

Stellaris® LM3S6432 Microcontroller

DATA SHEET

Copyright

Copyright © 2007-2014 Texas Instruments Incorporated All rights reserved. Stellaris and StellarisWare® are registered trademarks of Texas Instruments Incorporated. ARM and Thumb are registered trademarks and Cortex is a trademark of ARM Limited. Other names and brands may be claimed as the property of others.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

A Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

Texas Instruments Incorporated 108 Wild Basin, Suite 350 Austin, TX 78746 http://www.ti.com/stellaris







http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm

Table of Contents

Revision H	listory	24
About This	Document	30
About This M	Manual	30
Related Docu	uments	30
Documentation	ion Conventions	31
1	Architectural Overview	33
1.1	Product Features	
1.2	Target Applications	
1.3	High-Level Block Diagram	
1.4	Functional Overview	
1.4.1		
1.4.2	2 Motor Control Peripherals	44
1.4.3	·	
1.4.4	4 Serial Communications Peripherals	45
1.4.5	·	
1.4.6	6 Memory Peripherals	47
1.4.7		
1.4.8	B Hardware Details	48
2	The Cortex-M3 Processor	49
_ 2.1	Block Diagram	
2.2	Overview	
2.2.1		
2.2.2	·	
2.2.3		
2.2.4	, ,	
2.3	Programming Model	
2.3.1		
2.3.2		
2.3.3		
2.3.4		
2.3.5	· · · · · · · · · · · · · · · · · · ·	
2.3.6	3 Data Types	68
2.4	Memory Model	68
2.4.1	1 Memory Regions, Types and Attributes	69
2.4.2	* * *	
2.4.3	Behavior of Memory Accesses	70
2.4.4	Software Ordering of Memory Accesses	71
2.4.5	5 Bit-Banding	72
2.4.6	6 Data Storage	74
2.4.7	7 Synchronization Primitives	75
2.5	Exception Model	76
2.5.1	1 Exception States	77
2.5.2	2 Exception Types	77
2.5.3	3 Exception Handlers	80

2.5.4	Vector Table	80
2.5.5	Exception Priorities	81
2.5.6	Interrupt Priority Grouping	82
2.5.7	Exception Entry and Return	82
2.6	Fault Handling	84
2.6.1	Fault Types	85
2.6.2	Fault Escalation and Hard Faults	85
2.6.3	Fault Status Registers and Fault Address Registers	86
2.6.4	Lockup	
2.7	Power Management	86
2.7.1	Entering Sleep Modes	87
2.7.2	Wake Up from Sleep Mode	87
2.8	Instruction Set Summary	88
3	Cortex-M3 Peripherals	91
3.1	Functional Description	
3.1.1	System Timer (SysTick)	
3.1.2	Nested Vectored Interrupt Controller (NVIC)	
3.1.3	System Control Block (SCB)	
3.1.4	Memory Protection Unit (MPU)	
3.2	Register Map	
3.3	System Timer (SysTick) Register Descriptions	
3.4	NVIC Register Descriptions	
3.5	System Control Block (SCB) Register Descriptions	
3.6	Memory Protection Unit (MPU) Register Descriptions	
4	JTAG Interface	
4.1	Block Diagram	
4.2	Signal Description	
4.3	Functional Description	
4.3.1	JTAG Interface Pins	
4.3.2	JTAG TAP Controller	
4.3.3	Shift Registers	
4.3.4	Operational Considerations	
4.4	Initialization and Configuration	
4.5	Register Descriptions	
4.5.1	Instruction Register (IR)	
4.5.2	Data Registers	
5	System Control	168
5.1	Signal Description	
5.2	·	
-	Functional Description	168
5.2.1	Functional Description	
	Device Identification	169
5.2.2	Device Identification	169 169
5.2.2 5.2.3	Device Identification Reset Control Power Control	169 169 173
5.2.2 5.2.3 5.2.4	Device Identification Reset Control Power Control Clock Control	169 169 173
5.2.2 5.2.3 5.2.4 5.2.5	Device Identification Reset Control Power Control Clock Control System Control	
5.2.2 5.2.3 5.2.4	Device Identification Reset Control Power Control Clock Control	

6	Internal Memory	231
6.1	Block Diagram	231
6.2	Functional Description	231
6.2.1	SRAM Memory	231
6.2.2	Flash Memory	232
6.3	Flash Memory Initialization and Configuration	234
6.3.1	Flash Programming	234
6.3.2	Nonvolatile Register Programming	235
6.4	Register Map	236
6.5	Flash Register Descriptions (Flash Control Offset)	237
6.6	Flash Register Descriptions (System Control Offset)	245
7	General-Purpose Input/Outputs (GPIOs)	258
7.1	Signal Description	
7.2	Functional Description	263
7.2.1	Data Control	264
7.2.2	Interrupt Control	265
7.2.3	Mode Control	266
7.2.4	Commit Control	266
7.2.5	Pad Control	266
7.2.6	Identification	267
7.3	Initialization and Configuration	267
7.4	Register Map	268
7.5	Register Descriptions	270
8	General-Purpose Timers	305
8.1	Block Diagram	306
8.2	Signal Description	306
8.3	Functional Description	307
8.3.1	GPTM Reset Conditions	307
8.3.2	32-Bit Timer Operating Modes	307
8.3.3	16-Bit Timer Operating Modes	
8.4	Initialization and Configuration	
8.4.1	32-Bit One-Shot/Periodic Timer Mode	
8.4.2	,	
8.4.3		
8.4.4	16-Bit Input Edge Count Mode	
8.4.5	16-Bit Input Edge Timing Mode	
8.4.6	16-Bit PWM Mode	
8.5	Register Map	
8.6	Register Descriptions	
9	Watchdog Timer	
9.1	Block Diagram	
9.2	Functional Description	
9.3	Initialization and Configuration	
9.4	Register Map	
9.5	Register Descriptions	344
10	Analog-to-Digital Converter (ADC)	365
10.1	Block Diagram	365

10.2	Signal Description	
10.3	Functional Description	367
10.3.1	Sample Sequencers	367
10.3.2	Module Control	367
10.3.3	Hardware Sample Averaging Circuit	368
10.3.4	Analog-to-Digital Converter	368
10.3.5	Differential Sampling	369
10.3.6	Test Modes	371
10.3.7	Internal Temperature Sensor	371
10.4	Initialization and Configuration	372
10.4.1	Module Initialization	
10.4.2	Sample Sequencer Configuration	
10.5	Register Map	
10.6	Register Descriptions	
11	Universal Asynchronous Receivers/Transmitters (UARTs)	
11.1	Block Diagram	
11.2	Signal Description	
11.3	Functional Description	
11.3.1	Transmit/Receive Logic	
11.3.1	Baud-Rate Generation	
11.3.2	Data Transmission	
11.3.4	Serial IR (SIR)	
11.3.5	FIFO Operation	
11.3.6	Interrupts	
11.3.7	Loopback Operation	
11.3.7	IrDA SIR block	
11.4	Initialization and Configuration	
11.5	Register Map	
11.6	Register Descriptions	
12	Synchronous Serial Interface (SSI)	
12.1	Block Diagram	
12.2	Signal Description	
12.3	Functional Description	
12.3.1	Bit Rate Generation	
	FIFO Operation	
	Interrupts	
	Frame Formats	
12.4	Initialization and Configuration	
12.5	Register Map	
12.6	Register Descriptions	
13	Inter-Integrated Circuit (I ² C) Interface	481
13.1	Block Diagram	
13.2	Signal Description	482
13.3	Functional Description	482
13.3.1	I ² C Bus Functional Overview	483
13.3.2	Available Speed Modes	485
	Interrupts	
	Loopback Operation	

13.3.5	Command Sequence Flow Charts	486
13.4	Initialization and Configuration	494
13.5	Register Map	495
13.6	Register Descriptions (I ² C Master)	496
13.7	Register Descriptions (I ² C Slave)	509
14	Ethernet Controller	518
14.1	Block Diagram	
14.2	Signal Description	
14.3	Functional Description	
14.3.1	MAC Operation	
	Internal MII Operation	
	PHY Operation	
14.3.4	Interrupts	
14.4	Initialization and Configuration	
14.4.1	Hardware Configuration	
14.4.2	Software Configuration	
14.5	Ethernet Register Map	
14.6	Ethernet MAC Register Descriptions	
14.7	MII Management Register Descriptions	
15	Analog Comparators	566
15.1	Block Diagram	
15.2	Signal Description	
15.3	Functional Description	
15.3.1	Internal Reference Programming	
15.4	Initialization and Configuration	
15.5	Register Map	
15.6	Register Descriptions	
16	Pulse Width Modulator (PWM)	
16.1	Block Diagram	
16.2	Signal Description	
16.3	Functional Description	
16.3.1	PWM Timer	
	PWM Comparators	
	PWM Signal Generator	
	Dead-Band Generator	
	Interrupt/ADC-Trigger Selector	
16.3.6	Synchronization Methods	
16.3.7	Fault Conditions	583
16.3.8	Output Control Block	584
16.4	Initialization and Configuration	584
16.5	Register Map	585
16.6	Register Descriptions	586
17	Pin Diagram	616
18	Signal Tables	
18.1	100-Pin LQFP Package Pin Tables	
18.1.1	Signals by Pin Number	
		622

18.1.3	Signals by Function, Except for GPIO	625
18.1.4	GPIO Pins and Alternate Functions	628
18.2	108-Ball BGA Package Pin Tables	629
18.2.1	Signals by Pin Number	629
18.2.2	Signals by Signal Name	
18.2.3	Signals by Function, Except for GPIO	
18.2.4	GPIO Pins and Alternate Functions	
18.3	Connections for Unused Signals	
19	Operating Characteristics	
20	Electrical Characteristics	645
20.1	DC Characteristics	
20.1.1	Maximum Ratings	645
20.1.2	Recommended DC Operating Conditions	
20.1.3		
	GPIO Module Characteristics	
20.1.5	Power Specifications	
20.1.6	Flash Memory Characteristics	
20.1.7	Ethernet Controller	648
20.2	AC Characteristics	
	Load Conditions	
20.2.2	Clocks	648
20.2.3	JTAG and Boundary Scan	
20.2.4	Reset	
20.2.5	Sleep Modes	
	General-Purpose I/O (GPIO)	
	Analog-to-Digital Converter	
	Synchronous Serial Interface (SSI)	
20.2.9	Inter-Integrated Circuit (I ² C) Interface	
20.2.10	Ethernet Controller	
	Analog Comparator	
Α	Serial Flash Loader	
A.1	Serial Flash Loader	
A.2	Interfaces	
A.2.1	UART	
A.2.2	SSI	
A.3	Packet Handling	
A.3.1	Packet Format	
A.3.2	Sending Packets	
A.3.3	Receiving Packets	
A.4	Commands	
A.4.1	COMMAND_PING (0X20)	
A.4.2	COMMAND_GET_STATUS (0x23)	
A.4.3	COMMAND_DOWNLOAD (0x21)	
A.4.4	COMMAND_SEND_DATA (0x24)	
A.4.5	COMMAND_RUN (0x22)	
A.4.6	COMMAND_RESET (0x25)	
В		666

Ordering and Contact Information	687
Part Markings	687
Kits	687
Support Information	688
Package Information	689
100-Pin LQFP Package	689
Package Dimensions	689
Tray Dimensions	691
Tape and Reel Dimensions	691
108-Ball BGA Package	693
Package Dimensions	693
Tray Dimensions	695
Tape and Reel Dimensions	696
	Ordering Information Part Markings Kits Support Information Package Information 100-Pin LQFP Package Package Dimensions Tray Dimensions Tape and Reel Dimensions 108-Ball BGA Package Package Dimensions Tray Dimensions Tray Dimensions

List of Figures

Figure 1-1.	Stellaris LM3S6432 Microcontroller High-Level Block Diagram	42
Figure 2-1.	CPU Block Diagram	51
Figure 2-2.	TPIU Block Diagram	52
Figure 2-3.	Cortex-M3 Register Set	54
Figure 2-4.	Bit-Band Mapping	74
Figure 2-5.	Data Storage	75
Figure 2-6.	Vector Table	81
Figure 2-7.	Exception Stack Frame	83
Figure 3-1.	SRD Use Example	97
Figure 4-1.	JTAG Module Block Diagram	156
Figure 4-2.	Test Access Port State Machine	160
Figure 4-3.	IDCODE Register Format	166
Figure 4-4.	BYPASS Register Format	166
Figure 4-5.	Boundary Scan Register Format	167
Figure 5-1.	Basic RST Configuration	170
Figure 5-2.	External Circuitry to Extend Power-On Reset	171
Figure 5-3.	Reset Circuit Controlled by Switch	171
Figure 5-4.	Power Architecture	174
Figure 5-5.	Main Clock Tree	176
Figure 6-1.	Flash Block Diagram	231
Figure 7-1.	GPIO Port Block Diagram	264
Figure 7-2.	GPIODATA Write Example	265
Figure 7-3.	GPIODATA Read Example	265
Figure 8-1.	GPTM Module Block Diagram	306
Figure 8-2.	16-Bit Input Edge Count Mode Example	310
Figure 8-3.	16-Bit Input Edge Time Mode Example	311
Figure 8-4.	16-Bit PWM Mode Example	312
Figure 9-1.	WDT Module Block Diagram	342
Figure 10-1.	ADC Module Block Diagram	366
Figure 10-2.	Differential Sampling Range, V _{IN_ODD} = 1.5 V	370
Figure 10-3.	Differential Sampling Range, V _{IN_ODD} = 0.75 V	370
Figure 10-4.	Differential Sampling Range, V _{IN_ODD} = 2.25 V	371
Figure 10-5.	Internal Temperature Sensor Characteristic	
Figure 11-1.	UART Module Block Diagram	
Figure 11-2.	UART Character Frame	403
Figure 11-3.	IrDA Data Modulation	405
Figure 12-1.	SSI Module Block Diagram	443
Figure 12-2.	TI Synchronous Serial Frame Format (Single Transfer)	446
Figure 12-3.	TI Synchronous Serial Frame Format (Continuous Transfer)	447
Figure 12-4.	Freescale SPI Format (Single Transfer) with SPO=0 and SPH=0	447
Figure 12-5.	Freescale SPI Format (Continuous Transfer) with SPO=0 and SPH=0	448
Figure 12-6.	Freescale SPI Frame Format with SPO=0 and SPH=1	
Figure 12-7.	Freescale SPI Frame Format (Single Transfer) with SPO=1 and SPH=0	449
Figure 12-8.	Freescale SPI Frame Format (Continuous Transfer) with SPO=1 and SPH=0	
Figure 12-9.	Freescale SPI Frame Format with SPO=1 and SPH=1	
Figure 12-10.	MICROWIRE Frame Format (Single Frame)	451

