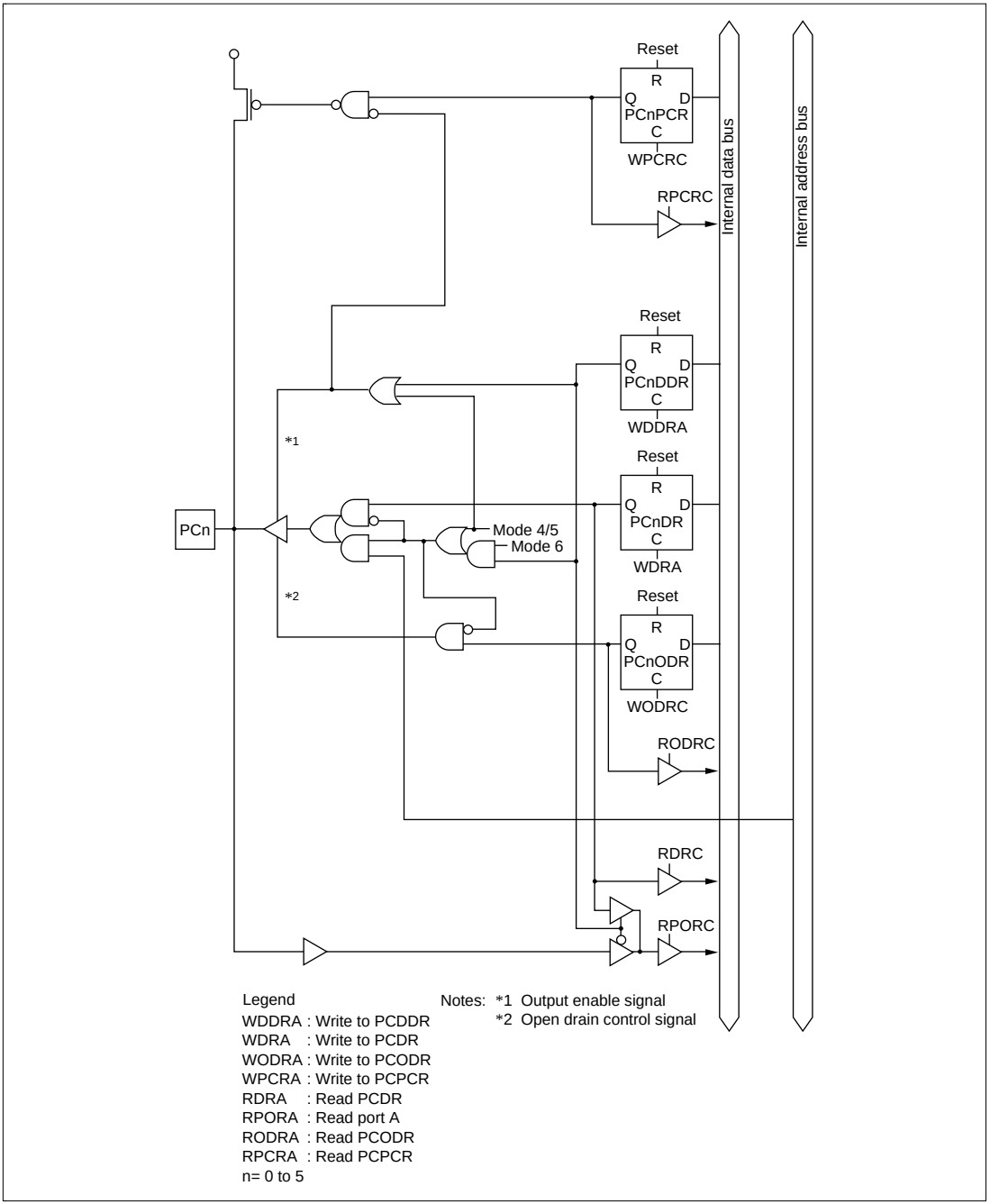


Figure C-8 (a) Port C Block Diagram (Pins PC0 to PC5)

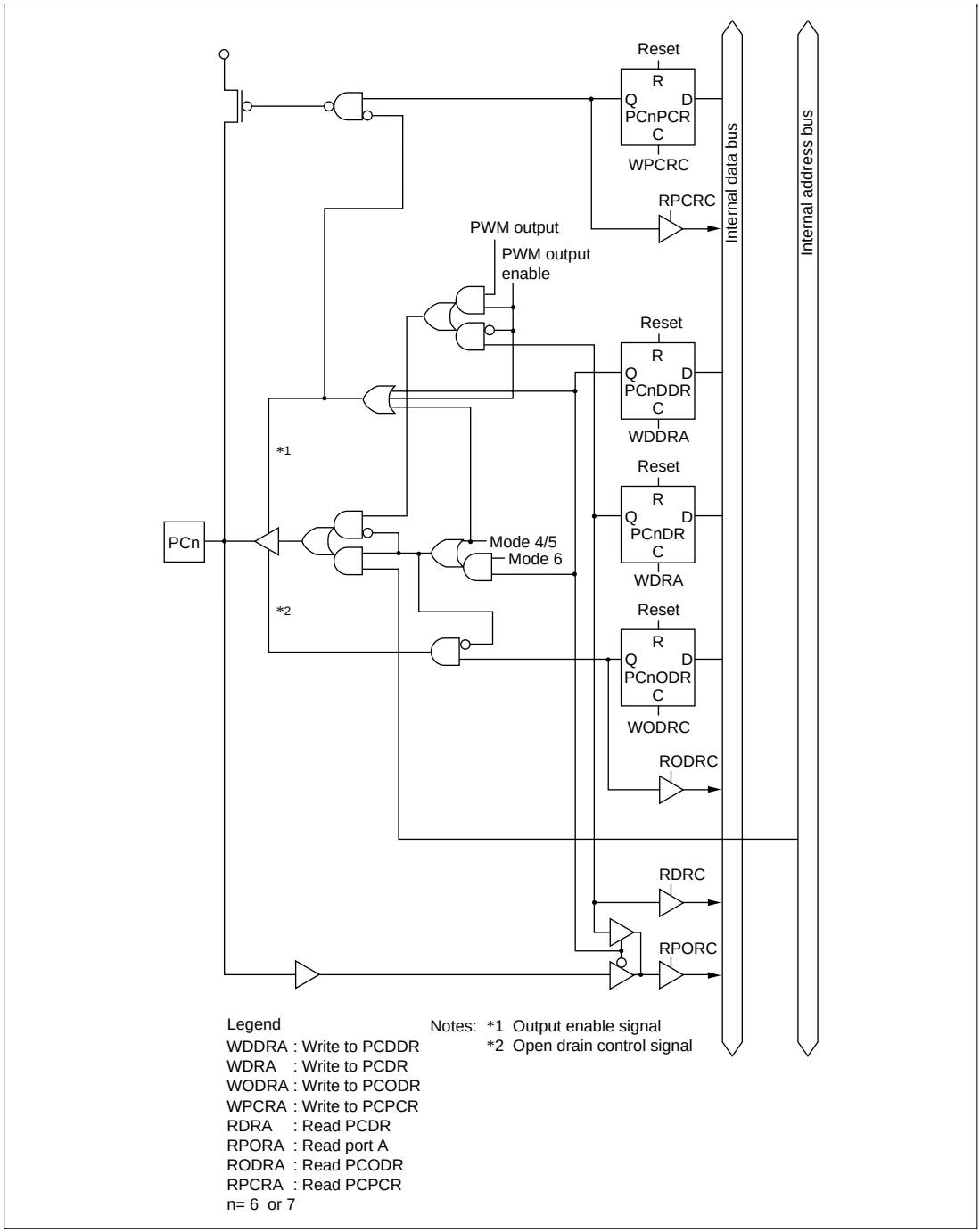


Figure C-8 (b) Port C Block Diagram (Pins PC6 and PC7)