Figure 12-11.	MICROWIRE Frame Format (Continuous Transfer)	452
Figure 12-12.	MICROWIRE Frame Format, SSIFss Input Setup and Hold Requirements	453
Figure 13-1.	I ² C Block Diagram	482
Figure 13-2.	I ² C Bus Configuration	483
Figure 13-3.	START and STOP Conditions	483
Figure 13-4.	Complete Data Transfer with a 7-Bit Address	484
Figure 13-5.	R/S Bit in First Byte	
Figure 13-6.	Data Validity During Bit Transfer on the I ² C Bus	484
Figure 13-7.	Master Single SEND	488
Figure 13-8.	Master Single RECEIVE	489
Figure 13-9.	Master Burst SEND	490
Figure 13-10.	Master Burst RECEIVE	
Figure 13-11.	Master Burst RECEIVE after Burst SEND	492
Figure 13-12.	Master Burst SEND after Burst RECEIVE	493
Figure 13-13.	Slave Command Sequence	494
Figure 14-1.	Ethernet Controller	519
Figure 14-2.	Ethernet Controller Block Diagram	519
Figure 14-3.	Ethernet Frame	
Figure 14-4.	Interface to an Ethernet Jack	527
Figure 15-1.	Analog Comparator Module Block Diagram	567
Figure 15-2.	Structure of Comparator Unit	568
Figure 15-3.	Comparator Internal Reference Structure	
Figure 16-1.	PWM Unit Diagram	
Figure 16-2.	PWM Module Block Diagram	
Figure 16-3.	PWM Count-Down Mode	581
Figure 16-4.	PWM Count-Up/Down Mode	582
Figure 16-5.	PWM Generation Example In Count-Up/Down Mode	
Figure 16-6.	PWM Dead-Band Generator	
Figure 17-1.	100-Pin LQFP Package Pin Diagram	616
Figure 17-2.	108-Ball BGA Package Pin Diagram (Top View)	
Figure 20-1.	Load Conditions	
Figure 20-2.	JTAG Test Clock Input Timing	651
Figure 20-3.	JTAG Test Access Port (TAP) Timing	651
Figure 20-4.	JTAG TRST Timing	
-	External Reset Timing (RST)	
-	Power-On Reset Timing	
Figure 20-7.	Brown-Out Reset Timing	652
Figure 20-8.	Software Reset Timing	
Figure 20-9.	Watchdog Reset Timing	653
-	ADC Input Equivalency Diagram	
•	SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing	
J	Measurement	655
Figure 20-12.	SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer	656
-	SSI Timing for SPI Frame Format (FRF=00), with SPH=1	
•	I ² C Timing	
-	External XTLP Oscillator Characteristics	
•	Stellaris LM3S6432 100-Pin LQFP Package Dimensions	
	100-Pin LQFP Tray Dimensions	

Figure D-3.	100-Pin LQFP Tape and Reel Dimensions	692
Figure D-4.	Stellaris LM3S6432 108-Ball BGA Package Dimensions	693
Figure D-5.	108-Ball BGA Tray Dimensions	695
Figure D-6.	108-Ball BGA Tape and Reel Dimensions	696

List of Tables

Table 1.	Revision History	24
Table 2.	Documentation Conventions	
Table 2-1.	Summary of Processor Mode, Privilege Level, and Stack Use	54
Table 2-2.	Processor Register Map	55
Table 2-3.	PSR Register Combinations	60
Table 2-4.	Memory Map	68
Table 2-5.	Memory Access Behavior	70
Table 2-6.	SRAM Memory Bit-Banding Regions	72
Table 2-7.	Peripheral Memory Bit-Banding Regions	72
Table 2-8.	Exception Types	78
Table 2-9.	Interrupts	79
Table 2-10.	Exception Return Behavior	84
Table 2-11.	Faults	85
Table 2-12.	Fault Status and Fault Address Registers	86
Table 2-13.	Cortex-M3 Instruction Summary	88
Table 3-1.	Core Peripheral Register Regions	91
Table 3-2.	Memory Attributes Summary	94
Table 3-3.	TEX, S, C, and B Bit Field Encoding	97
Table 3-4.	Cache Policy for Memory Attribute Encoding	98
Table 3-5.	AP Bit Field Encoding	98
Table 3-6.	Memory Region Attributes for Stellaris Microcontrollers	98
Table 3-7.	Peripherals Register Map	99
Table 3-8.	Interrupt Priority Levels	124
Table 3-9.	Example SIZE Field Values	152
Table 4-1.	JTAG_SWD_SWO Signals (100LQFP)	156
Table 4-2.	JTAG_SWD_SWO Signals (108BGA)	157
Table 4-3.	JTAG Port Pins Reset State	157
Table 4-4.	JTAG Instruction Register Commands	164
Table 5-1.	System Control & Clocks Signals (100LQFP)	168
Table 5-2.	System Control & Clocks Signals (108BGA)	168
Table 5-3.	Reset Sources	169
Table 5-4.	Clock Source Options	175
Table 5-5.	Possible System Clock Frequencies Using the SYSDIV Field	177
Table 5-6.	Examples of Possible System Clock Frequencies Using the SYSDIV2 Field	177
Table 5-7.	System Control Register Map	181
Table 5-8.	RCC2 Fields that Override RCC fields	195
Table 6-1.	Flash Protection Policy Combinations	232
Table 6-2.	User-Programmable Flash Memory Resident Registers	236
Table 6-3.	Flash Register Map	
Table 7-1.	GPIO Pins With Non-Zero Reset Values	259
Table 7-2.	GPIO Pins and Alternate Functions (100LQFP)	
Table 7-3.	GPIO Pins and Alternate Functions (108BGA)	260
Table 7-4.	GPIO Signals (100LQFP)	261
Table 7-5.	GPIO Signals (108BGA)	
Table 7-6.	GPIO Pad Configuration Examples	
Table 7-7.	GPIO Interrupt Configuration Example	267

Table 7-8.	GPIO Register Map	269
Table 8-1.	Available CCP Pins	306
Table 8-2.	General-Purpose Timers Signals (100LQFP)	307
Table 8-3.	General-Purpose Timers Signals (108BGA)	307
Table 8-4.	16-Bit Timer With Prescaler Configurations	309
Table 8-5.	Timers Register Map	316
Table 9-1.	Watchdog Timer Register Map	343
Table 10-1.	ADC Signals (100LQFP)	366
Table 10-2.	ADC Signals (108BGA)	366
Table 10-3.	Samples and FIFO Depth of Sequencers	367
Table 10-4.	Differential Sampling Pairs	
Table 10-5.	ADC Register Map	373
Table 11-1.	UART Signals (100LQFP)	402
Table 11-2.	UART Signals (108BGA)	
Table 11-3.	UART Register Map	
Table 12-1.	SSI Signals (100LQFP)	444
Table 12-2.	SSI Signals (108BGA)	
Table 12-3.	SSI Register Map	
Table 13-1.	I2C Signals (100LQFP)	
Table 13-2.	I2C Signals (108BGA)	
Table 13-3.	Examples of I ² C Master Timer Period versus Speed Mode	
Table 13-4.	Inter-Integrated Circuit (I ² C) Interface Register Map	
Table 13-5.	Write Field Decoding for I2CMCS[3:0] Field (Sheet 1 of 3)	
Table 14-1.	Ethernet Signals (100LQFP)	
Table 14-2.	Ethernet Signals (108BGA)	
Table 14-3.	TX & RX FIFO Organization	
Table 14-4.	Ethernet Register Map	
Table 15-1.	Analog Comparators Signals (100LQFP)	
Table 15-2.	Analog Comparators Signals (108BGA)	
Table 15-3.	Internal Reference Voltage and ACREFCTL Field Values	
Table 15-4.	Analog Comparators Register Map	
Table 16-1.	PWM Signals (100LQFP)	
Table 16-2.	PWM Signals (108BGA)	
Table 16-3.	PWM Register Map	
Table 18-1.	Signals by Pin Number	
Table 18-2.	Signals by Signal Name	
Table 18-3.	Signals by Function, Except for GPIO	
Table 18-4.	GPIO Pins and Alternate Functions	
Table 18-5.	Signals by Pin Number	
Table 18-6.	Signals by Signal Name	
Table 18-7.	Signals by Function, Except for GPIO	
Table 18-8.	GPIO Pins and Alternate Functions	
Table 18-9.		
Table 18-10.	Connections for Unused Signals (100-pin LQFP)	
	Connections for Unused Signals, 108-pin BGA	
Table 19-1.	Temperature Characteristics	
Table 19-2.	Thermal Characteristics	
Table 19-3.	ESD Absolute Maximum Ratings	
Table 20-1.	Maximum Ratings	645

Table 20-2.	Recommended DC Operating Conditions	645
Table 20-3.	LDO Regulator Characteristics	646
Table 20-4.	GPIO Module DC Characteristics	646
Table 20-5.	Detailed Power Specifications	647
Table 20-6.	Flash Memory Characteristics	647
Table 20-7.	Ethernet Controller DC Characteristics	648
Table 20-8.	Phase Locked Loop (PLL) Characteristics	648
Table 20-9.	Actual PLL Frequency	649
Table 20-10.	Clock Characteristics	649
Table 20-11.	Crystal Characteristics	649
Table 20-12.	System Clock Characteristics with ADC Operation	650
Table 20-13.	JTAG Characteristics	
Table 20-14.	Reset Characteristics	
Table 20-15.	Sleep Modes AC Characteristics	653
Table 20-16.	GPIO Characteristics	653
Table 20-17.	ADC Characteristics	654
Table 20-18.	ADC Module Internal Reference Characteristics	655
Table 20-19.	SSI Characteristics	655
Table 20-20.	I ² C Characteristics	656
Table 20-21.	100BASE-TX Transmitter Characteristics	657
Table 20-22.	100BASE-TX Transmitter Characteristics (informative)	657
Table 20-23.	100BASE-TX Receiver Characteristics	658
Table 20-24.	10BASE-T Transmitter Characteristics	658
Table 20-25.	10BASE-T Transmitter Characteristics (informative)	658
Table 20-26.	10BASE-T Receiver Characteristics	658
Table 20-27.	Isolation Transformers	658
Table 20-28.	Ethernet Reference Crystal	659
Table 20-29.	External XTLP Oscillator Characteristics	659
Table 20-30.	Analog Comparator Characteristics	660
Table 20-31.	Analog Comparator Voltage Reference Characteristics	660

List of Registers

The Cortex	-M3 Processor	49
Register 1:	Cortex General-Purpose Register 0 (R0)	56
Register 2:	Cortex General-Purpose Register 1 (R1)	56
Register 3:	Cortex General-Purpose Register 2 (R2)	56
Register 4:	Cortex General-Purpose Register 3 (R3)	56
Register 5:	Cortex General-Purpose Register 4 (R4)	56
Register 6:	Cortex General-Purpose Register 5 (R5)	56
Register 7:	Cortex General-Purpose Register 6 (R6)	56
Register 8:	Cortex General-Purpose Register 7 (R7)	56
Register 9:	Cortex General-Purpose Register 8 (R8)	
Register 10:	Cortex General-Purpose Register 9 (R9)	
Register 11:	Cortex General-Purpose Register 10 (R10)	
Register 12:	Cortex General-Purpose Register 11 (R11)	
Register 13:	Cortex General-Purpose Register 12 (R12)	
Register 14:	Stack Pointer (SP)	57
Register 15:	Link Register (LR)	
Register 16:	Program Counter (PC)	
Register 17:	Program Status Register (PSR)	
Register 18:	Priority Mask Register (PRIMASK)	
Register 19:	Fault Mask Register (FAULTMASK)	
Register 20:	Base Priority Mask Register (BASEPRI)	66
Register 21:	Control Register (CONTROL)	67
Cortex-M3	Peripherals	91
Register 1:	SysTick Control and Status Register (STCTRL), offset 0x010	102
Register 2:	SysTick Reload Value Register (STRELOAD), offset 0x014	104
Register 3:	SysTick Current Value Register (STCURRENT), offset 0x018	105
Register 4:	Interrupt 0-31 Set Enable (EN0), offset 0x100	106
Register 5:	Interrupt 32-43 Set Enable (EN1), offset 0x104	107
Register 6:	Interrupt 0-31 Clear Enable (DIS0), offset 0x180	108
Register 7:	Interrupt 32-43 Clear Enable (DIS1), offset 0x184	109
Register 8:	Interrupt 0-31 Set Pending (PEND0), offset 0x200	110
Register 9:	Interrupt 32-43 Set Pending (PEND1), offset 0x204	111
Register 10:	Interrupt 0-31 Clear Pending (UNPEND0), offset 0x280	
Register 11:	Interrupt 32-43 Clear Pending (UNPEND1), offset 0x284	
Register 12:		
	Interrupt 0-31 Active Bit (ACTIVE0), offset 0x300	114
Register 13:	Interrupt 0-31 Active Bit (ACTIVE0), offset 0x300	
Register 14:		115
-	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304	115 116
Register 14:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304	115 116 116 116
Register 14: Register 15: Register 16:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304 Interrupt 0-3 Priority (PRI0), offset 0x400 Interrupt 4-7 Priority (PRI1), offset 0x404 Interrupt 8-11 Priority (PRI2), offset 0x408 Interrupt 12-15 Priority (PRI3), offset 0x40C	115 116 116 116
Register 14: Register 15: Register 16: Register 17:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304	115 116 116 116
Register 14: Register 15:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304 Interrupt 0-3 Priority (PRI0), offset 0x400 Interrupt 4-7 Priority (PRI1), offset 0x404 Interrupt 8-11 Priority (PRI2), offset 0x408 Interrupt 12-15 Priority (PRI3), offset 0x40C	115 116 116 116 116
Register 14: Register 15: Register 16: Register 17: Register 18:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304 Interrupt 0-3 Priority (PRI0), offset 0x400 Interrupt 4-7 Priority (PRI1), offset 0x404 Interrupt 8-11 Priority (PRI2), offset 0x408 Interrupt 12-15 Priority (PRI3), offset 0x40C Interrupt 16-19 Priority (PRI4), offset 0x410	115 116 116 116 116 116
Register 14: Register 15: Register 16: Register 17: Register 18: Register 19:	Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304 Interrupt 0-3 Priority (PRI0), offset 0x400 Interrupt 4-7 Priority (PRI1), offset 0x404 Interrupt 8-11 Priority (PRI2), offset 0x408 Interrupt 12-15 Priority (PRI3), offset 0x40C Interrupt 16-19 Priority (PRI4), offset 0x410 Interrupt 20-23 Priority (PRI5), offset 0x414	115 116 116 116 116 116 116