C.9 Port D Block Diagram

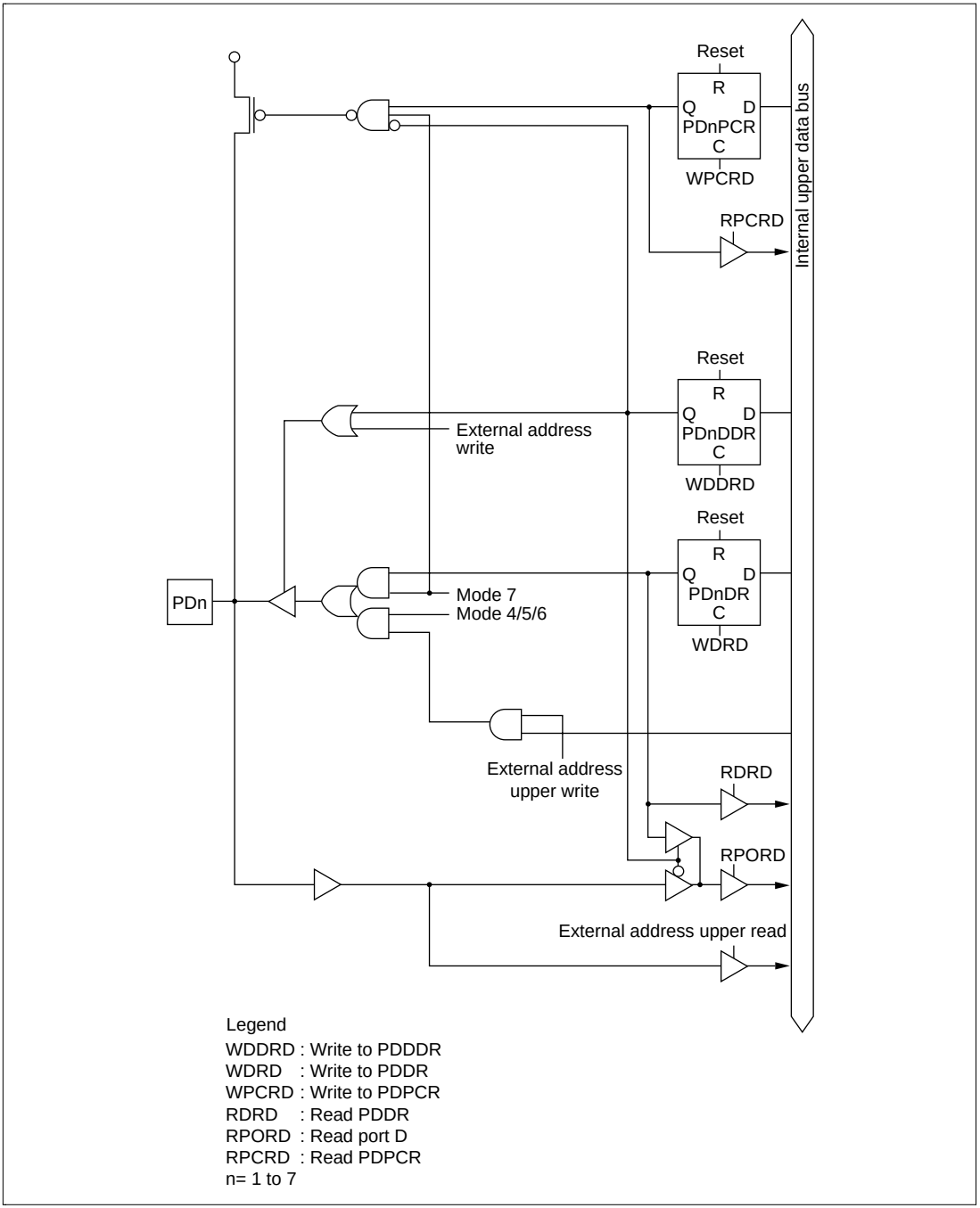


Figure C-9 Port D Block Diagram (Pin PDn)

C.10 Port E Block Diagram

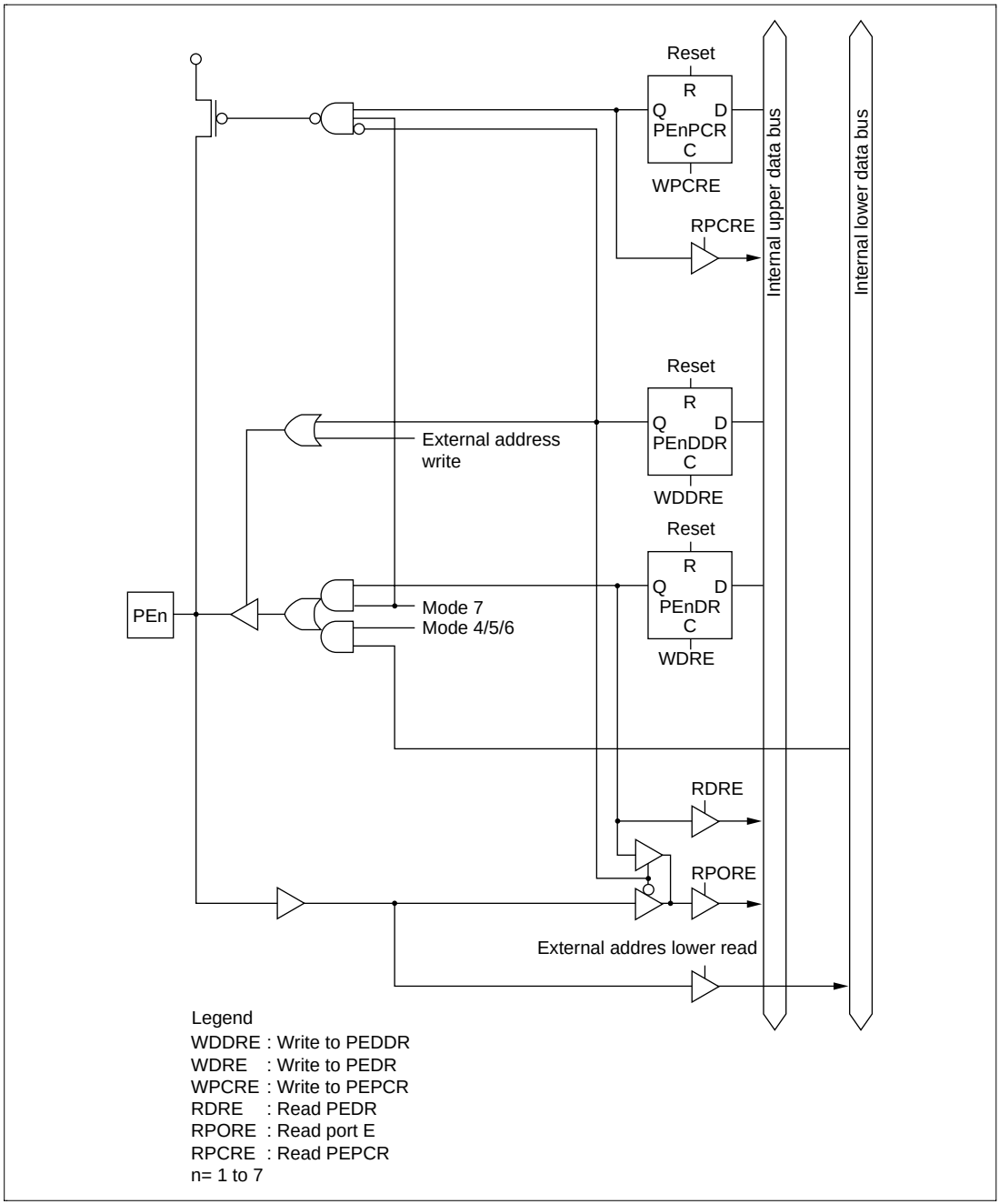


Figure C-10 Port E Block Diagram (Pin PEn)

C.11 Port E Block Diagram

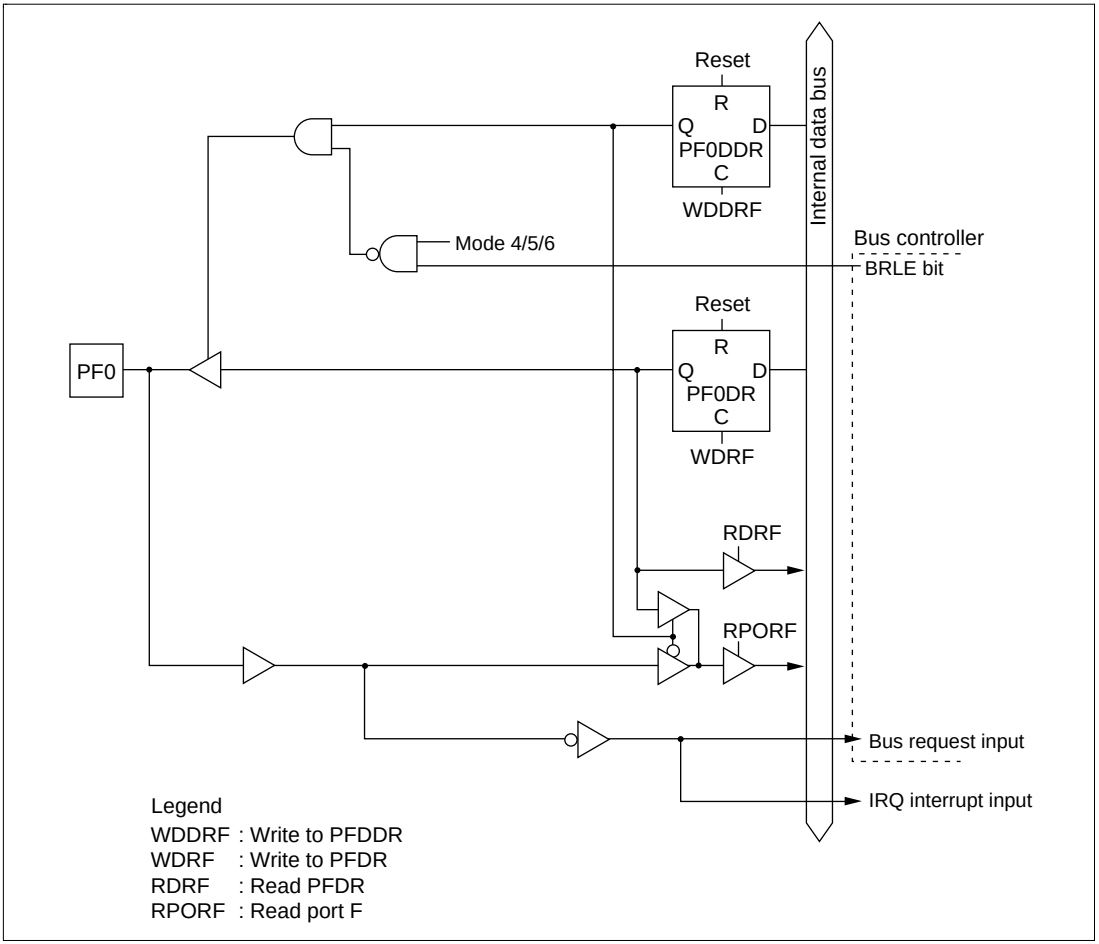


Figure C-11 (a) Port F Block Diagram (Pin PF0)