Register 23:	Interrupt 36-39 Priority (PRI9), offset 0x424	116
Register 24:	Interrupt 40-43 Priority (PRI10), offset 0x428	116
Register 25:	Software Trigger Interrupt (SWTRIG), offset 0xF00	118
Register 26:	CPU ID Base (CPUID), offset 0xD00	119
Register 27:	Interrupt Control and State (INTCTRL), offset 0xD04	120
Register 28:	Vector Table Offset (VTABLE), offset 0xD08	123
Register 29:	Application Interrupt and Reset Control (APINT), offset 0xD0C	124
Register 30:	System Control (SYSCTRL), offset 0xD10	126
Register 31:	Configuration and Control (CFGCTRL), offset 0xD14	128
Register 32:	System Handler Priority 1 (SYSPRI1), offset 0xD18	
Register 33:	System Handler Priority 2 (SYSPRI2), offset 0xD1C	131
Register 34:	System Handler Priority 3 (SYSPRI3), offset 0xD20	132
Register 35:	System Handler Control and State (SYSHNDCTRL), offset 0xD24	133
Register 36:	Configurable Fault Status (FAULTSTAT), offset 0xD28	
Register 37:	Hard Fault Status (HFAULTSTAT), offset 0xD2C	143
Register 38:	Memory Management Fault Address (MMADDR), offset 0xD34	144
Register 39:	Bus Fault Address (FAULTADDR), offset 0xD38	145
Register 40:	MPU Type (MPUTYPE), offset 0xD90	146
Register 41:	MPU Control (MPUCTRL), offset 0xD94	147
Register 42:	MPU Region Number (MPUNUMBER), offset 0xD98	149
Register 43:	MPU Region Base Address (MPUBASE), offset 0xD9C	
Register 44:	MPU Region Base Address Alias 1 (MPUBASE1), offset 0xDA4	
Register 45:	MPU Region Base Address Alias 2 (MPUBASE2), offset 0xDAC	
Register 46:	MPU Region Base Address Alias 3 (MPUBASE3), offset 0xDB4	150
Register 47:	MPU Region Attribute and Size (MPUATTR), offset 0xDA0	
Register 48:	MPU Region Attribute and Size Alias 1 (MPUATTR1), offset 0xDA8	
Register 49:	MPU Region Attribute and Size Alias 2 (MPUATTR2), offset 0xDB0	
Register 50:	MPU Region Attribute and Size Alias 3 (MPUATTR3), offset 0xDB8	
System Co	ntrol	168
Register 1:	Device Identification 0 (DID0), offset 0x000	
Register 2:	Brown-Out Reset Control (PBORCTL), offset 0x030	
Register 3:	LDO Power Control (LDOPCTL), offset 0x034	
Register 4:	Raw Interrupt Status (RIS), offset 0x050	
Register 5:	Interrupt Mask Control (IMC), offset 0x054	
Register 6:	Masked Interrupt Status and Clear (MISC), offset 0x058	
Register 7:	Reset Cause (RESC), offset 0x05C	
Register 8:	Run-Mode Clock Configuration (RCC), offset 0x060	
Register 9:	XTAL to PLL Translation (PLLCFG), offset 0x064	
Register 10:	Run-Mode Clock Configuration 2 (RCC2), offset 0x070	
Register 11:	Deep Sleep Clock Configuration (DSLPCLKCFG), offset 0x144	
Register 12:	Device Identification 1 (DID1), offset 0x004	
Register 13:	Device Capabilities 0 (DC0), offset 0x008	
Register 14:	Device Capabilities 1 (DC1), offset 0x000	
Register 15:	Device Capabilities 2 (DC2), offset 0x014	
Register 16:	Device Capabilities 3 (DC3), offset 0x014	
Register 17:	Device Capabilities 4 (DC4), offset 0x01C	
Register 18:	Run Mode Clock Gating Control Register 0 (RCGC0), offset 0x100	
Register 19:	Sleep Mode Clock Gating Control Register 0 (SCGC0), offset 0x100	
LOUIDIGE 10.	CIOUD MOUC CIOUR CUMING CONTROL INCUIDED O LOCUCOUNT. UHBEL UN HTD	

Register 20:	Deep Sleep Mode Clock Gating Control Register 0 (DCGC0), offset 0x120	213
Register 21:	Run Mode Clock Gating Control Register 1 (RCGC1), offset 0x104	215
Register 22:	Sleep Mode Clock Gating Control Register 1 (SCGC1), offset 0x114	217
Register 23:	Deep Sleep Mode Clock Gating Control Register 1 (DCGC1), offset 0x124	219
Register 24:	Run Mode Clock Gating Control Register 2 (RCGC2), offset 0x108	
Register 25:	Sleep Mode Clock Gating Control Register 2 (SCGC2), offset 0x118	223
Register 26:	Deep Sleep Mode Clock Gating Control Register 2 (DCGC2), offset 0x128	225
Register 27:	Software Reset Control 0 (SRCR0), offset 0x040	227
Register 28:	Software Reset Control 1 (SRCR1), offset 0x044	228
Register 29:	Software Reset Control 2 (SRCR2), offset 0x048	230
Internal Me	mory	231
Register 1:	Flash Memory Address (FMA), offset 0x000	238
Register 2:	Flash Memory Data (FMD), offset 0x004	
Register 3:	Flash Memory Control (FMC), offset 0x008	240
Register 4:	Flash Controller Raw Interrupt Status (FCRIS), offset 0x00C	242
Register 5:	Flash Controller Interrupt Mask (FCIM), offset 0x010	243
Register 6:	Flash Controller Masked Interrupt Status and Clear (FCMISC), offset 0x014	244
Register 7:	USec Reload (USECRL), offset 0x140	
Register 8:	Flash Memory Protection Read Enable 0 (FMPRE0), offset 0x130 and 0x200	247
Register 9:	Flash Memory Protection Program Enable 0 (FMPPE0), offset 0x134 and 0x400	248
Register 10:	User Debug (USER_DBG), offset 0x1D0	249
Register 11:	User Register 0 (USER_REG0), offset 0x1E0	250
Register 12:	User Register 1 (USER_REG1), offset 0x1E4	251
Register 13:	Flash Memory Protection Read Enable 1 (FMPRE1), offset 0x204	252
Register 14:	Flash Memory Protection Read Enable 2 (FMPRE2), offset 0x208	253
Register 15:	Flash Memory Protection Read Enable 3 (FMPRE3), offset 0x20C	254
Register 16:	Flash Memory Protection Program Enable 1 (FMPPE1), offset 0x404	255
Register 17:	Flash Memory Protection Program Enable 2 (FMPPE2), offset 0x408	256
Register 18:	Flash Memory Protection Program Enable 3 (FMPPE3), offset 0x40C	257
General-Pu	rpose Input/Outputs (GPIOs)	258
Register 1:	GPIO Data (GPIODATA), offset 0x000	271
Register 2:	GPIO Direction (GPIODIR), offset 0x400	272
Register 3:	GPIO Interrupt Sense (GPIOIS), offset 0x404	273
Register 4:	GPIO Interrupt Both Edges (GPIOIBE), offset 0x408	274
Register 5:	GPIO Interrupt Event (GPIOIEV), offset 0x40C	275
Register 6:	GPIO Interrupt Mask (GPIOIM), offset 0x410	276
Register 7:	GPIO Raw Interrupt Status (GPIORIS), offset 0x414	277
Register 8:	GPIO Masked Interrupt Status (GPIOMIS), offset 0x418	
Register 9:	GPIO Interrupt Clear (GPIOICR), offset 0x41C	279
Register 10:	GPIO Alternate Function Select (GPIOAFSEL), offset 0x420	280
Register 11:	GPIO 2-mA Drive Select (GPIODR2R), offset 0x500	282
Register 12:	GPIO 4-mA Drive Select (GPIODR4R), offset 0x504	283
Register 13:	GPIO 8-mA Drive Select (GPIODR8R), offset 0x508	
Register 14:	GPIO Open Drain Select (GPIOODR), offset 0x50C	285
Register 15:	GPIO Pull-Up Select (GPIOPUR), offset 0x510	286
Register 16:	GPIO Pull-Down Select (GPIOPDR), offset 0x514	
Register 17:	GPIO Slew Rate Control Select (GPIOSLR), offset 0x518	288
Register 18:	GPIO Digital Enable (GPIODEN), offset 0x51C	289

Register 19:	GPIO Lock (GPIOLOCK), offset 0x520	290
Register 20:	GPIO Commit (GPIOCR), offset 0x524	291
Register 21:	GPIO Peripheral Identification 4 (GPIOPeriphID4), offset 0xFD0	293
Register 22:	GPIO Peripheral Identification 5 (GPIOPeriphID5), offset 0xFD4	294
Register 23:	GPIO Peripheral Identification 6 (GPIOPeriphID6), offset 0xFD8	295
Register 24:	GPIO Peripheral Identification 7 (GPIOPeriphID7), offset 0xFDC	296
Register 25:	GPIO Peripheral Identification 0 (GPIOPeriphID0), offset 0xFE0	297
Register 26:	GPIO Peripheral Identification 1 (GPIOPeriphID1), offset 0xFE4	298
Register 27:	GPIO Peripheral Identification 2 (GPIOPeriphID2), offset 0xFE8	299
Register 28:	GPIO Peripheral Identification 3 (GPIOPeriphID3), offset 0xFEC	300
Register 29:	GPIO PrimeCell Identification 0 (GPIOPCellID0), offset 0xFF0	
Register 30:	GPIO PrimeCell Identification 1 (GPIOPCellID1), offset 0xFF4	302
Register 31:	GPIO PrimeCell Identification 2 (GPIOPCellID2), offset 0xFF8	
Register 32:	GPIO PrimeCell Identification 3 (GPIOPCellID3), offset 0xFFC	304
General-Pu	rpose Timers	305
Register 1:	GPTM Configuration (GPTMCFG), offset 0x000	317
Register 2:	GPTM TimerA Mode (GPTMTAMR), offset 0x004	318
Register 3:	GPTM TimerB Mode (GPTMTBMR), offset 0x008	320
Register 4:	GPTM Control (GPTMCTL), offset 0x00C	322
Register 5:	GPTM Interrupt Mask (GPTMIMR), offset 0x018	325
Register 6:	GPTM Raw Interrupt Status (GPTMRIS), offset 0x01C	327
Register 7:	GPTM Masked Interrupt Status (GPTMMIS), offset 0x020	328
Register 8:	GPTM Interrupt Clear (GPTMICR), offset 0x024	
Register 9:	GPTM TimerA Interval Load (GPTMTAILR), offset 0x028	331
Register 10:	GPTM TimerB Interval Load (GPTMTBILR), offset 0x02C	332
Register 11:	GPTM TimerA Match (GPTMTAMATCHR), offset 0x030	333
Register 12:	GPTM TimerB Match (GPTMTBMATCHR), offset 0x034	334
Register 13:	GPTM TimerA Prescale (GPTMTAPR), offset 0x038	
Register 14:	GPTM TimerB Prescale (GPTMTBPR), offset 0x03C	
Register 15:	GPTM TimerA Prescale Match (GPTMTAPMR), offset 0x040	
Register 16:	GPTM TimerB Prescale Match (GPTMTBPMR), offset 0x044	
Register 17:	GPTM TimerA (GPTMTAR), offset 0x048	
Register 18:	GPTM TimerB (GPTMTBR), offset 0x04C	340
Watchdog 1	Fimer	341
Register 1:	Watchdog Load (WDTLOAD), offset 0x000	345
Register 2:	Watchdog Value (WDTVALUE), offset 0x004	346
Register 3:	Watchdog Control (WDTCTL), offset 0x008	347
Register 4:	Watchdog Interrupt Clear (WDTICR), offset 0x00C	348
Register 5:	Watchdog Raw Interrupt Status (WDTRIS), offset 0x010	349
Register 6:	Watchdog Masked Interrupt Status (WDTMIS), offset 0x014	350
Register 7:	Watchdog Test (WDTTEST), offset 0x418	351
Register 8:	Watchdog Lock (WDTLOCK), offset 0xC00	
Register 9:	Watchdog Peripheral Identification 4 (WDTPeriphID4), offset 0xFD0	
Register 10:	Watchdog Peripheral Identification 5 (WDTPeriphID5), offset 0xFD4	
Register 11:	Watchdog Peripheral Identification 6 (WDTPeriphID6), offset 0xFD8	
Register 12:	Watchdog Peripheral Identification 7 (WDTPeriphID7), offset 0xFDC	
Register 13:	Watchdog Peripheral Identification 0 (WDTPeriphID0), offset 0xFE0	
Register 14:	Watchdog Peripheral Identification 1 (WDTPeriphID1), offset 0xFE4	358

Register 15:	Watchdog Peripheral Identification 2 (WDTPeriphID2), offset 0xFE8	359
Register 16:	Watchdog Peripheral Identification 3 (WDTPeriphID3), offset 0xFEC	360
Register 17:	Watchdog PrimeCell Identification 0 (WDTPCellID0), offset 0xFF0	361
Register 18:	Watchdog PrimeCell Identification 1 (WDTPCellID1), offset 0xFF4	362
Register 19:	Watchdog PrimeCell Identification 2 (WDTPCellID2), offset 0xFF8	363
Register 20:	Watchdog PrimeCell Identification 3 (WDTPCellID3), offset 0xFFC	364
Analog-to-[Digital Converter (ADC)	365
Register 1:	ADC Active Sample Sequencer (ADCACTSS), offset 0x000	
Register 2:	ADC Raw Interrupt Status (ADCRIS), offset 0x004	
Register 3:	ADC Interrupt Mask (ADCIM), offset 0x008	
Register 4:	ADC Interrupt Status and Clear (ADCISC), offset 0x00C	
Register 5:	ADC Overflow Status (ADCOSTAT), offset 0x010	
Register 6:	ADC Event Multiplexer Select (ADCEMUX), offset 0x014	
Register 7:	ADC Underflow Status (ADCUSTAT), offset 0x018	
Register 8:	ADC Sample Sequencer Priority (ADCSSPRI), offset 0x020	
Register 9:	ADC Processor Sample Sequence Initiate (ADCPSSI), offset 0x028	
Register 10:	ADC Sample Averaging Control (ADCSAC), offset 0x030	387
Register 11:	ADC Sample Sequence Input Multiplexer Select 0 (ADCSSMUX0), offset 0x040	
Register 12:	ADC Sample Sequence Control 0 (ADCSSCTL0), offset 0x044	
Register 13:	ADC Sample Sequence Result FIFO 0 (ADCSSFIFO0), offset 0x048	393
Register 14:	ADC Sample Sequence Result FIFO 1 (ADCSSFIFO1), offset 0x068	
Register 15:	ADC Sample Sequence Result FIFO 2 (ADCSSFIFO2), offset 0x088	
Register 16:	ADC Sample Sequence Result FIFO 3 (ADCSSFIFO3), offset 0x0A8	393
Register 17:	ADC Sample Sequence FIFO 0 Status (ADCSSFSTAT0), offset 0x04C	394
Register 18:	ADC Sample Sequence FIFO 1 Status (ADCSSFSTAT1), offset 0x06C	
Register 19:	ADC Sample Sequence FIFO 2 Status (ADCSSFSTAT2), offset 0x08C	394
Register 20:	ADC Sample Sequence FIFO 3 Status (ADCSSFSTAT3), offset 0x0AC	394
Register 21:	ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1), offset 0x060	395
Register 22:	ADC Sample Sequence Input Multiplexer Select 2 (ADCSSMUX2), offset 0x080	395
Register 23:	ADC Sample Sequence Control 1 (ADCSSCTL1), offset 0x064	396
Register 24:	ADC Sample Sequence Control 2 (ADCSSCTL2), offset 0x084	396
Register 25:	ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3), offset 0x0A0	398
Register 26:	ADC Sample Sequence Control 3 (ADCSSCTL3), offset 0x0A4	399
Register 27:	ADC Test Mode Loopback (ADCTMLB), offset 0x100	400
Universal A	synchronous Receivers/Transmitters (UARTs)	401
Register 1:	UART Data (UARTDR), offset 0x000	
Register 2:	UART Receive Status/Error Clear (UARTRSR/UARTECR), offset 0x004	412
Register 3:	UART Flag (UARTFR), offset 0x018	414
Register 4:	UART IrDA Low-Power Register (UARTILPR), offset 0x020	416
Register 5:	UART Integer Baud-Rate Divisor (UARTIBRD), offset 0x024	417
Register 6:	UART Fractional Baud-Rate Divisor (UARTFBRD), offset 0x028	
Register 7:	UART Line Control (UARTLCRH), offset 0x02C	419
Register 8:	UART Control (UARTCTL), offset 0x030	421
Register 9:	UART Interrupt FIFO Level Select (UARTIFLS), offset 0x034	
Register 10:	UART Interrupt Mask (UARTIM), offset 0x038	
Register 11:	UART Raw Interrupt Status (UARTRIS), offset 0x03C	
Register 12:	UART Masked Interrupt Status (UARTMIS), offset 0x040	428
Register 13:	UART Interrupt Clear (UARTICR), offset 0x044	

Register 14:	UART Peripheral Identification 4 (UARTPeriphID4), offset 0xFD0	431
Register 15:	UART Peripheral Identification 5 (UARTPeriphID5), offset 0xFD4	432
Register 16:	UART Peripheral Identification 6 (UARTPeriphID6), offset 0xFD8	
Register 17:	UART Peripheral Identification 7 (UARTPeriphID7), offset 0xFDC	
Register 18:	UART Peripheral Identification 0 (UARTPeriphID0), offset 0xFE0	
Register 19:	UART Peripheral Identification 1 (UARTPeriphID1), offset 0xFE4	
Register 20:	UART Peripheral Identification 2 (UARTPeriphID2), offset 0xFE8	
Register 21:	UART Peripheral Identification 3 (UARTPeriphID3), offset 0xFEC	
Register 22:	UART PrimeCell Identification 0 (UARTPCellID0), offset 0xFF0	
Register 23:	UART PrimeCell Identification 1 (UARTPCellID1), offset 0xFF4	
Register 24:	UART PrimeCell Identification 2 (UARTPCellID2), offset 0xFF8	
Register 25:	UART PrimeCell Identification 3 (UARTPCellID3), offset 0xFFC	442
Synchrono	us Serial Interface (SSI)	443
Register 1:	SSI Control 0 (SSICR0), offset 0x000	
Register 2:	SSI Control 1 (SSICR1), offset 0x004	458
Register 3:	SSI Data (SSIDR), offset 0x008	
Register 4:	SSI Status (SSISR), offset 0x00C	
Register 5:	SSI Clock Prescale (SSICPSR), offset 0x010	
Register 6:	SSI Interrupt Mask (SSIIM), offset 0x014	
Register 7:	SSI Raw Interrupt Status (SSIRIS), offset 0x018	
Register 8:	SSI Masked Interrupt Status (SSIMIS), offset 0x01C	
Register 9:	SSI Interrupt Clear (SSIICR), offset 0x020	
Register 10:	SSI Peripheral Identification 4 (SSIPeriphID4), offset 0xFD0	
Register 11:	SSI Peripheral Identification 5 (SSIPeriphID5), offset 0xFD4	
Register 12:	SSI Peripheral Identification 6 (SSIPeriphID6), offset 0xFD8	
Register 13:	SSI Peripheral Identification 7 (SSIPeriphID7), offset 0xFDC	
Register 14:	SSI Peripheral Identification 0 (SSIPeriphID0), offset 0xFE0	
Register 15:	SSI Peripheral Identification 1 (SSIPeriphID1), offset 0xFE4	
Register 16:	SSI Peripheral Identification 2 (SSIPeriphID2), offset 0xFE8	
Register 17:	SSI Peripheral Identification 3 (SSIPeriphID3), offset 0xFEC	
Register 18:	SSI PrimeCell Identification 0 (SSIPCellID0), offset 0xFF0	
Register 19:	SSI PrimeCell Identification 1 (SSIPCellID1), offset 0xFF4	
Register 20:	SSI PrimeCell Identification 2 (SSIPCellID2), offset 0xFF8	
Register 21:	SSI PrimeCell Identification 3 (SSIPCellID3), offset 0xFFC	
_	rated Circuit (I ² C) Interface	
Register 1:	I ² C Master Slave Address (I2CMSA), offset 0x000	
Register 2:	I ² C Master Control/Status (I2CMCS), offset 0x004	
Register 3:	I ² C Master Data (I2CMDR), offset 0x008	
Register 4:	I ² C Master Timer Period (I2CMTPR), offset 0x00C	
Register 5:	I ² C Master Interrupt Mask (I2CMIMR), offset 0x010	
Register 6:	I ² C Master Raw Interrupt Status (I2CMRIS), offset 0x014	
Register 7:	I ² C Master Masked Interrupt Status (I2CMMIS), offset 0x018	
Register 8:	I ² C Master Interrupt Clear (I2CMICR), offset 0x01C	
Register 9:	I ² C Master Configuration (I2CMCR), offset 0x020	
Register 10:	I ² C Slave Own Address (I2CSOAR), offset 0x800	
Register 11:	I ² C Slave Control/Status (I2CSCSR), offset 0x804	
Register 12:	I ² C Slave Data (I2CSDR), offset 0x808	513

Register 13:	I ² C Slave Interrupt Mask (I2CSIMR), offset 0x80C	514
Register 14:	I ² C Slave Raw Interrupt Status (I2CSRIS), offset 0x810	515
Register 15:	I ² C Slave Masked Interrupt Status (I2CSMIS), offset 0x814	516
Register 16:	I ² C Slave Interrupt Clear (I2CSICR), offset 0x818	
Ethernet Co	ontroller	. 518
Register 1:	Ethernet MAC Raw Interrupt Status/Acknowledge (MACRIS/MACIACK), offset 0x000	
Register 2:	Ethernet MAC Interrupt Mask (MACIM), offset 0x004	
Register 3:	Ethernet MAC Receive Control (MACRCTL), offset 0x008	
Register 4:	Ethernet MAC Transmit Control (MACTCTL), offset 0x00C	
Register 5:	Ethernet MAC Data (MACDATA), offset 0x010	
Register 6:	Ethernet MAC Individual Address 0 (MACIA0), offset 0x014	
Register 7:	Ethernet MAC Individual Address 1 (MACIA1), offset 0x018	
Register 8:	Ethernet MAC Threshold (MACTHR), offset 0x01C	
Register 9:	Ethernet MAC Management Control (MACMCTL), offset 0x020	
Register 10:	Ethernet MAC Management Divider (MACMDV), offset 0x024	
Register 11:	Ethernet MAC Management Transmit Data (MACMTXD), offset 0x02C	
Register 12:	Ethernet MAC Management Receive Data (MACMRXD), offset 0x030	
Register 13:	Ethernet MAC Number of Packets (MACNP), offset 0x034	
Register 14:	Ethernet MAC Transmission Request (MACTR), offset 0x038	
Register 15:	Ethernet PHY Management Register 0 – Control (MR0), address 0x00	
Register 16:	Ethernet PHY Management Register 1 – Status (MR1), address 0x01	
Register 17:	Ethernet PHY Management Register 2 – PHY Identifier 1 (MR2), address 0x02	
Register 18:	Ethernet PHY Management Register 3 – PHY Identifier 2 (MR3), address 0x03	
Register 19:	Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4), address 0x04	
Register 20:	Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability (MR5), address 0x05	
Register 21:	Ethernet PHY Management Register 6 – Auto-Negotiation Expansion (MR6), address 0x06	
Register 22:	Ethernet PHY Management Register 16 – Vendor-Specific (MR16), address 0x10	
Register 23:	Ethernet PHY Management Register 17 – Interrupt Control/Status (MR17), address 0x11	
Register 24:	Ethernet PHY Management Register 18 – Diagnostic (MR18), address 0x12	
Register 25:	Ethernet PHY Management Register 19 – Transceiver Control (MR19), address 0x13	
Register 26:	Ethernet PHY Management Register 23 – LED Configuration (MR23), address 0x17	
Register 27:	Ethernet PHY Management Register 24 –MDI/MDIX Control (MR24), address 0x18	
Analog Cor	nparators	. 566
Register 1:	Analog Comparator Masked Interrupt Status (ACMIS), offset 0x000	
Register 2:	Analog Comparator Raw Interrupt Status (ACRIS), offset 0x004	
Register 3:	Analog Comparator Interrupt Enable (ACINTEN), offset 0x008	
Register 4:	Analog Comparator Reference Voltage Control (ACREFCTL), offset 0x010	
Register 5:	Analog Comparator Status 0 (ACSTAT0), offset 0x020	
Register 6:	Analog Comparator Status 1 (ACSTAT1), offset 0x040	
Register 7:	Analog Comparator Control 0 (ACCTL0), offset 0x024	
Register 8:	Analog Comparator Control 1 (ACCTL1), offset 0x044	
•	h Modulator (PWM)	
Register 1:	PWM Master Control (PWMCTL), offset 0x000	
Register 2:	PWM Time Base Sync (PWMSYNC), offset 0x004	

Register 3:	PWM Output Enable (PWMENABLE), offset 0x008	589
Register 4:	PWM Output Inversion (PWMINVERT), offset 0x00C	
Register 5:	PWM Output Fault (PWMFAULT), offset 0x010	
Register 6:	PWM Interrupt Enable (PWMINTEN), offset 0x014	592
Register 7:	PWM Raw Interrupt Status (PWMRIS), offset 0x018	593
Register 8:	PWM Interrupt Status and Clear (PWMISC), offset 0x01C	594
Register 9:	PWM Status (PWMSTATUS), offset 0x020	595
Register 10:	PWM0 Control (PWM0CTL), offset 0x040	596
Register 11:	PWM0 Interrupt and Trigger Enable (PWM0INTEN), offset 0x044	598
Register 12:	PWM0 Raw Interrupt Status (PWM0RIS), offset 0x048	601
Register 13:	PWM0 Interrupt Status and Clear (PWM0ISC), offset 0x04C	602
Register 14:	PWM0 Load (PWM0LOAD), offset 0x050	603
Register 15:	PWM0 Counter (PWM0COUNT), offset 0x054	604
Register 16:	PWM0 Compare A (PWM0CMPA), offset 0x058	605
Register 17:	PWM0 Compare B (PWM0CMPB), offset 0x05C	606
Register 18:	PWM0 Generator A Control (PWM0GENA), offset 0x060	607
Register 19:	PWM0 Generator B Control (PWM0GENB), offset 0x064	610
Register 20:	PWM0 Dead-Band Control (PWM0DBCTL), offset 0x068	613
Register 21:	PWM0 Dead-Band Rising-Edge Delay (PWM0DBRISE), offset 0x06C	614
Register 22:	PWM0 Dead-Band Falling-Edge-Delay (PWM0DBFALL), offset 0x070	615

Revision History

The revision history table notes changes made between the indicated revisions of the LM3S6432 data sheet.

Table 1. Revision History

Date	Revision	Description
July 2014	15852.2743	■ In JTAG chapter, clarified JTAG-to-SWD Switching and SWD-to-JTAG Switching.
		■ In System Control chapter, clarified behavior of Reset Cause (RESC) register external reset bit.
		 In Internal Memory chapter: Added sections on Execute-Only Protection, Read-Only Protection, and Permanently Disabling Debug. Noted that the Boot Configuration (BOOTCFG) register requires a POR before committed changes to the Flash-resident registers take effect.
		 In UART chapter: Clarified that the transmit interrupt is based on a transition through level. Corrected reset for UART Raw Interrupt Status (UARTRIS) register.
		■ In Electrical Characteristics chapter, updated Crystal Characteristics and Ethernet Reference Crystal tables.
		■ In Ordering and Contact Information appendix, moved orderable part numbers table to addendum.
		Additional minor data sheet clarifications and corrections.
June 2012	12746.2515	■ Corrected missing interrupt 9 in "Interrupts" table.
		Minor data sheet clarifications and corrections.
November 2011	11108	■ Added module-specific pin tables to each chapter in the new Signal Description sections.
		■ In Timer chapter, clarified that in 16-Bit Input Edge Time Mode, the timer is capable of capturing three types of events: rising edge, falling edge, or both.
		■ In UART chapter, clarified interrupt behavior.
		■ In SSI chapter, corrected SSICIk in the figure "Synchronous Serial Frame Format (Single Transfer)".
		■ In Signal Tables chapter:
		Corrected pin numbers in table "Connections for Unused Signals" (other pin tables were correct).
		Corrected buffer type for PWMn signals in pin tables.
		■ In Electrical Characteristics chapter:
		 Added parameter "Input voltage for a GPIO configured as an analog input" to the "Maximum Ratings" table.
		 Corrected Nom values for parameters "TCK clock Low time" and "TCK clock High time" in "JTAG Characteristics" table.
		 Corrected missing values for "Conversion time" and "Conversion rate" parameters in "ADC Characteristics" table.
		Additional minor data sheet clarifications and corrections.