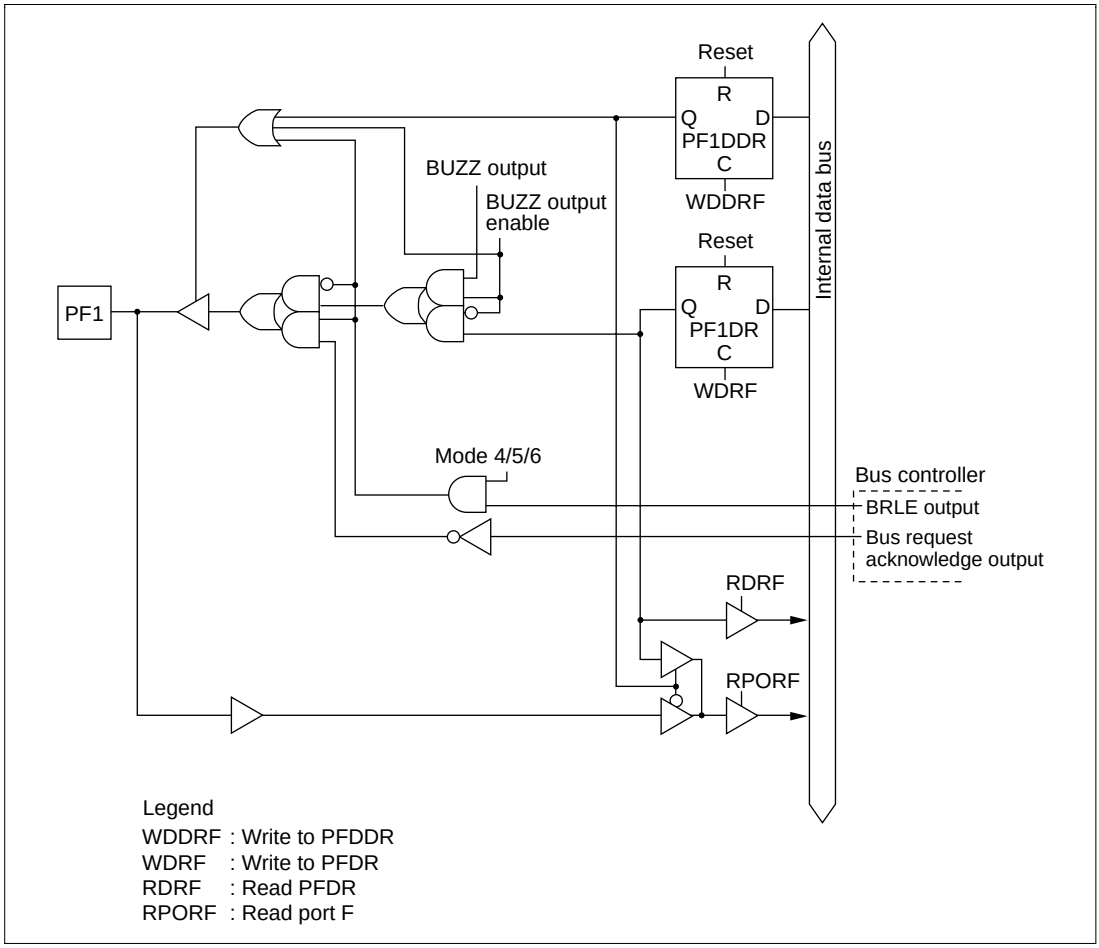


Figure C-11 (b) Port F Block Diagram (Pin PF1)

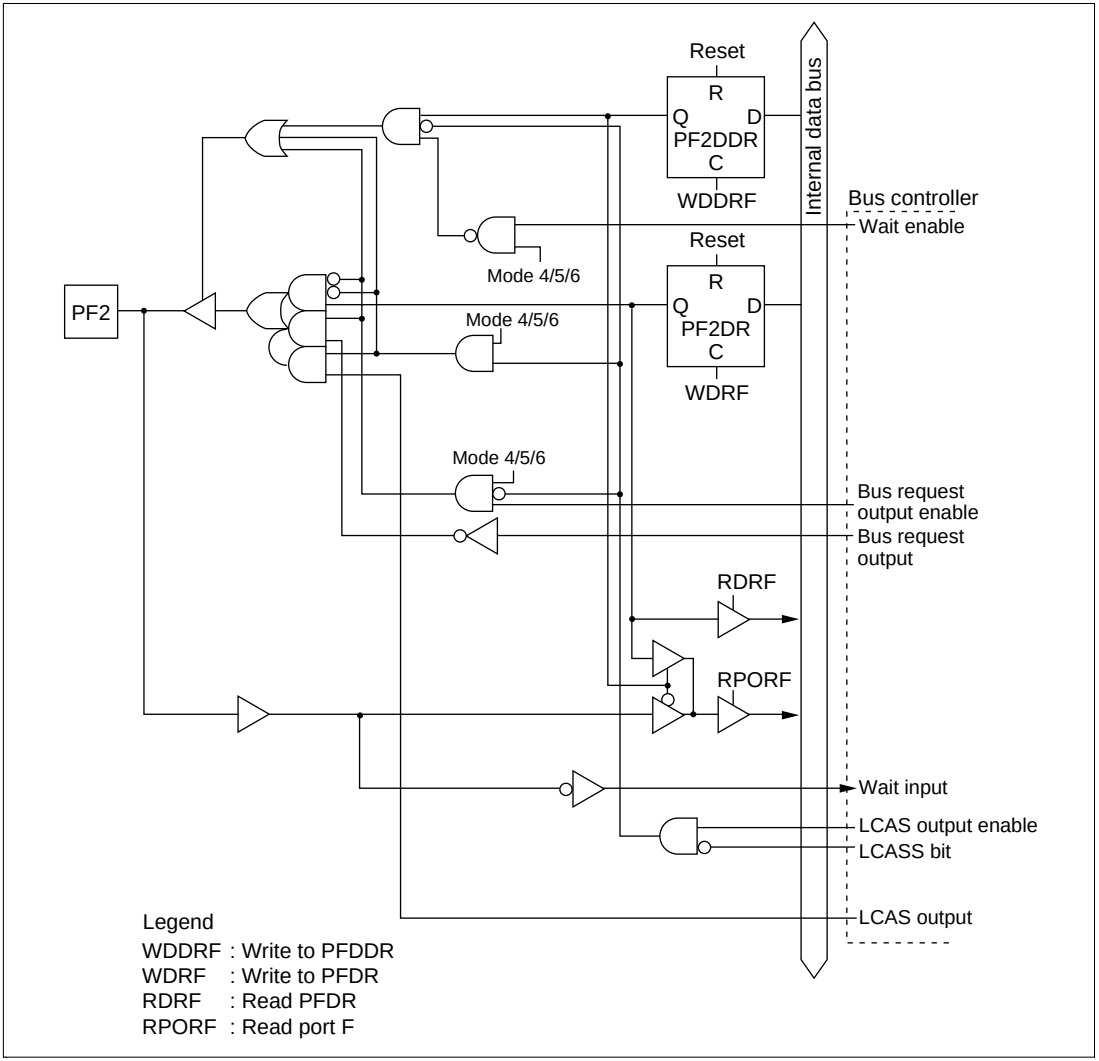


Figure C-11 (c) Port F Block Diagram (Pin PF2)