Table 1. Revision History (continued)

Date	Revision	Description
January 2011	9102	■ In Application Interrupt and Reset Control (APINT) register, changed bit name from SYSRESETREQ to SYSRESREQ.
		■ Added DEBUG (Debug Priority) bit field to System Handler Priority 3 (SYSPRI3) register.
		■ Added "Reset Sources" table to System Control chapter.
		Removed mention of false-start bit detection in the UART chapter. This feature is not supported.
		■ Added note that specific module clocks must be enabled before that module's registers can be programmed. There must be a delay of 3 system clocks after the module clock is enabled before any of that module's registers are accessed.
		■ Changed I ² C slave register base addresses and offsets to be relative to the I ² C module base address of 0x4002.0000, so register bases and offsets were changed for all I ² C slave registers. Note that the hw_i2c.h file in the StellarisWare [®] Driver Library uses a base address of 0x4002.0800 for the I ² C slave registers. Be aware when using registers with offsets between 0x800 and 0x818 that StellarisWare uses the old slave base address for these offsets.
		■ Added GNDPHY and VCCPHY to Connections for Unused Signals tables.
		■ Corrected nonlinearity and offset error parameters (E _L , E _D and E _O) in ADC Characteristics table.
		Added specification for maximum input voltage on a non-power pin when the microcontroller is unpowered (V _{NON} parameter in Maximum Ratings table).
		Additional minor data sheet clarifications and corrections.
September 2010	7787	■ Reorganized ARM Cortex-M3 Processor Core, Memory Map and Interrupts chapters, creating two new chapters, The Cortex-M3 Processor and Cortex-M3 Peripherals. Much additional content was added, including all the Cortex-M3 registers.
		■ Changed register names to be consistent with StellarisWare names: the Cortex-M3 Interrupt Control and Status (ICSR) register to the Interrupt Control and State (INTCTRL) register, and the Cortex-M3 Interrupt Set Enable (SETNA) register to the Interrupt 0-31 Set Enable (EN0) register.
		■ Added clarification of instruction execution during Flash operations.
		■ Modified Figure 7-1 on page 264 to clarify operation of the GPIO inputs when used as an alternate function.
		■ Added caution not to apply a Low value to PB7 when debugging; a Low value on the pin causes the JTAG controller to be reset, resulting in a loss of JTAG communication.
		■ In General-Purpose Timers chapter, clarified operation of the 32-bit RTC mode.
		■ In Electrical Characteristics chapter: - Added I _{LKG} parameter (GPIO input leakage current) to Table 20-4 on page 646. - Corrected values for t _{CLKRF} parameter (SSIC1k rise/fall time) in Table 20-19 on page 655. - Added "Ethernet Controller DC Characteristics" table (see Table 20-7 on page 648).
		■ Added dimensions for Tray and Tape and Reel shipping mediums.

Table 1. Revision History (continued)

Date	Revision	Description
June 2010	7393	■ Corrected base address for SRAM in architectural overview chapter.
		■ Clarified system clock operation, adding content to "Clock Control" on page 174.
		■ In Signal Tables chapter, added table "Connections for Unused Signals."
		■ In "Thermal Characteristics" table, corrected thermal resistance value from 34 to 32.
		■ In "Reset Characteristics" table, corrected value for supply voltage (VDD) rise time.
		Additional minor data sheet clarifications and corrections.
April 2010	7007	■ Added caution note to the I ² C Master Timer Period (I2CMTPR) register description and changed field width to 7 bits.
		■ Removed erroneous text about restoring the Flash Protection registers.
		■ Added note about RST signal routing.
		■ Clarified the function of the TnSTALL bit in the GPTMCTL register.
		■ Corrected XTALNPHY pin description.
		■ Additional minor data sheet clarifications and corrections.
January 2010	6712	■ In "System Control" section, clarified Debug Access Port operation after Sleep modes.
		■ Clarified wording on Flash memory access errors.
		■ Added section on Flash interrupts.
		■ Changed the reset value of the ADC Sample Sequence Result FIFO n (ADCSSFIFOn) registers to be indeterminate.
		■ Clarified operation of SSI transmit FIFO.
		■ Made these changes to the Operating Characteristics chapter:
		Added storage temperature ratings to "Temperature Characteristics" table
		Added "ESD Absolute Maximum Ratings" table
		■ Made these changes to the Electrical Characteristics chapter:
		In "Flash Memory Characteristics" table, corrected Mass erase time
		Added sleep and deep-sleep wake-up times ("Sleep Modes AC Characteristics" table)
		In "Reset Characteristics" table, corrected units for supply voltage (VDD) rise time

Table 1. Revision History (continued)

Date	Revision	Description
October 2009	6462	■ Deleted MAXADCSPD bit field from DCGC0 register as it is not applicable in Deep-Sleep mode.
		■ Deleted reset value for 16-bit mode from GPTMTAILR , GPTMTAMATCHR , and GPTMTAR registers because the module resets in 32-bit mode.
		■ Clarified PWM source for ADC triggering.
		■ Made these changes to the Electrical Characteristics chapter:
		$-\hspace{0.1cm}$ Removed V_{SIH} and V_{SIL} parameters from Operating Conditions table.
		Added table showing actual PLL frequency depending on input crystal.
		Changed the name of the t _{HIB_REG_WRITE} parameter to t _{HIB_REG_ACCESS} .
		Revised ADC electrical specifications to clarify, including reorganizing and adding new data.
		Changed SSI set up and hold times to be expressed in system clocks, not ns.
July 2009	5920	Corrected ordering numbers.
July 2009	5902	■ Clarified Power-on reset and RST pin operation; added new diagrams.
		Clarified explanation of nonvolatile register programming in Internal Memory chapter.
		■ Added explanation of reset value to FMPRE0/1/2/3, FMPPE0/1/2/3, USER_DBG, and USER_REG0/1 registers.
		■ Added description for Ethernet PHY power-saving modes.
		■ Corrected the reset values for bits 6 and 7 in the Ethernet MR24 register.
		■ Changed buffer type for WAKE pin to TTL and HIB pin to OD.
		■ In ADC characteristics table, changed Max value for GAIN parameter from ±1 to ±3 and added E _{IR} (Internal voltage reference error) parameter.
		Additional minor data sheet clarifications and corrections.
April 2009	5367	■ Added JTAG/SWD clarification (see "Communication with JTAG/SWD" on page 162).
		Added clarification that the PLL operates at 400 MHz, but is divided by two prior to the application of the output divisor.
		■ Added "GPIO Module DC Characteristics" table (see Table 20-4 on page 646).
		Additional minor data sheet clarifications and corrections.
January 2009	4660	■ Corrected bit type for RELOAD bit field in SysTick Reload Value register; changed to R/W.
		■ Clarification added as to what happens when the SSI in slave mode is required to transmit but there is no data in the TX FIFO.
		■ Added "Hardware Configuration" section to Ethernet Controller chapter.
		Additional minor data sheet clarifications and corrections.
November 2008	4283	■ Revised High-Level Block Diagram.
		Additional minor data sheet clarifications and corrections were made.
	<u> </u>	

Table 1. Revision History (continued)

Date	Revision	Description
October 2008	4149	■ Corrected values for DSOSCSRC bit field in Deep Sleep Clock Configuration (DSLPCLKCFG) register.
		■ The FMA value for the FMPRE3 register was incorrect in the Flash Resident Registers table in the Internal Memory chapter. The correct value is 0x0000.0006.
		In the Ethernet chapter, major improvements were made including a rewrite of the conceptual information and the addition of new figures to clarify how to use the Ethernet Controller interface.
		■ Incorrect Comparator Operating Modes tables were removed from the Analog Comparators chapter.
August 2008	3447	■ Added note on clearing interrupts to Interrupts chapter.
		■ Added Power Architecture diagram to System Control chapter.
		Additional minor data sheet clarifications and corrections.
July 2008	3108	■ Corrected resistor value in ERBIAS signal description.
		Additional minor data sheet clarifications and corrections.
May 2008	2972	■ The 108-Ball BGA pin diagram and pin tables had an error. The following signals were erroneously indicated as available and have now been changed to a No Connect (NC):
		- Ball C1: Changed ₽E7 to NC
		- Ball C2: Changed ₽E6 to NC
		- Ball D2: Changed ₽E5 to NC
		■ As noted in the PCN, three of the nine Ethernet LED configuration options are no longer supported: TX Activity (0x2), RX Activity (0x3), and Collision (0x4). These values for the LED0 and LED1 bit fields in the MR23 register are now marked as reserved.
		■ As noted in the PCN, the option to provide VDD25 power from external sources was removed. Use the LDO output as the source of VDD25 input.
		■ As noted in the PCN, pin 41 (ball K3 on the BGA package) was renamed from GNDPHY to ERBIAS. A 12.4-kΩ resistor should be connected between ERBIAS and ground to accommodate future device revisions (see "Functional Description" on page 521).
		■ Additional minor data sheet clarifications and corrections.
April 2008	2881	■ The Θ_{JA} value was changed from 55.3 to 34 in the "Thermal Characteristics" table in the Operating Characteristics chapter.
		■ Bit 31 of the DC3 register was incorrectly described in prior versions of the data sheet. A reset of 1 indicates that an even CCP pin is present and can be used as a 32-KHz input clock.
		■ Values for I _{DD_HIBERNATE} were added to the "Detailed Power Specifications" table in the "Electrical Characteristics" chapter.
		■ The "Hibernation Module DC Electricals" table was added to the "Electrical Characteristics" chapter
		■ The maximum value on Core supply voltage (V _{DD25}) in the "Maximum Ratings" table in the "Electrica Characteristics" chapter was changed from 4 to 3.
		■ The operational frequency of the internal 30-kHz oscillator clock source is 30 kHz ± 50% (prior data sheets incorrectly noted it as 30 kHz ± 30%).
		A value of 0x3 in bits 5:4 of the MISC register (OSCSRC) indicates the 30-KHz internal oscillator is the input source for the oscillator. Prior data sheets incorrectly noted 0x3 as a reserved value.

Table 1. Revision History (continued)

Date	Revision	Description
		■ The reset for bits 6:4 of the RCC2 register (OSCSRC2) is 0x1 (IOSC). Prior data sheets incorrectly noted the reset was 0x0 (MOSC).
		A note on high-current applications was added to the GPIO chapter:
		For special high-current applications, the GPIO output buffers may be used with the following restrictions. With the GPIO pins configured as 8-mA output drivers, a total of four GPIO outputs may be used to sink current loads up to 18 mA each. At 18-mA sink current loading, the VOL value is specified as 1.2 V. The high-current GPIO package pins must be selected such that there are only a maximum of two per side of the physical package or BGA pin group with the total number of high-current GPIO outputs not exceeding four for the entire package.
		■ A note on Schmitt inputs was added to the GPIO chapter:
		Pins configured as digital inputs are Schmitt-triggered.
		■ The Buffer type on the WAKE pin changed from OD to - in the Signal Tables.
		■ The "Differential Sampling Range" figures in the ADC chapter were clarified.
		■ The last revision of the data sheet (revision 2550) introduced two errors that have now been corrected:
		 The LQFP pin diagrams and pin tables were missing the comparator positive and negative input pins.
		The base address was listed incorrectly in the FMPRE0 and FMPPE0 register bit diagrams.
		Additional minor data sheet clarifications and corrections.
March 2008	2550	Started tracking revision history.

About This Document

This data sheet provides reference information for the LM3S6432 microcontroller, describing the functional blocks of the system-on-chip (SoC) device designed around the ARM® Cortex™-M3 core.

Audience

This manual is intended for system software developers, hardware designers, and application developers.

About This Manual

This document is organized into sections that correspond to each major feature.

Related Documents

The following related documents are available on the Stellaris® web site at www.ti.com/stellaris:

- Stellaris® Errata
- ARM® Cortex™-M3 Errata
- Cortex™-M3/M4 Instruction Set Technical User's Manual
- Stellaris® Graphics Library User's Guide
- Stellaris® Peripheral Driver Library User's Guide

The following related documents are also referenced:

- ARM® Debug Interface V5 Architecture Specification
- ARM® Embedded Trace Macrocell Architecture Specification
- IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture

This documentation list was current as of publication date. Please check the web site for additional documentation, including application notes and white papers.

Documentation Conventions

This document uses the conventions shown in Table 2 on page 31.

Table 2. Documentation Conventions

Notation	Meaning			
General Register Notation				
REGISTER	APB registers are indicated in uppercase bold. For example, PBORCTL is the Power-On and Brown-Out Reset Control register. If a register name contains a lowercase n, it represents more than one register. For example, SRCRn represents any (or all) of the three Software Reset Control registers: SRCR0 , SRCR1 , and SRCR2 .			
bit	A single bit in a register.			
bit field	Two or more consecutive and related bits.			
offset 0xnnn	A hexadecimal increment to a register's address, relative to that module's base address as specified in Table 2-4 on page 68.			
Register N	Registers are numbered consecutively throughout the document to aid in referencing them. The register number has no meaning to software.			
reserved	Register bits marked <i>reserved</i> are reserved for future use. In most cases, reserved bits are set to 0; however, user software should not rely on the value of a reserved bit. To provide software compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.			
уу:хх	The range of register bits inclusive from xx to yy. For example, 31:15 means bits 15 through 31 in that register.			
Register Bit/Field Types	This value in the register bit diagram indicates whether software running on the controller can change the value of the bit field.			
RC	Software can read this field. The bit or field is cleared by hardware after reading the bit/field.			
RO	Software can read this field. Always write the chip reset value.			
R/W	Software can read or write this field.			
R/WC	Software can read or write this field. Writing to it with any value clears the register.			
R/W1C	Software can read or write this field. A write of a 0 to a W1C bit does not affect the bit value in the register. A write of a 1 clears the value of the bit in the register; the remaining bits remain unchanged.			
	This register type is primarily used for clearing interrupt status bits where the read operation provides the interrupt status and the write of the read value clears only the interrupts being reported at the time the register was read.			
R/W1S	Software can read or write a 1 to this field. A write of a 0 to a R/W1S bit does not affect the bit value in the register.			
W1C	Software can write this field. A write of a 0 to a W1C bit does not affect the bit value in the register. A write of a 1 clears the value of the bit in the register; the remaining bits remain unchanged. A read of the register returns no meaningful data.			
	This register is typically used to clear the corresponding bit in an interrupt register.			
WO	Only a write by software is valid; a read of the register returns no meaningful data.			
Register Bit/Field Reset Value	This value in the register bit diagram shows the bit/field value after any reset, unless noted.			
0	Bit cleared to 0 on chip reset.			
1	Bit set to 1 on chip reset.			
-	Nondeterministic.			
Pin/Signal Notation				
[]	Pin alternate function; a pin defaults to the signal without the brackets.			
pin	Refers to the physical connection on the package.			
signal	Refers to the electrical signal encoding of a pin.			