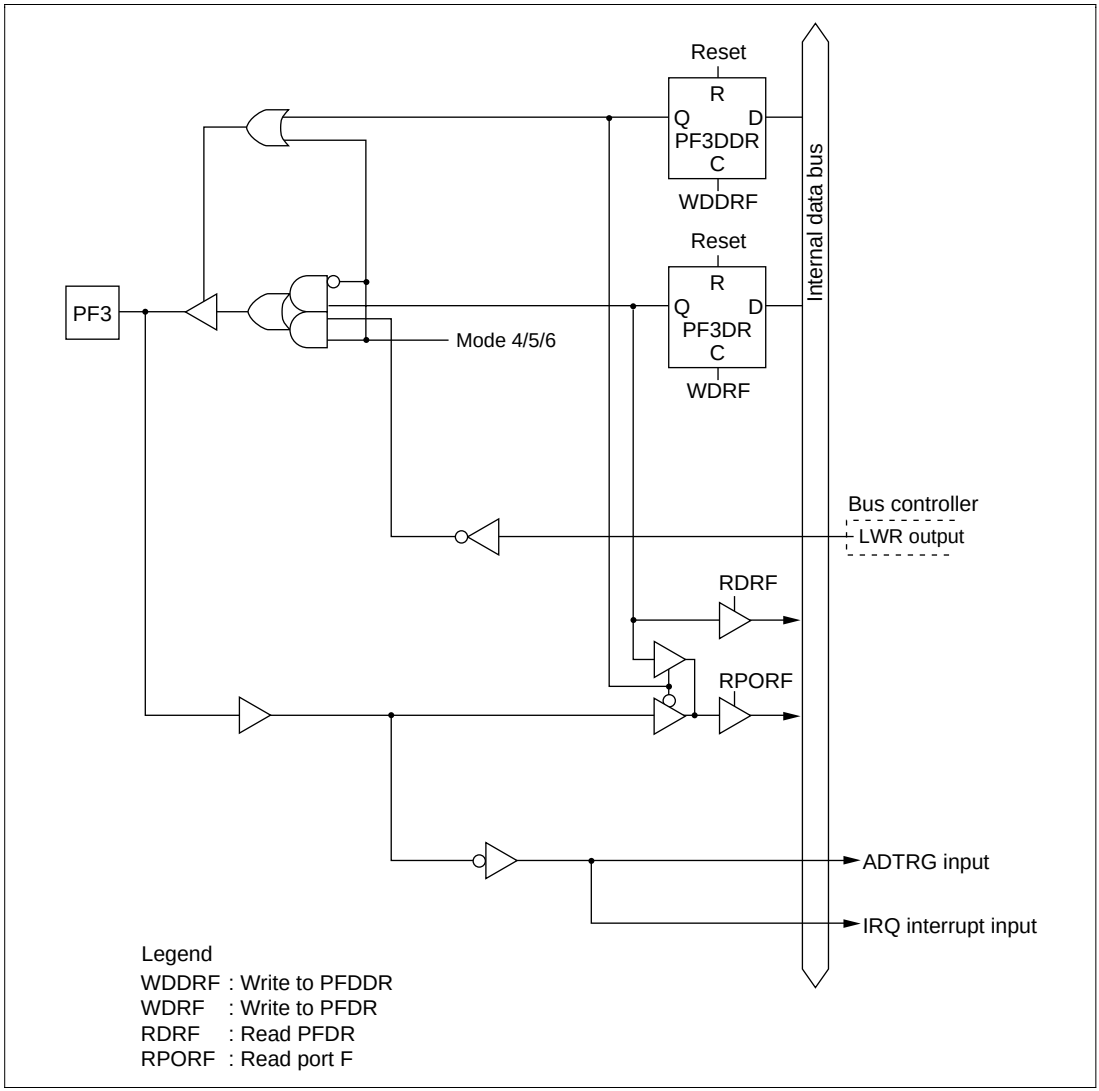


Figure C-11 (d) Port F Block Diagram (Pin PF3)

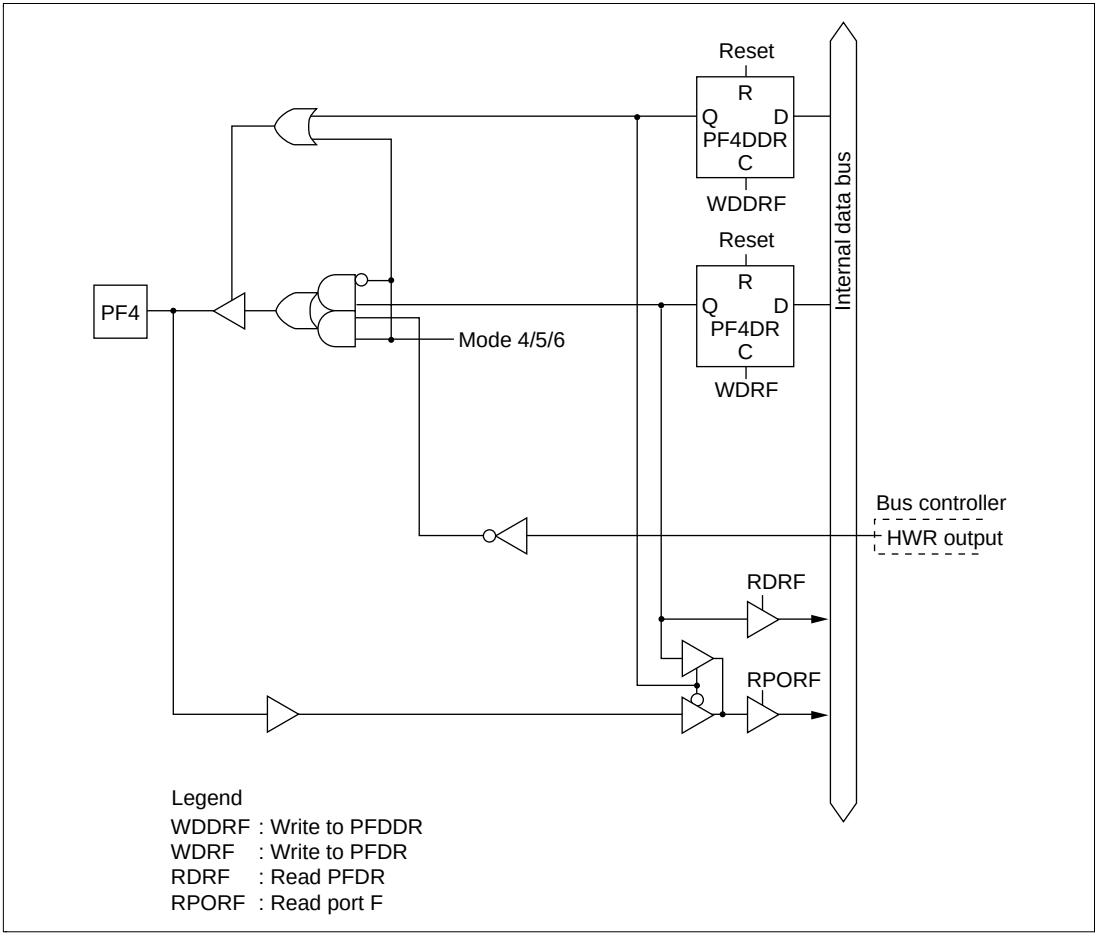


Figure C-11 (e) Port F Block Diagram (Pin PF4)

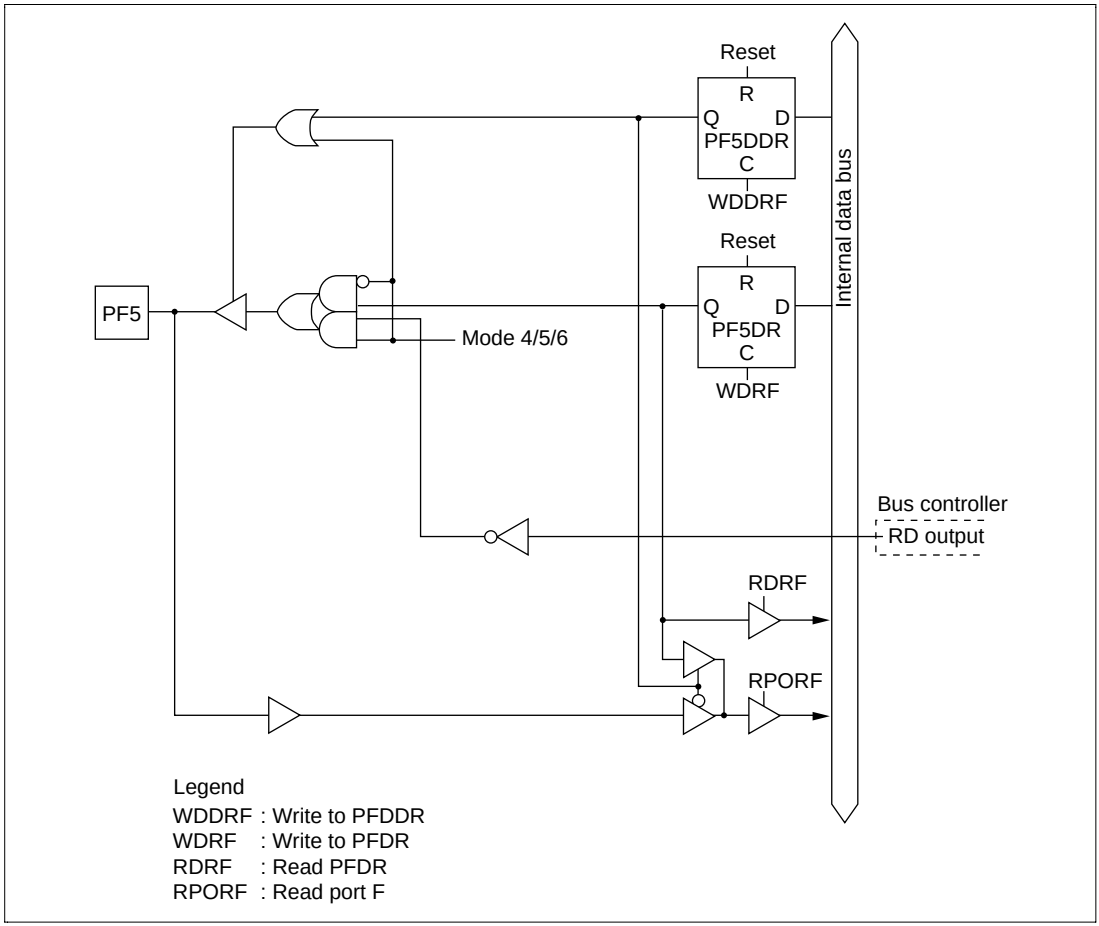


Figure C-11 (f) Port F Block Diagram (Pin PF5)

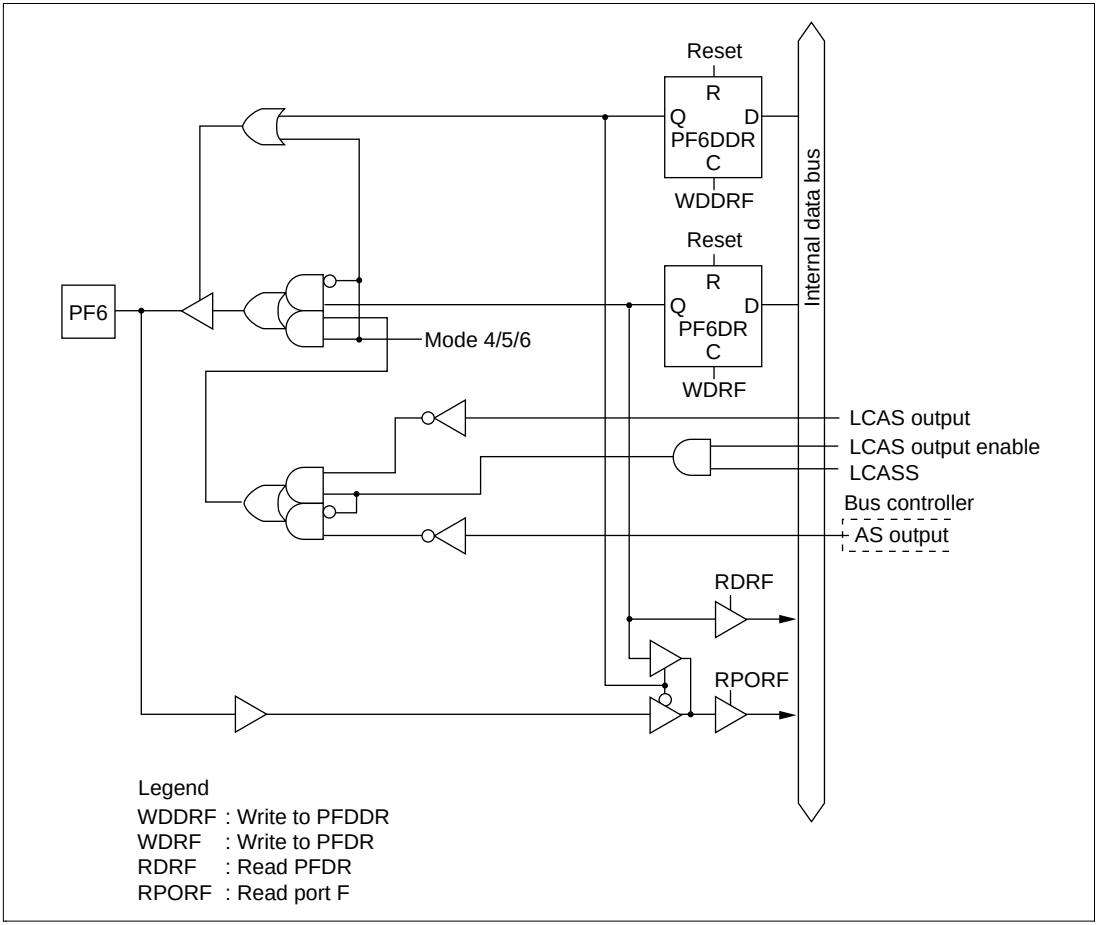


Figure C-11 (g) Port F Block Diagram (Pin PF6)

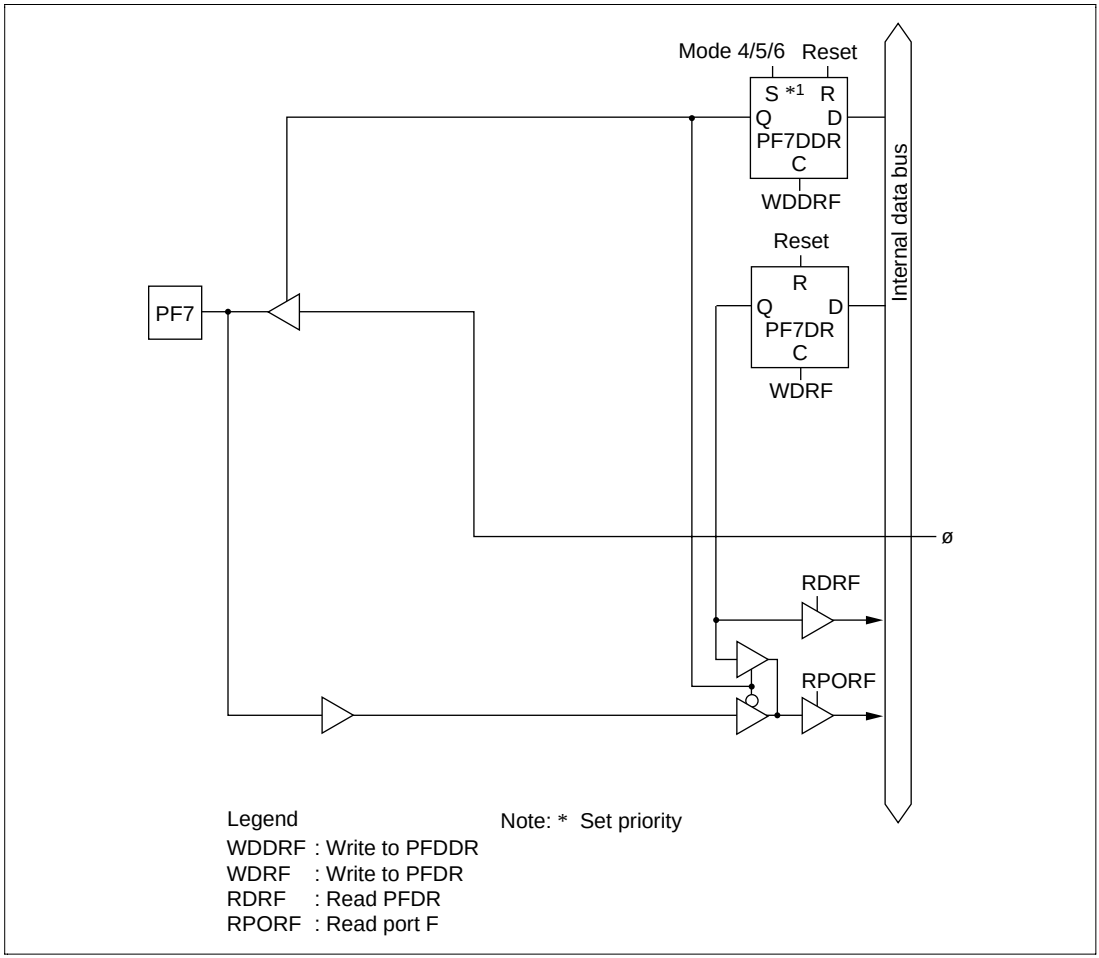


Figure C-11 (h) Port F Block Diagram (Pin PF7)