Table 2. Documentation Conventions (continued)

Notation	Meaning
assert a signal	Change the value of the signal from the logically False state to the logically True state. For active High signals, the asserted signal value is 1 (High); for active Low signals, the asserted signal value is 0 (Low). The active polarity (High or Low) is defined by the signal name (see SIGNAL and SIGNAL below).
deassert a signal	Change the value of the signal from the logically True state to the logically False state.
SIGNAL	Signal names are in uppercase and in the Courier font. An overbar on a signal name indicates that it is active Low. To assert SIGNAL is to drive it Low; to deassert SIGNAL is to drive it High.
SIGNAL	Signal names are in uppercase and in the Courier font. An active High signal has no overbar. To assert SIGNAL is to drive it High; to deassert SIGNAL is to drive it Low.
Numbers	
Х	An uppercase X indicates any of several values is allowed, where X can be any legal pattern. For example, a binary value of 0X00 can be either 0100 or 0000, a hex value of 0xX is 0x0 or 0x1, and so on.
0x	Hexadecimal numbers have a prefix of 0x. For example, 0x00FF is the hexadecimal number FF.
	All other numbers within register tables are assumed to be binary. Within conceptual information, binary numbers are indicated with a b suffix, for example, 1011b, and decimal numbers are written without a prefix or suffix.

1 Architectural Overview

The Stellaris[®] family of microcontrollers—the first ARM® Cortex[™]-M3 based controllers—brings high-performance 32-bit computing to cost-sensitive embedded microcontroller applications. These pioneering parts deliver customers 32-bit performance at a cost equivalent to legacy 8- and 16-bit devices, all in a package with a small footprint.

The Stellaris family offers efficient performance and extensive integration, favorably positioning the device into cost-conscious applications requiring significant control-processing and connectivity capabilities. The Stellaris LM3S6000 series combines both a 10/100 Ethernet Media Access Control (MAC) and Physical (PHY) layer, marking the first time that integrated connectivity is available with an ARM Cortex-M3 MCU and the only integrated 10/100 Ethernet MAC and PHY available in an ARM architecture MCU.

The LM3S6432 microcontroller is targeted for industrial applications, including remote monitoring, electronic point-of-sale machines, test and measurement equipment, network appliances and switches, factory automation, HVAC and building control, gaming equipment, motion control, medical instrumentation, and fire and security.

In addition, the LM3S6432 microcontroller offers the advantages of ARM's widely available development tools, System-on-Chip (SoC) infrastructure IP applications, and a large user community. Additionally, the microcontroller uses ARM's Thumb®-compatible Thumb-2 instruction set to reduce memory requirements and, thereby, cost. Finally, the LM3S6432 microcontroller is code-compatible to all members of the extensive Stellaris family; providing flexibility to fit our customers' precise needs.

Texas Instruments offers a complete solution to get to market quickly, with evaluation and development boards, white papers and application notes, an easy-to-use peripheral driver library, and a strong support, sales, and distributor network. See "Ordering and Contact Information" on page 687 for ordering information for Stellaris family devices.

1.1 Product Features

The LM3S6432 microcontroller includes the following product features:

- 32-Bit RISC Performance
 - 32-bit ARM® Cortex™-M3 v7M architecture optimized for small-footprint embedded applications
 - System timer (SysTick), providing a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism
 - Thumb®-compatible Thumb-2-only instruction set processor core for high code density
 - 50-MHz operation
 - Hardware-division and single-cycle-multiplication
 - Integrated Nested Vectored Interrupt Controller (NVIC) providing deterministic interrupt handling
 - 29 interrupts with eight priority levels

- Memory protection unit (MPU), providing a privileged mode for protected operating system functionality
- Unaligned data access, enabling data to be efficiently packed into memory
- Atomic bit manipulation (bit-banding), delivering maximum memory utilization and streamlined peripheral control
- ARM® Cortex™-M3 Processor Core
 - Compact core.
 - Thumb-2 instruction set, delivering the high-performance expected of an ARM core in the memory size usually associated with 8- and 16-bit devices; typically in the range of a few kilobytes of memory for microcontroller class applications.
 - Rapid application execution through Harvard architecture characterized by separate buses for instruction and data.
 - Exceptional interrupt handling, by implementing the register manipulations required for handling an interrupt in hardware.
 - Deterministic, fast interrupt processing: always 12 cycles, or just 6 cycles with tail-chaining
 - Memory protection unit (MPU) to provide a privileged mode of operation for complex applications.
 - Migration from the ARM7™ processor family for better performance and power efficiency.
 - Full-featured debug solution
 - Serial Wire JTAG Debug Port (SWJ-DP)
 - Flash Patch and Breakpoint (FPB) unit for implementing breakpoints
 - Data Watchpoint and Trigger (DWT) unit for implementing watchpoints, trigger resources, and system profiling
 - Instrumentation Trace Macrocell (ITM) for support of printf style debugging
 - Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer
 - Optimized for single-cycle flash usage
 - Three sleep modes with clock gating for low power
 - Single-cycle multiply instruction and hardware divide
 - Atomic operations
 - ARM Thumb2 mixed 16-/32-bit instruction set
 - 1.25 DMIPS/MHz
- JTAG

- IEEE 1149.1-1990 compatible Test Access Port (TAP) controller
- Four-bit Instruction Register (IR) chain for storing JTAG instructions
- IEEE standard instructions: BYPASS, IDCODE, SAMPLE/PRELOAD, EXTEST and INTEST
- ARM additional instructions: APACC, DPACC and ABORT
- Integrated ARM Serial Wire Debug (SWD)

Internal Memory

- 96 KB single-cycle flash
 - User-managed flash block protection on a 2-KB block basis
 - · User-managed flash data programming
 - User-defined and managed flash-protection block
- 32 KB single-cycle SRAM

GPIOs

- 14-43 GPIOs, depending on configuration
- 5-V-tolerant in input configuration
- Fast toggle capable of a change every two clock cycles
- Programmable control for GPIO interrupts
 - · Interrupt generation masking
 - · Edge-triggered on rising, falling, or both
 - · Level-sensitive on High or Low values
- Bit masking in both read and write operations through address lines
- Can initiate an ADC sample sequence
- Pins configured as digital inputs are Schmitt-triggered.
- Programmable control for GPIO pad configuration
 - · Weak pull-up or pull-down resistors
 - 2-mA, 4-mA, and 8-mA pad drive for digital communication; up to four pads can be configured with an 18-mA pad drive for high-current applications
 - · Slew rate control for the 8-mA drive
 - Open drain enables
 - Digital input enables

■ General-Purpose Timers

- Three General-Purpose Timer Modules (GPTM), each of which provides two 16-bit timers/counters. Each GPTM can be configured to operate independently:
 - As a single 32-bit timer
 - As one 32-bit Real-Time Clock (RTC) to event capture
 - For Pulse Width Modulation (PWM)
 - To trigger analog-to-digital conversions
- 32-bit Timer modes
 - Programmable one-shot timer
 - Programmable periodic timer
 - Real-Time Clock when using an external 32.768-KHz clock as the input
 - User-enabled stalling when the controller asserts CPU Halt flag during debug
 - ADC event trigger
- 16-bit Timer modes
 - General-purpose timer function with an 8-bit prescaler (for one-shot and periodic modes only)
 - Programmable one-shot timer
 - · Programmable periodic timer
 - User-enabled stalling when the controller asserts CPU Halt flag during debug
 - ADC event trigger
- 16-bit Input Capture modes
 - Input edge count capture
 - Input edge time capture
- 16-bit PWM mode
 - Simple PWM mode with software-programmable output inversion of the PWM signal
- ARM FiRM-compliant Watchdog Timer
 - 32-bit down counter with a programmable load register
 - Separate watchdog clock with an enable
 - Programmable interrupt generation logic with interrupt masking
 - Lock register protection from runaway software

- Reset generation logic with an enable/disable
- User-enabled stalling when the controller asserts the CPU Halt flag during debug

ADC

- Three analog input channels
- Single-ended and differential-input configurations
- On-chip internal temperature sensor
- Sample rate of 250 thousand samples/second
- Flexible, configurable analog-to-digital conversion
- Four programmable sample conversion sequences from one to eight entries long, with corresponding conversion result FIFOs
- Flexible trigger control
 - Controller (software)
 - Timers
 - Analog Comparators
 - PWM
 - GPIO
- Hardware averaging of up to 64 samples for improved accuracy
- Converter uses an internal 3-V reference
- Power and ground for the analog circuitry is separate from the digital power and ground

■ UART

- Two fully programmable 16C550-type UARTs with IrDA support
- Separate 16x8 transmit (TX) and receive (RX) FIFOs to reduce CPU interrupt service loading
- Programmable baud-rate generator allowing speeds up to 3.125 Mbps
- Programmable FIFO length, including 1-byte deep operation providing conventional double-buffered interface
- FIFO trigger levels of 1/8, 1/4, 1/2, 3/4, and 7/8
- Standard asynchronous communication bits for start, stop, and parity
- Line-break generation and detection
- Fully programmable serial interface characteristics
 - 5, 6, 7, or 8 data bits

- Even, odd, stick, or no-parity bit generation/detection
- 1 or 2 stop bit generation
- IrDA serial-IR (SIR) encoder/decoder providing
 - Programmable use of IrDA Serial Infrared (SIR) or UART input/output
 - Support of IrDA SIR encoder/decoder functions for data rates up to 115.2 Kbps half-duplex
 - Support of normal 3/16 and low-power (1.41-2.23 μs) bit durations
 - Programmable internal clock generator enabling division of reference clock by 1 to 256 for low-power mode bit duration
- Synchronous Serial Interface (SSI)
 - Master or slave operation
 - Programmable clock bit rate and prescale
 - Separate transmit and receive FIFOs, 16 bits wide, 8 locations deep
 - Programmable interface operation for Freescale SPI, MICROWIRE, or Texas Instruments synchronous serial interfaces
 - Programmable data frame size from 4 to 16 bits
 - Internal loopback test mode for diagnostic/debug testing

■ I²C

- Devices on the I²C bus can be designated as either a master or a slave
 - Supports both sending and receiving data as either a master or a slave
 - Supports simultaneous master and slave operation
- Four I²C modes
 - Master transmit
 - Master receive
 - · Slave transmit
 - Slave receive
- Two transmission speeds: Standard (100 Kbps) and Fast (400 Kbps)
- Master and slave interrupt generation
 - Master generates interrupts when a transmit or receive operation completes (or aborts due to an error)
 - · Slave generates interrupts when data has been sent or requested by a master

- Master with arbitration and clock synchronization, multimaster support, and 7-bit addressing mode
- 10/100 Ethernet Controller
 - Conforms to the IEEE 802.3-2002 specification
 - 10BASE-T/100BASE-TX IEEE-802.3 compliant. Requires only a dual 1:1 isolation transformer interface to the line
 - 10BASE-T/100BASE-TX ENDEC, 100BASE-TX scrambler/descrambler
 - Full-featured auto-negotiation
 - Multiple operational modes
 - Full- and half-duplex 100 Mbps
 - Full- and half-duplex 10 Mbps
 - · Power-saving and power-down modes
 - Highly configurable
 - · Programmable MAC address
 - · LED activity selection
 - Promiscuous mode support
 - CRC error-rejection control
 - · User-configurable interrupts
 - Physical media manipulation
 - Automatic MDI/MDI-X cross-over correction
 - Register-programmable transmit amplitude
 - Automatic polarity correction and 10BASE-T signal reception
- Analog Comparators
 - Two independent integrated analog comparators
 - Configurable for output to drive an output pin, generate an interrupt, or initiate an ADC sample sequence
 - Compare external pin input to external pin input or to internal programmable voltage reference
 - Compare a test voltage against any one of these voltages
 - · An individual external reference voltage
 - A shared single external reference voltage

· A shared internal reference voltage

PWM

- One PWM generator blocks, each with one 16-bit counter, two PWM comparators, a PWM signal generator, a dead-band generator, and an interrupt/ADC-trigger selector
- One fault input in hardware to promote low-latency shutdown
- One 16-bit counter
 - Runs in Down or Up/Down mode
 - Output frequency controlled by a 16-bit load value
 - Load value updates can be synchronized
 - Produces output signals at zero and load value
- Two PWM comparators
 - Comparator value updates can be synchronized
 - · Produces output signals on match
- PWM generator
 - Output PWM signal is constructed based on actions taken as a result of the counter and PWM comparator output signals
 - Produces two independent PWM signals
- Dead-band generator
 - Produces two PWM signals with programmable dead-band delays suitable for driving a half-H bridge
 - · Can be bypassed, leaving input PWM signals unmodified
- Flexible output control block with PWM output enable of each PWM signal
 - · PWM output enable of each PWM signal
 - Optional output inversion of each PWM signal (polarity control)
 - Optional fault handling for each PWM signal
 - Synchronization of timers in the PWM generator blocks
 - Interrupt status summary of the PWM generator blocks
- Can initiate an ADC sample sequence

Power

 On-chip Low Drop-Out (LDO) voltage regulator, with programmable output user-adjustable from 2.25 V to 2.75 V

- Low-power options on controller: Sleep and Deep-sleep modes
- Low-power options for peripherals: software controls shutdown of individual peripherals
- 3.3-V supply brown-out detection and reporting via interrupt or reset
- Flexible Reset Sources
 - Power-on reset (POR)
 - Reset pin assertion
 - Brown-out (BOR) detector alerts to system power drops
 - Software reset
 - Watchdog timer reset
 - Internal low drop-out (LDO) regulator output goes unregulated
- Industrial and extended temperature 100-pin RoHS-compliant LQFP package
- Industrial-range 108-ball RoHS-compliant BGA package

1.2 Target Applications

- Remote monitoring
- Electronic point-of-sale (POS) machines
- Test and measurement equipment
- Network appliances and switches
- Factory automation
- HVAC and building control
- Gaming equipment
- Motion control
- Medical instrumentation
- Fire and security
- Power and energy
- Transportation

1.3 High-Level Block Diagram

Figure 1-1 on page 42 depicts the features on the Stellaris LM3S6432 microcontroller.