C.12 Port G Block Diagram

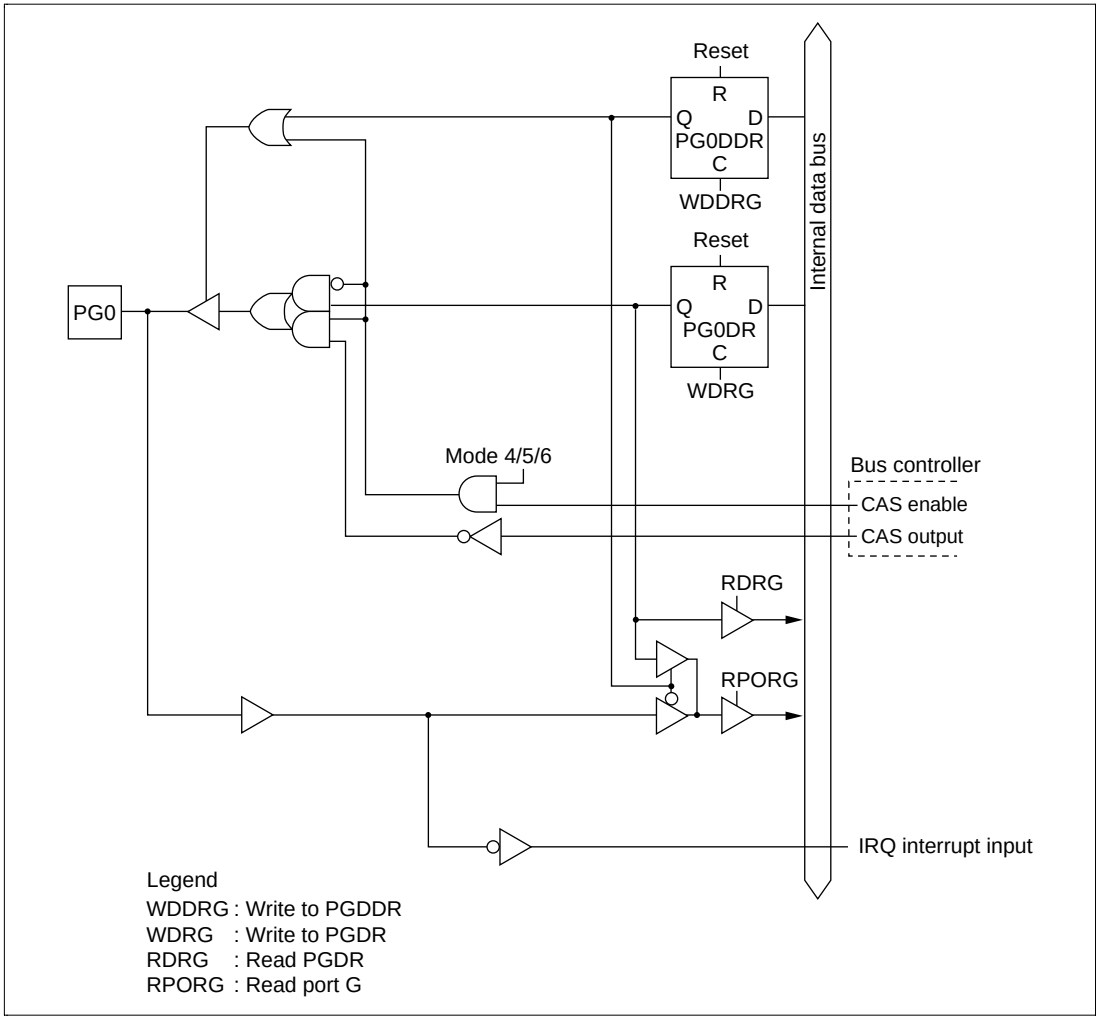


Figure C-12 (a) Port G Block Diagram (Pin PG0)

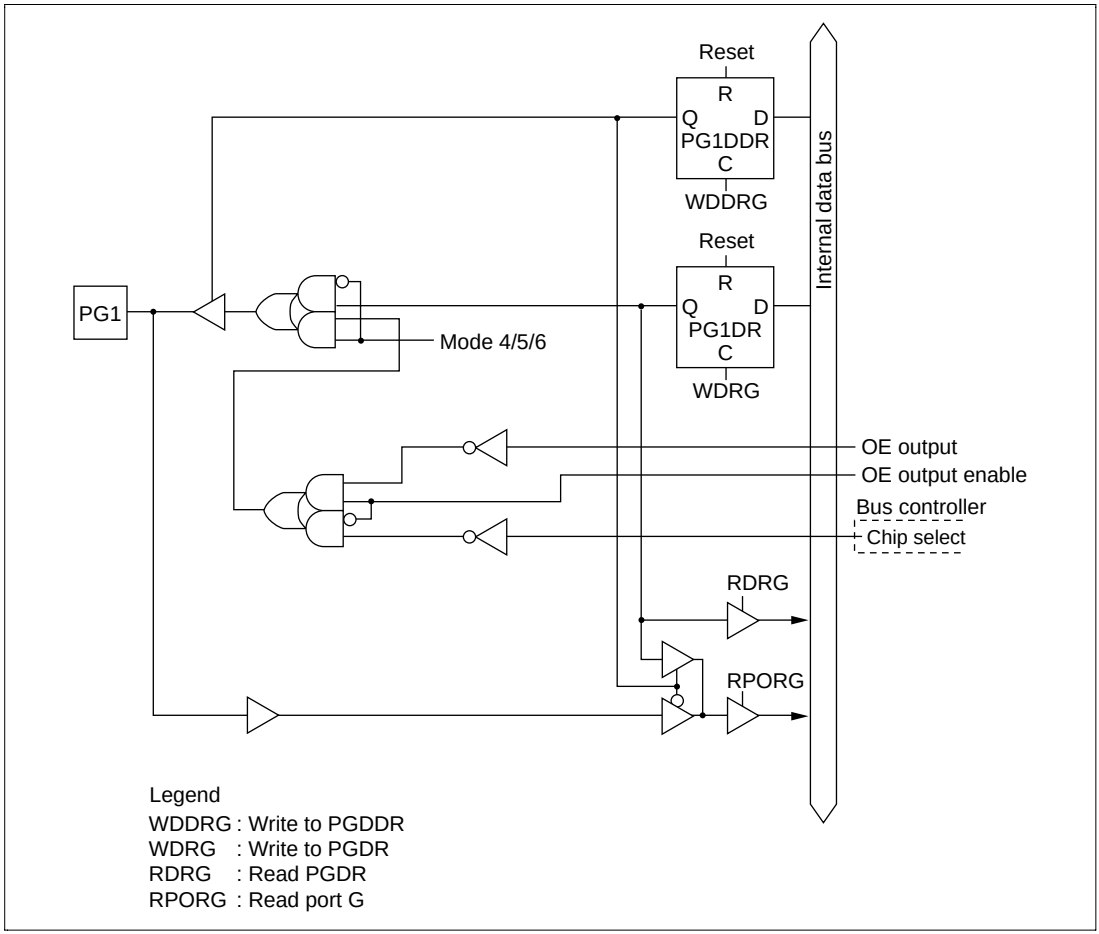


Figure C-12 (b) Port G Block Diagram (Pin PG1)

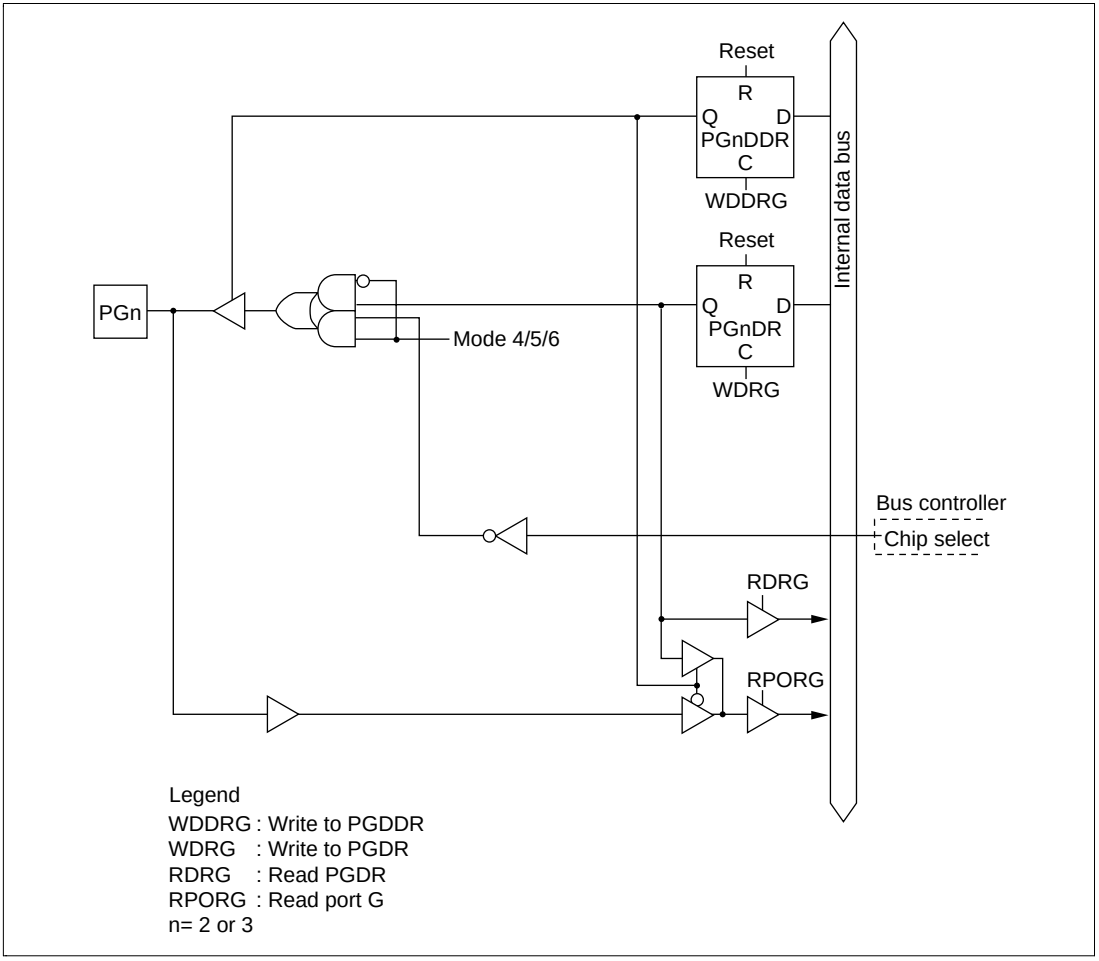


Figure C-12 (c) Port G Block Diagram (Pin PG2 and PG3)

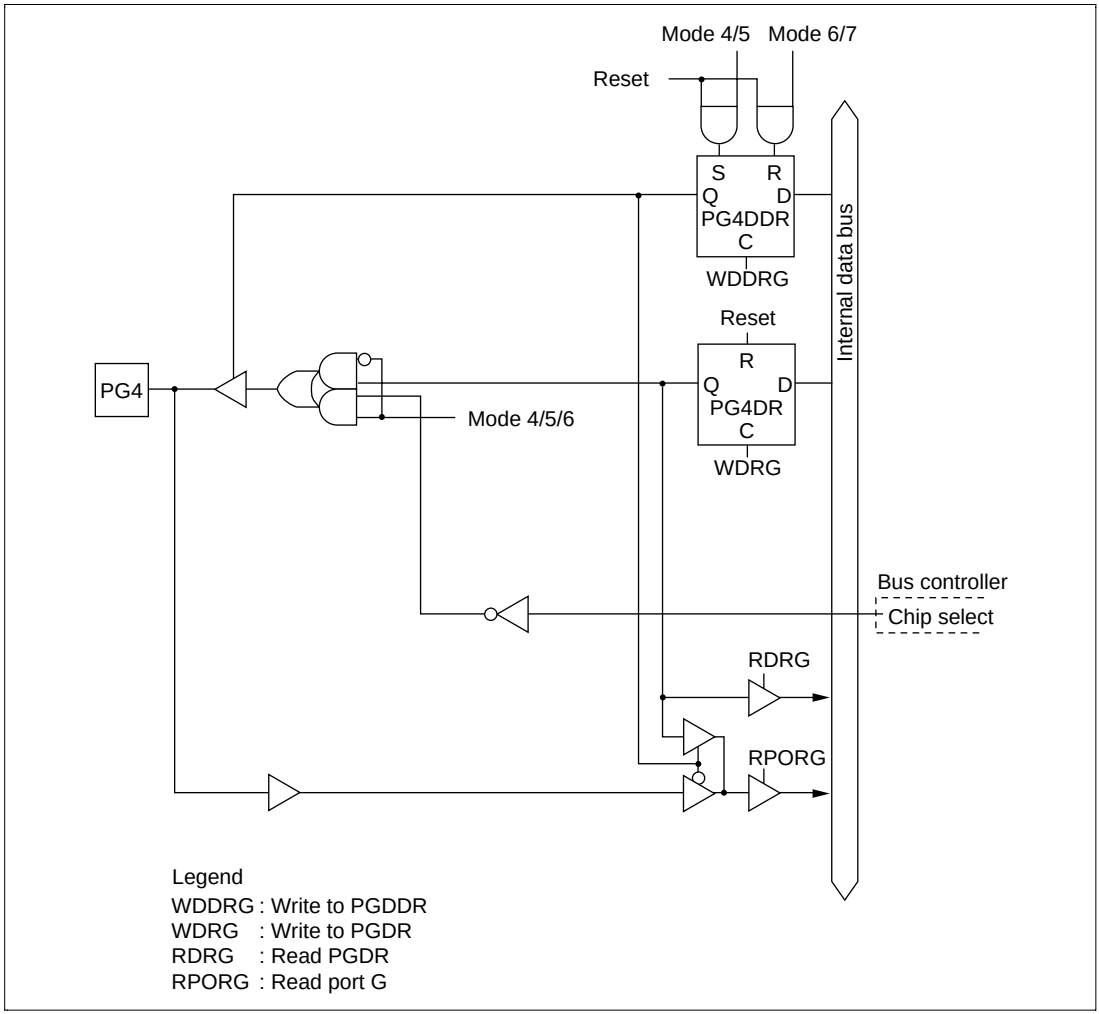


Figure C-12 (d) Port G Block Diagram (Pin PG4)

Appendix D Pin States

D.1 Port States in Each Mode

Table D-1 I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	4 to 7	T	kept	T	kept	kept	I/O port
Port 3	4 to 7	T	kept	T	kept	kept	I/O port
Port 4	4 to 7	T	T	T	T	T	Input port
P73/$\overline{CS7}$ P72/$\overline{CS6}$ P71/$\overline{CS5}$ P70/$\overline{CS4}$	7	T	kept	T	kept	kept	I/O port
	4 to 6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] H	T	[DDR = 0] Input port [DDR = 1] $\overline{CS7}$ to $\overline{CS4}$
Port 9	4 to 7	T	T	T	T	T	Input port
Port A	4, 5	L	kept	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Otherwise] kept	[Address output] T [Otherwise] kept	[Address output] A19 to A17 [Otherwise] I/O port
	6	T	kept	T			
	7	T	kept	T	kept	kept	I/O port
Port B	4, 5	L	kept	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Otherwise] kept	[Address output] T [Otherwise] kept	[Address output] A15 to A8 [Otherwise] I/O port
	6	T	kept	T			
	7	T	kept	T	kept	kept	I/O port

Port Name Pin Name	MCU Operating Mode		Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port C	4, 5		L	kept	T	[OPE = 0] T [OPE = 1] kept	T	A7 to A0
	6		T	kept	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] kept [DDR = 0] kept	T	[DDR = 1] A7 to A0 [DDR = 0] I/O port
	7		T	kept	T	kept	kept	I/O port
Port D	4 to 6		T	T*	T	T	T	Data bus
	7		T	kept	T	kept	kept	I/O port
Port E	4 to 6	8 bit bus	T	kept	T	kept	kept	I/O port
		16 bit bus	T	T*	T	T	T	Data bus
	7		T	kept	T	kept	kept	I/O port
PF7/ø	4 to 6		Clock output	kept	T	[DDR = 0] T [DDR = 1] H	kept	[DDR = 0] T [DDR = 1] Clock output
	7		T	kept	T	[DDR = 0] T [DDR = 1] H	kept	[DDR = 0] T [DDR = 1] Clock output
PF6/$\overline{AS}$ $\overline{LCAS}$	4 to 6		H	H	T	[OPE = 0] T [LCAS output, OPE = 1] LCAS [AS output, OPE = 1] H	T	[LCAS output] $\overline{LCAS}$ [Otherwise] $\overline{AS}$
	7		T	kept	T	kept	kept	I/O port

Port Name Pin Name	MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF5/$\overline{RD}$ PF4/$\overline{HWR}$ PF3/$\overline{LWR}$/ $\overline{ADTRG}$/ $\overline{IRQ3}$	4 to 6	H	H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$
	7	T	kept	T	kept	kept	I/O port
PF2/$\overline{LCAS}$/ $\overline{WAIT}$/ $\overline{BREQO}$	4 to 6	T	[CAS output] H [Otherwise] kept	T	[LCAS output, OPE = 0] T [LCAS output, OPE = 1] $\overline{LCAS}$ [Otherwise] kept	[LCAS output] T [BREQOE = 1] $\overline{BREQO}$ [WAITE = 1] T	[LCAS output] $\overline{LCAS}$ [BREQOE = 1] $\overline{BREQO}$ [WAITE = 1] $\overline{WAIT}$
	7	T	kept	T	kept	kept	I/O port
PF1/$\overline{BACK}$ BUZZ	4 to 6	T	kept	T	[BRLE = 0, BUZZE = 0] I/O port [BRLE = 0, BUZZE = 1] H [BRLE = 1] H	[BRLE = 0, BUZZE = 0] I/O port [BRLE = 0, BUZZE = 1] H [BRLE = 1] L	[BRLE = 0, BUZZE = 0] I/O port [BRLE = 0, BUZZE = 1] BUZZ [BRLE = 1] $\overline{BACK}$
	7	T	kept	T	kept	kept	I/O port
PF0/$\overline{BREQ}$/ $\overline{IRQ2}$	4 to 6	T	kept	T	[BRLE = 0] kept [BRLE = 1] T	T	[BRLE = 0] I/O port [BRLE = 1] $\overline{BREQO}$
	7	T	kept	T	kept	kept	I/O port
PG4/$\overline{CS0}$	4, 5	H	kept	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] H [DDR = 0] T	T	[DDR = 0] Input port [DDR = 1] $\overline{CS0}$
	6	T					
	7	T	kept	T	kept	kept	I/O port

Port Name Pin Name	MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PG3/$\overline{CS1}$ PG2/$\overline{CS2}$	4 to 6	T	kept	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] H [DDR = 0] T	T	[DDR = 0] Input port [DDR = 1] $\overline{CS2}$ to $\overline{CS1}$
	7	T	kept	T	kept	kept	I/O port
PG1/$\overline{CS3}$/ $\overline{OE}$/$\overline{IRQ7}$	4 to 6	T	kept	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] H [DDR = 0] T	T	[DDR = 0] Input port [OE = 0, DDR = 1] $\overline{CS3}$ [OE = 1, DDR = 1] $\overline{OE}$
	7	T	kept	T	kept	kept	I/O port
PG0/$\overline{CAS}$/ $\overline{IRQ6}$	4 to 6	T	kept	T	[DRAME = 0] kept [DRAME = 1, OPE = 1] $\overline{CAS}$ [DRAME = 1, OPE = 1] T	T	[DRAME = 0] I/O port [DRAME = 1] $\overline{CAS}$
	7	T	kept	T	kept	kept	I/O port

Legend:

H : High level
L : Low level
T : High impedance
kept : Input port becomes high-impedance, output port retains state
DDR : Data direction register
OPE : Output port enable
WAITE : Wait input enable
BRLE : Bus release enable
BREQOE : BREQO pin enable
DRAME : DRAM space setting
LCASE : DRAM space setting, CW2 = LCASS = 0

Note: * Indicates the state after completion of the executing bus cycle.

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{RES}$ signal low at least 10 states before the $\overline{STBY}$ signal goes low, as shown below. $\overline{RES}$ must remain low until $\overline{STBY}$ signal goes low (delay from $\overline{STBY}$ low to $\overline{RES}$ high: 0 ns or more).