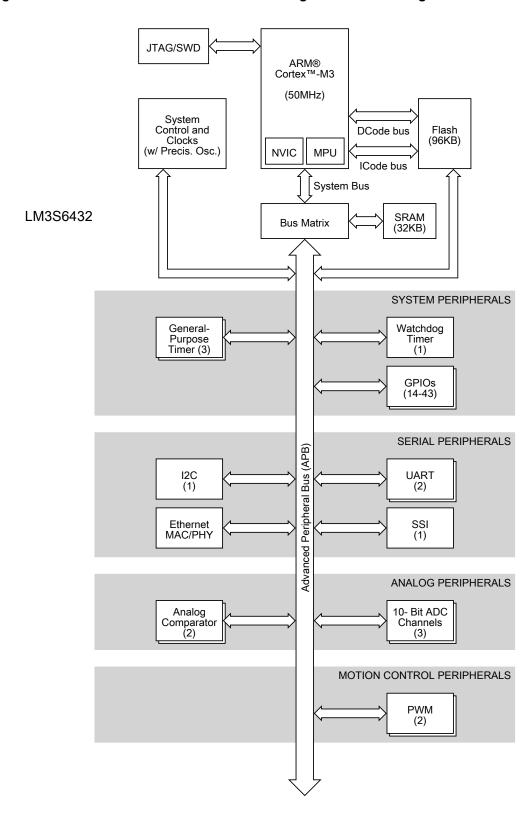


Figure 1-1. Stellaris LM3S6432 Microcontroller High-Level Block Diagram

1.4 Functional Overview

The following sections provide an overview of the features of the LM3S6432 microcontroller. The page number in parenthesis indicates where that feature is discussed in detail. Ordering and support information can be found in "Ordering and Contact Information" on page 687.

1.4.1 ARM Cortex™-M3

1.4.1.1 Processor Core (see page 49)

All members of the Stellaris product family, including the LM3S6432 microcontroller, are designed around an ARM Cortex™-M3 processor core. The ARM Cortex-M3 processor provides the core for a high-performance, low-cost platform that meets the needs of minimal memory implementation, reduced pin count, and low-power consumption, while delivering outstanding computational performance and exceptional system response to interrupts.

1.4.1.2 **Memory Map** (see page 68)

A memory map lists the location of instructions and data in memory. The memory map for the LM3S6432 controller can be found in Table 2-4 on page 68. Register addresses are given as a hexadecimal increment, relative to the module's base address as shown in the memory map.

1.4.1.3 System Timer (SysTick) (see page 91)

Cortex-M3 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:

- An RTOS tick timer which fires at a programmable rate (for example, 100 Hz) and invokes a SysTick routine.
- A high-speed alarm timer using the system clock.
- A variable rate alarm or signal timer—the duration is range-dependent on the reference clock used and the dynamic range of the counter.
- A simple counter. Software can use this to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

1.4.1.4 Nested Vectored Interrupt Controller (NVIC) (see page 92)

The LM3S6432 controller includes the ARM Nested Vectored Interrupt Controller (NVIC) on the ARM® Cortex™-M3 core. The NVIC and Cortex-M3 prioritize and handle all exceptions. All exceptions are handled in Handler Mode. The processor state is automatically stored to the stack on an exception, and automatically restored from the stack at the end of the Interrupt Service Routine (ISR). The vector is fetched in parallel to the state saving, which enables efficient interrupt entry. The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoration. Software can set eight priority levels on 7 exceptions (system handlers) and 29 interrupts.

1.4.1.5 System Control Block (SCB) (see page 94)

The SCB provides system implementation information and system control, including configuration, control, and reporting of system exceptions.

1.4.1.6 Memory Protection Unit (MPU) (see page 94)

The MPU supports the standard ARMv7 Protected Memory System Architecture (PMSA) model. The MPU provides full support for protection regions, overlapping protection regions, access permissions, and exporting memory attributes to the system.

1.4.2 Motor Control Peripherals

To enhance motor control, the LM3S6432 controller features Pulse Width Modulation (PWM) outputs.

1.4.2.1 PWM

Pulse width modulation (PWM) is a powerful technique for digitally encoding analog signal levels. High-resolution counters are used to generate a square wave, and the duty cycle of the square wave is modulated to encode an analog signal. Typical applications include switching power supplies and motor control.

On the LM3S6432, PWM motion control functionality can be achieved through:

- Dedicated, flexible motion control hardware using the PWM pins
- The motion control features of the general-purpose timers using the CCP pins

PWM Pins (see page 578)

The LM3S6432 PWM module consists of one PWM generator blocks and a control block. Each PWM generator block contains one timer (16-bit down or up/down counter), two comparators, a PWM signal generator, a dead-band generator, and an interrupt/ADC-trigger selector. The control block determines the polarity of the PWM signals, and which signals are passed through to the pins.

Each PWM generator block produces two PWM signals that can either be independent signals or a single pair of complementary signals with dead-band delays inserted. The output of the PWM generation blocks are managed by the output control block before being passed to the device pins.

CCP Pins (see page 311)

The General-Purpose Timer Module's CCP (Capture Compare PWM) pins are software programmable to support a simple PWM mode with a software-programmable output inversion of the PWM signal.

Fault Pin (see page 583)

The LM3S6432 PWM module includes one fault-condition handling input to quickly provide low-latency shutdown and prevent damage to the motor being controlled.

1.4.3 Analog Peripherals

To handle analog signals, the LM3S6432 microcontroller offers an Analog-to-Digital Converter (ADC).

For support of analog signals, the LM3S6432 microcontroller offers two analog comparators.

1.4.3.1 ADC (see page 365)

An analog-to-digital converter (ADC) is a peripheral that converts a continuous analog voltage to a discrete digital number.

The LM3S6432 ADC module features 10-bit conversion resolution and supports three input channels, plus an internal temperature sensor. Four buffered sample sequences allow rapid sampling of up to eight analog input sources without controller intervention. Each sample sequence provides flexible programming with fully configurable input source, trigger events, interrupt generation, and sequence priority.

1.4.3.2 Analog Comparators (see page 566)

An analog comparator is a peripheral that compares two analog voltages, and provides a logical output that signals the comparison result.

The LM3S6432 microcontroller provides two independent integrated analog comparators that can be configured to drive an output or generate an interrupt or ADC event.

A comparator can compare a test voltage against any one of these voltages:

- An individual external reference voltage
- A shared single external reference voltage
- A shared internal reference voltage

The comparator can provide its output to a device pin, acting as a replacement for an analog comparator on the board, or it can be used to signal the application via interrupts or triggers to the ADC to cause it to start capturing a sample sequence. The interrupt generation and ADC triggering logic is separate. This means, for example, that an interrupt can be generated on a rising edge and the ADC triggered on a falling edge.

1.4.4 Serial Communications Peripherals

The LM3S6432 controller supports both asynchronous and synchronous serial communications with:

- Two fully programmable 16C550-type UARTs
- One SSI module
- One I²C module
- Ethernet controller

1.4.4.1 **UART** (see page 401)

A Universal Asynchronous Receiver/Transmitter (UART) is an integrated circuit used for RS-232C serial communications, containing a transmitter (parallel-to-serial converter) and a receiver (serial-to-parallel converter), each clocked separately.

The LM3S6432 controller includes two fully programmable 16C550-type UARTs that support data transfer speeds up to 3.125 Mbps. (Although similar in functionality to a 16C550 UART, it is not register-compatible.) In addition, each UART is capable of supporting IrDA.

Separate 16x8 transmit (TX) and receive (RX) FIFOs reduce CPU interrupt service loading. The UART can generate individually masked interrupts from the RX, TX, modem status, and error

conditions. The module provides a single combined interrupt when any of the interrupts are asserted and are unmasked.

1.4.4.2 SSI (see page 443)

Synchronous Serial Interface (SSI) is a four-wire bi-directional full and low-speed communications interface.

The LM3S6432 controller includes one SSI module that provides the functionality for synchronous serial communications with peripheral devices, and can be configured to use the Freescale SPI, MICROWIRE, or TI synchronous serial interface frame formats. The size of the data frame is also configurable, and can be set between 4 and 16 bits, inclusive.

The SSI module performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data transmitted to a peripheral device. The TX and RX paths are buffered with internal FIFOs, allowing up to eight 16-bit values to be stored independently.

The SSI module can be configured as either a master or slave device. As a slave device, the SSI module can also be configured to disable its output, which allows a master device to be coupled with multiple slave devices.

The SSI module also includes a programmable bit rate clock divider and prescaler to generate the output serial clock derived from the SSI module's input clock. Bit rates are generated based on the input clock and the maximum bit rate is determined by the connected peripheral.

1.4.4.3 I²C (see page 481)

The Inter-Integrated Circuit (I²C) bus provides bi-directional data transfer through a two-wire design (a serial data line SDA and a serial clock line SCL).

The I²C bus interfaces to external I²C devices such as serial memory (RAMs and ROMs), networking devices, LCDs, tone generators, and so on. The I²C bus may also be used for system testing and diagnostic purposes in product development and manufacture.

The LM3S6432 controller includes one I^2C module that provides the ability to communicate to other IC devices over an I^2C bus. The I^2C bus supports devices that can both transmit and receive (write and read) data.

Devices on the I²C bus can be designated as either a master or a slave. The I²C module supports both sending and receiving data as either a master or a slave, and also supports the simultaneous operation as both a master and a slave. The four I²C modes are: Master Transmit, Master Receive, Slave Transmit, and Slave Receive.

A Stellaris I²C module can operate at two speeds: Standard (100 Kbps) and Fast (400 Kbps).

Both the I²C master and slave can generate interrupts. The I²C master generates interrupts when a transmit or receive operation completes (or aborts due to an error). The I²C slave generates interrupts when data has been sent or requested by a master.

1.4.4.4 Ethernet Controller (see page 518)

Ethernet is a frame-based computer networking technology for local area networks (LANs). Ethernet has been standardized as IEEE 802.3. It defines a number of wiring and signaling standards for the physical layer, two means of network access at the Media Access Control (MAC)/Data Link Layer, and a common addressing format.

The Stellaris® Ethernet Controller consists of a fully integrated media access controller (MAC) and network physical (PHY) interface device. The Ethernet Controller conforms to IEEE 802.3

specifications and fully supports 10BASE-T and 100BASE-TX standards. In addition, the Ethernet Controller supports automatic MDI/MDI-X cross-over correction.

1.4.5 System Peripherals

1.4.5.1 Programmable GPIOs (see page 258)

General-purpose input/output (GPIO) pins offer flexibility for a variety of connections.

The Stellaris GPIO module is comprised of seven physical GPIO blocks, each corresponding to an individual GPIO port. The GPIO module is FiRM-compliant (compliant to the ARM Foundation IP for Real-Time Microcontrollers specification) and supports 14-43 programmable input/output pins. The number of GPIOs available depends on the peripherals being used (see "Signal Tables" on page 618 for the signals available to each GPIO pin).

The GPIO module features programmable interrupt generation as either edge-triggered or level-sensitive on all pins, programmable control for GPIO pad configuration, and bit masking in both read and write operations through address lines. Pins configured as digital inputs are Schmitt-triggered.

1.4.5.2 Three Programmable Timers (see page 305)

Programmable timers can be used to count or time external events that drive the Timer input pins.

The Stellaris General-Purpose Timer Module (GPTM) contains three GPTM blocks. Each GPTM block provides two 16-bit timers/counters that can be configured to operate independently as timers or event counters, or configured to operate as one 32-bit timer or one 32-bit Real-Time Clock (RTC). Timers can also be used to trigger analog-to-digital (ADC) conversions.

When configured in 32-bit mode, a timer can run as a Real-Time Clock (RTC), one-shot timer or periodic timer. When in 16-bit mode, a timer can run as a one-shot timer or periodic timer, and can extend its precision by using an 8-bit prescaler. A 16-bit timer can also be configured for event capture or Pulse Width Modulation (PWM) generation.

1.4.5.3 Watchdog Timer (see page 341)

A watchdog timer can generate an interrupt or a reset when a time-out value is reached. The watchdog timer is used to regain control when a system has failed due to a software error or to the failure of an external device to respond in the expected way.

The Stellaris Watchdog Timer module consists of a 32-bit down counter, a programmable load register, interrupt generation logic, and a locking register.

The Watchdog Timer can be configured to generate an interrupt to the controller on its first time-out, and to generate a reset signal on its second time-out. Once the Watchdog Timer has been configured, the lock register can be written to prevent the timer configuration from being inadvertently altered.

1.4.6 Memory Peripherals

The LM3S6432 controller offers both single-cycle SRAM and single-cycle Flash memory.

1.4.6.1 SRAM (see page 231)

The LM3S6432 static random access memory (SRAM) controller supports 32 KB SRAM. The internal SRAM of the Stellaris devices starts at base address 0x2000.0000 of the device memory map. To reduce the number of time-consuming read-modify-write (RMW) operations, ARM has introduced bit-banding technology in the new Cortex-M3 processor. With a bit-band-enabled processor, certain

regions in the memory map (SRAM and peripheral space) can use address aliases to access individual bits in a single, atomic operation.

1.4.6.2 Flash (see page 232)

The LM3S6432 Flash controller supports 96 KB of flash memory. The flash is organized as a set of 1-KB blocks that can be individually erased. Erasing a block causes the entire contents of the block to be reset to all 1s. These blocks are paired into a set of 2-KB blocks that can be individually protected. The blocks can be marked as read-only or execute-only, providing different levels of code protection. Read-only blocks cannot be erased or programmed, protecting the contents of those blocks from being modified. Execute-only blocks cannot be erased or programmed, and can only be read by the controller instruction fetch mechanism, protecting the contents of those blocks from being read by either the controller or by a debugger.

1.4.7 Additional Features

1.4.7.1 JTAG TAP Controller (see page 155)

The Joint Test Action Group (JTAG) port is an IEEE standard that defines a Test Access Port and Boundary Scan Architecture for digital integrated circuits and provides a standardized serial interface for controlling the associated test logic. The TAP, Instruction Register (IR), and Data Registers (DR) can be used to test the interconnections of assembled printed circuit boards and obtain manufacturing information on the components. The JTAG Port also provides a means of accessing and controlling design-for-test features such as I/O pin observation and control, scan testing, and debugging.

The JTAG port is composed of the standard five pins: TRST, TCK, TMS, TDI, and TDO. Data is transmitted serially into the controller on TDI and out of the controller on TDO. The interpretation of this data is dependent on the current state of the TAP controller. For detailed information on the operation of the JTAG port and TAP controller, please refer to the IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture.

The Stellaris JTAG controller works with the ARM JTAG controller built into the Cortex-M3 core. This is implemented by multiplexing the $\tt TDO$ outputs from both JTAG controllers. ARM JTAG instructions select the ARM $\tt TDO$ output while Stellaris JTAG instructions select the Stellaris $\tt TDO$ outputs. The multiplexer is controlled by the Stellaris JTAG controller, which has comprehensive programming for the ARM, Stellaris, and unimplemented JTAG instructions.