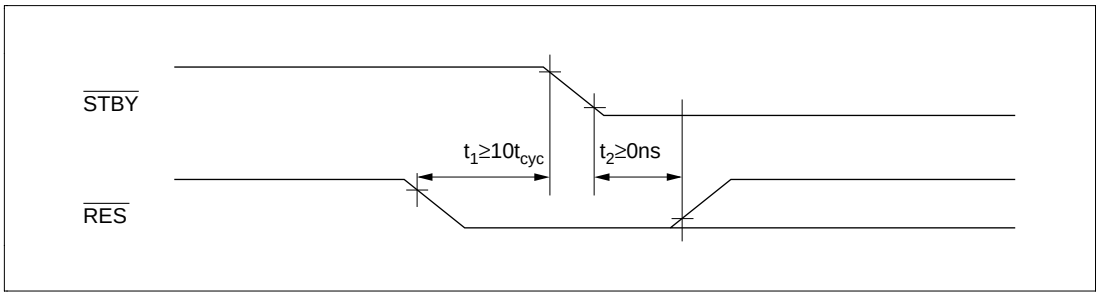


Figure E-1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained, $\overline{RES}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low and the NMI signal high approximately 100 ns or more before $\overline{STBY}$ goes high to execute a power-on reset.

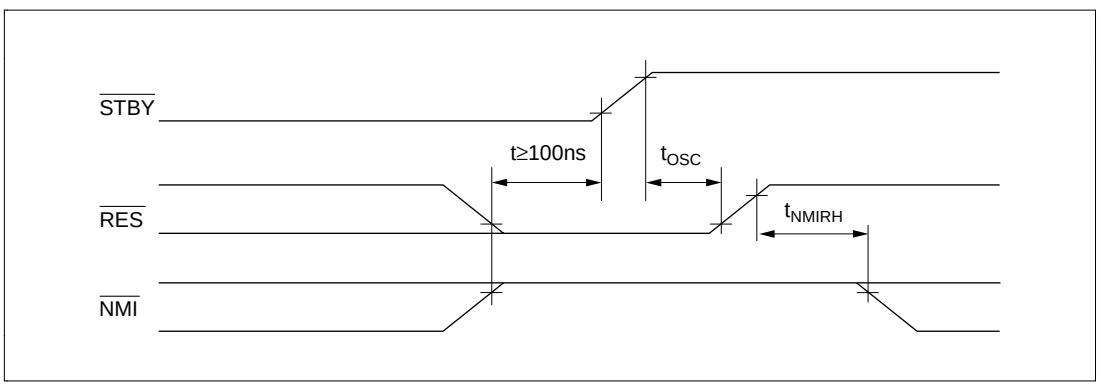


Figure E-2 Timing of Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

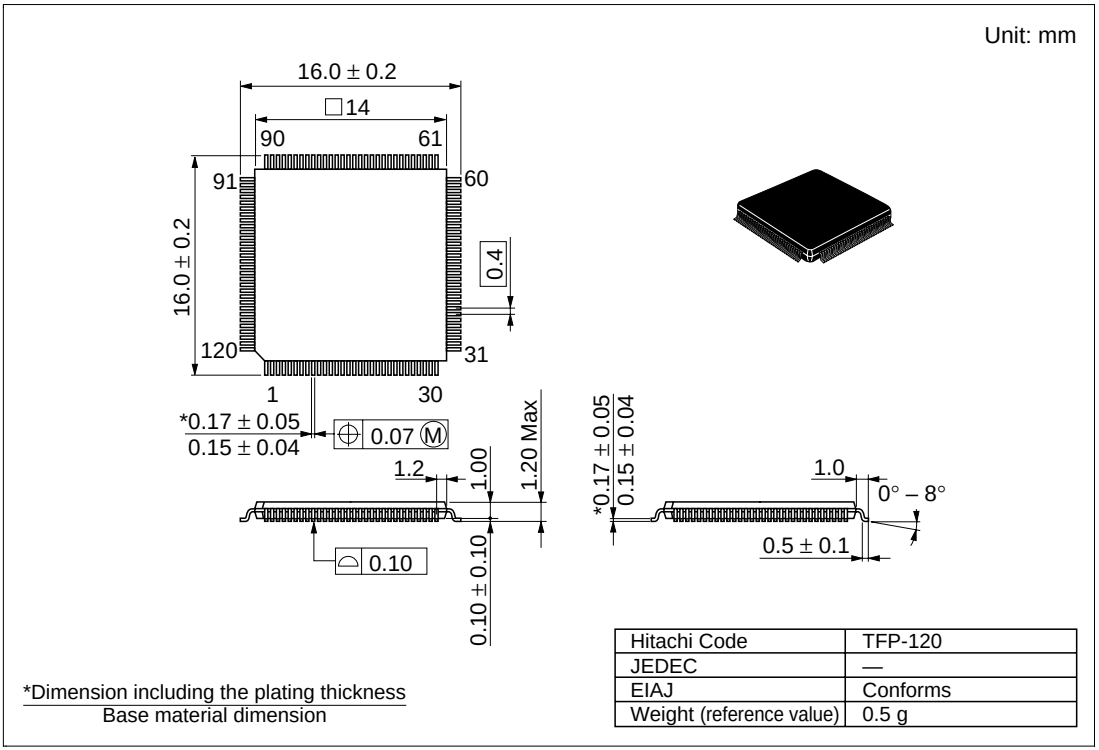
Table F.1 H8S/2633 Series Product Code Lineup

Product Type		Product Code	Mark Code	Package (Hitachi Package Code)
H8S/2633	F-ZTAT™	HD64F2633	HD64F2633F	128-pin QFP (FP-128)
			HD64F2633TE	120-pin TQFP (TFP-120)
	Mask ROM*	HD6432633	HD6432633F	128-pin QFP (FP-128)
			HD6432633TE	120-pin TQFP (TFP-120)
H8S/2632		HD6432632	HD6432632F	128-pin QFP (FP-128)
			HD6432632TE	120-pin TQFP (TFP-120)
H8S/2631		HD6432631	HD6432631F	128-pin QFP (FP-128)
			HD6432631TE	120-pin TQFP (TFP-120)

Note: * In the planning stage.

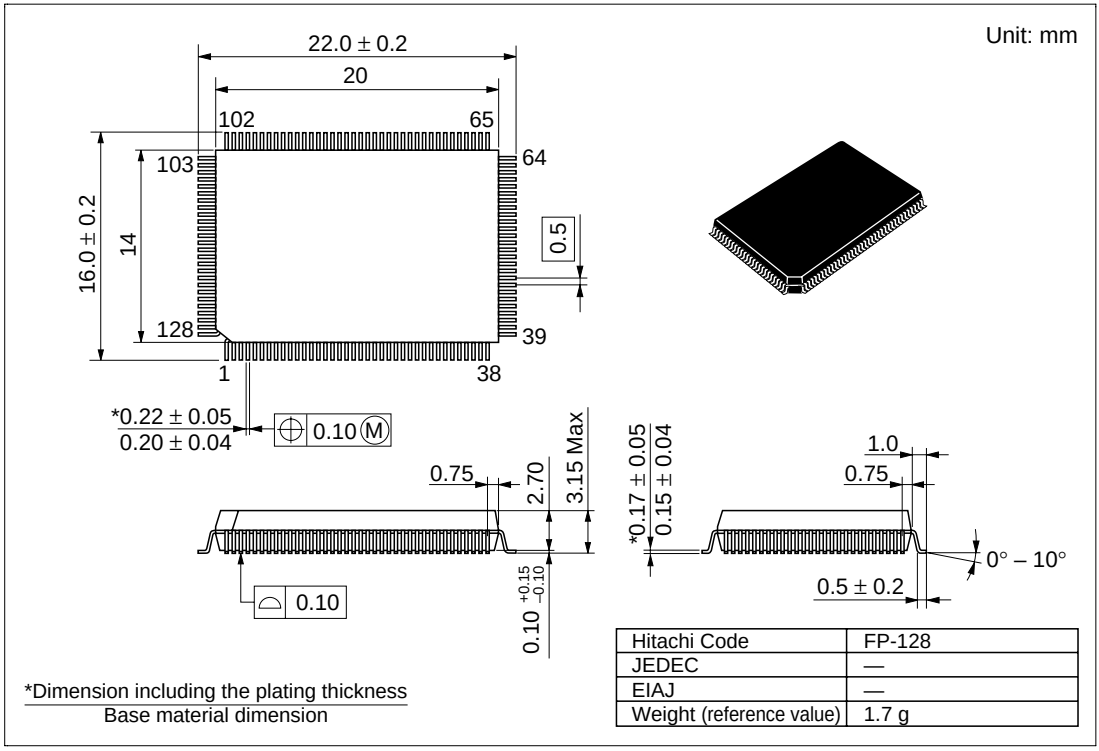
Appendix G Package Dimensions

Figures G-1 and G-2 show the TFP-120 and FP-128 package dimensions of the H8S/2633 Series.



Hitachi Code	TFP-120
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.5 g

Figure G-1 TFP-120 Package Dimensions



Hitachi Code	FP-128
JEDEC	—
EIAJ	—
Weight (reference value)	1.7 g

Figure G-2 FP-128 Package Dimensions

H8S/2633 Series, H8S/2633 F-ZTAT™ Hardware Manual

Publication Date: 1st Edition, December 1998
3rd Edition, March 2001
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.
Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.