1.4.7.2 System Control and Clocks (see page 168)

System control determines the overall operation of the device. It provides information about the device, controls the clocking of the device and individual peripherals, and handles reset detection and reporting.

1.4.8 Hardware Details

Details on the pins and package can be found in the following sections:

- "Pin Diagram" on page 616
- "Signal Tables" on page 618
- "Operating Characteristics" on page 644
- "Electrical Characteristics" on page 645
- "Package Information" on page 689

2 The Cortex-M3 Processor

The ARM® Cortex[™]-M3 processor provides a high-performance, low-cost platform that meets the system requirements of minimal memory implementation, reduced pin count, and low power consumption, while delivering outstanding computational performance and exceptional system response to interrupts. Features include:

- Compact core.
- Thumb-2 instruction set, delivering the high-performance expected of an ARM core in the memory size usually associated with 8- and 16-bit devices; typically in the range of a few kilobytes of memory for microcontroller class applications.
- Rapid application execution through Harvard architecture characterized by separate buses for instruction and data.
- Exceptional interrupt handling, by implementing the register manipulations required for handling an interrupt in hardware.
- Deterministic, fast interrupt processing: always 12 cycles, or just 6 cycles with tail-chaining
- Memory protection unit (MPU) to provide a privileged mode of operation for complex applications.
- Migration from the ARM7[™] processor family for better performance and power efficiency.
- Full-featured debug solution
 - Serial Wire JTAG Debug Port (SWJ-DP)
 - Flash Patch and Breakpoint (FPB) unit for implementing breakpoints
 - Data Watchpoint and Trigger (DWT) unit for implementing watchpoints, trigger resources, and system profiling
 - Instrumentation Trace Macrocell (ITM) for support of printf style debugging
 - Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer
- Optimized for single-cycle flash usage
- Three sleep modes with clock gating for low power
- Single-cycle multiply instruction and hardware divide
- Atomic operations
- ARM Thumb2 mixed 16-/32-bit instruction set
- 1.25 DMIPS/MHz

The Stellaris[®] family of microcontrollers builds on this core to bring high-performance 32-bit computing to cost-sensitive embedded microcontroller applications, such as factory automation and control, industrial control power devices, building and home automation, and stepper motor control.

This chapter provides information on the Stellaris implementation of the Cortex-M3 processor, including the programming model, the memory model, the exception model, fault handling, and power management.

For technical details on the instruction set, see the *Cortex*™-*M3/M4 Instruction Set Technical User's Manual*.

2.1 Block Diagram

The Cortex-M3 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M3 processor implements tightly coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M3 processor implements a version of the Thumb® instruction set based on Thumb-2 technology, ensuring high code density and reduced program memory requirements. The Cortex-M3 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M3 processor closely integrates a nested interrupt controller (NVIC), to deliver industry-leading interrupt performance. The Stellaris NVIC includes a non-maskable interrupt (NMI) and provides eight interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of interrupt service routines (ISRs), dramatically reducing interrupt latency. The hardware stacking of registers and the ability to suspend load-multiple and store-multiple operations further reduce interrupt latency. Interrupt handlers do not require any assembler stubs which removes code overhead from the ISRs. Tail-chaining optimization also significantly reduces the overhead when switching from one ISR to another. To optimize low-power designs, the NVIC integrates with the sleep modes, including Deep-sleep mode, which enables the entire device to be rapidly powered down.

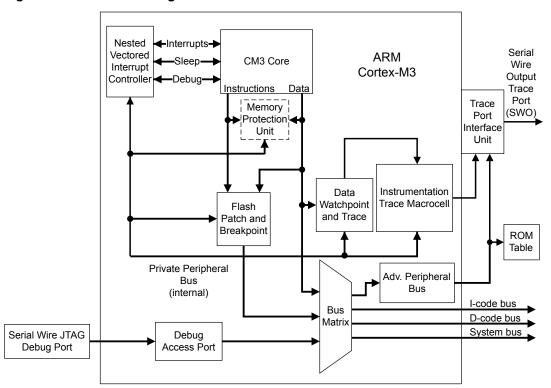


Figure 2-1. CPU Block Diagram

2.2 Overview

2.2.1 System-Level Interface

The Cortex-M3 processor provides multiple interfaces using AMBA® technology to provide high-speed, low-latency memory accesses. The core supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks, and thread-safe Boolean data handling.

The Cortex-M3 processor has a memory protection unit (MPU) that provides fine-grain memory control, enabling applications to implement security privilege levels and separate code, data and stack on a task-by-task basis.

2.2.2 Integrated Configurable Debug

The Cortex-M3 processor implements a complete hardware debug solution, providing high system visibility of the processor and memory through either a traditional JTAG port or a 2-pin Serial Wire Debug (SWD) port that is ideal for microcontrollers and other small package devices. The Stellaris implementation replaces the ARM SW-DP and JTAG-DP with the ARM CoreSight™-compliant Serial Wire JTAG Debug Port (SWJ-DP) interface. The SWJ-DP interface combines the SWD and JTAG debug ports into one module. See the *ARM® Debug Interface V5 Architecture Specification* for details on SWJ-DP.

For system trace, the processor integrates an Instrumentation Trace Macrocell (ITM) alongside data watchpoints and a profiling unit. To enable simple and cost-effective profiling of the system trace events, a Serial Wire Viewer (SWV) can export a stream of software-generated messages, data trace, and profiling information through a single pin.

The Flash Patch and Breakpoint Unit (FPB) provides up to eight hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to eight words in the program code in the CODE memory region. This enables applications stored in a read-only area of Flash memory to be patched in another area of on-chip SRAM or Flash memory. If a patch is required, the application programs the FPB to remap a number of addresses. When those addresses are accessed, the accesses are redirected to a remap table specified in the FPB configuration.

For more information on the Cortex-M3 debug capabilities, see the ARM® Debug Interface V5 Architecture Specification.

2.2.3 Trace Port Interface Unit (TPIU)

The TPIU acts as a bridge between the Cortex-M3 trace data from the ITM, and an off-chip Trace Port Analyzer, as shown in Figure 2-2 on page 52.

Debua Serial Wire ATB Trace Out ATB Asynchronous FIFO Trace Port Interface (serializer) Slave (SWO) Port APB APB Slave Interface Port

Figure 2-2. TPIU Block Diagram

2.2.4 Cortex-M3 System Component Details

The Cortex-M3 includes the following system components:

■ SysTick

A 24-bit count-down timer that can be used as a Real-Time Operating System (RTOS) tick timer or as a simple counter (see "System Timer (SysTick)" on page 91).

Nested Vectored Interrupt Controller (NVIC)

An embedded interrupt controller that supports low latency interrupt processing (see "Nested Vectored Interrupt Controller (NVIC)" on page 92).

■ System Control Block (SCB)

The programming model interface to the processor. The SCB provides system implementation information and system control, including configuration, control, and reporting of system exceptions (see "System Control Block (SCB)" on page 94).

■ Memory Protection Unit (MPU)

Improves system reliability by defining the memory attributes for different memory regions. The MPU provides up to eight different regions and an optional predefined background region (see "Memory Protection Unit (MPU)" on page 94).

2.3 Programming Model

This section describes the Cortex-M3 programming model. In addition to the individual core register descriptions, information about the processor modes and privilege levels for software execution and stacks is included.

2.3.1 Processor Mode and Privilege Levels for Software Execution

The Cortex-M3 has two modes of operation:

Thread mode

Used to execute application software. The processor enters Thread mode when it comes out of reset.

■ Handler mode

Used to handle exceptions. When the processor has finished exception processing, it returns to Thread mode.

In addition, the Cortex-M3 has two privilege levels:

Unprivileged

In this mode, software has the following restrictions:

- Limited access to the MSR and MRS instructions and no use of the CPS instruction
- No access to the system timer, NVIC, or system control block
- Possibly restricted access to memory or peripherals

Privileged

In this mode, software can use all the instructions and has access to all resources.

In Thread mode, the **CONTROL** register (see page 67) controls whether software execution is privileged or unprivileged. In Handler mode, software execution is always privileged.

Only privileged software can write to the **CONTROL** register to change the privilege level for software execution in Thread mode. Unprivileged software can use the SVC instruction to make a supervisor call to transfer control to privileged software.

2.3.2 Stacks

The processor uses a full descending stack, meaning that the stack pointer indicates the last stacked item on the memory. When the processor pushes a new item onto the stack, it decrements the stack pointer and then writes the item to the new memory location. The processor implements two stacks:

the main stack and the process stack, with a pointer for each held in independent registers (see the **SP** register on page 57).

In Thread mode, the **CONTROL** register (see page 67) controls whether the processor uses the main stack or the process stack. In Handler mode, the processor always uses the main stack. The options for processor operations are shown in Table 2-1 on page 54.

Table 2-1. Summary of Processor Mode, Privilege Level, and Stack Use

Processor Mode	Use	Privilege Level	Stack Used
Thread	Applications	Privileged or unprivileged ^a	Main stack or process stack ^a
Handler	Exception handlers	Always privileged	Main stack

a. See CONTROL (page 67).

2.3.3 Register Map

Figure 2-3 on page 54 shows the Cortex-M3 register set. Table 2-2 on page 55 lists the Core registers. The core registers are not memory mapped and are accessed by register name, so the base address is n/a (not applicable) and there is no offset.

Figure 2-3. Cortex-M3 Register Set



Table 2-2. Processor Register Map

Offset	Name	Туре	Reset	Description	See page
-	R0	R/W	-	Cortex General-Purpose Register 0	56
-	R1	R/W	-	Cortex General-Purpose Register 1	56
-	R2	R/W	-	Cortex General-Purpose Register 2	56
-	R3	R/W	-	Cortex General-Purpose Register 3	56
-	R4	R/W	-	Cortex General-Purpose Register 4	56
-	R5	R/W	-	Cortex General-Purpose Register 5	56
-	R6	R/W	-	Cortex General-Purpose Register 6	56
-	R7	R/W	-	Cortex General-Purpose Register 7	56
-	R8	R/W	-	Cortex General-Purpose Register 8	56
-	R9	R/W	-	Cortex General-Purpose Register 9	56
-	R10	R/W	-	Cortex General-Purpose Register 10	56
-	R11	R/W	-	Cortex General-Purpose Register 11	56
-	R12	R/W	-	Cortex General-Purpose Register 12	56
-	SP	R/W	-	Stack Pointer	57
-	LR	R/W	0xFFFF.FFFF	Link Register	58
-	PC	R/W	-	Program Counter	59
-	PSR	R/W	0x0100.0000	Program Status Register	60
-	PRIMASK	R/W	0x0000.0000	Priority Mask Register	64
-	FAULTMASK	R/W	0x0000.0000	Fault Mask Register	65
-	BASEPRI	R/W	0x0000.0000	Base Priority Mask Register	66
-	CONTROL	R/W	0x0000.0000	Control Register	67

2.3.4 Register Descriptions

This section lists and describes the Cortex-M3 registers, in the order shown in Figure 2-3 on page 54. The core registers are not memory mapped and are accessed by register name rather than offset.

Note: The register type shown in the register descriptions refers to type during program execution in Thread mode and Handler mode. Debug access can differ.

Register 1: Cortex General-Purpose Register 0 (R0)

Register 2: Cortex General-Purpose Register 1 (R1)

Register 3: Cortex General-Purpose Register 2 (R2)

Register 4: Cortex General-Purpose Register 3 (R3)

Register 5: Cortex General-Purpose Register 4 (R4)

Register 6: Cortex General-Purpose Register 5 (R5)

Register 7: Cortex General-Purpose Register 6 (R6)

Register 8: Cortex General-Purpose Register 7 (R7)

Register 9: Cortex General-Purpose Register 8 (R8)

Register 10: Cortex General-Purpose Register 9 (R9)

Register 11: Cortex General-Purpose Register 10 (R10)

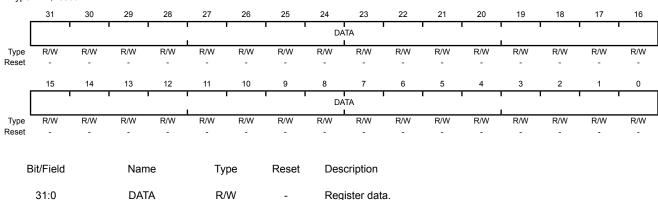
Register 12: Cortex General-Purpose Register 11 (R11)

Register 13: Cortex General-Purpose Register 12 (R12)

The **Rn** registers are 32-bit general-purpose registers for data operations and can be accessed from either privileged or unprivileged mode.

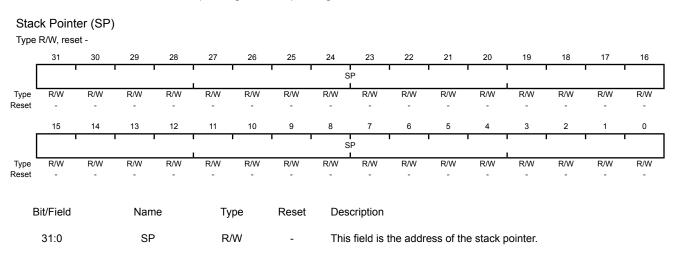
Cortex General-Purpose Register 0 (R0)





Register 14: Stack Pointer (SP)

The **Stack Pointer (SP)** is register R13. In Thread mode, the function of this register changes depending on the ASP bit in the **Control Register (CONTROL)** register. When the ASP bit is clear, this register is the **Main Stack Pointer (MSP)**. When the ASP bit is set, this register is the **Process Stack Pointer (PSP)**. On reset, the ASP bit is clear, and the processor loads the **MSP** with the value from address 0x0000.0000. The **MSP** can only be accessed in privileged mode; the **PSP** can be accessed in either privileged or unprivileged mode.



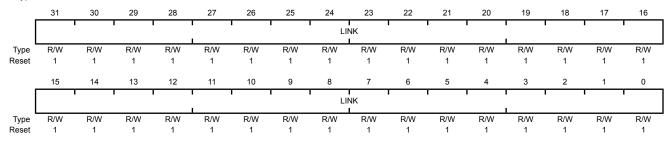
Register 15: Link Register (LR)

The **Link Register (LR)** is register R14, and it stores the return information for subroutines, function calls, and exceptions. **LR** can be accessed from either privileged or unprivileged mode.

 ${\tt EXC_RETURN}$ is loaded into LR on exception entry. See Table 2-10 on page 84 for the values and description.

Link Register (LR)

Type R/W, reset 0xFFFF.FFF



Bit/Field Name Type Reset Description

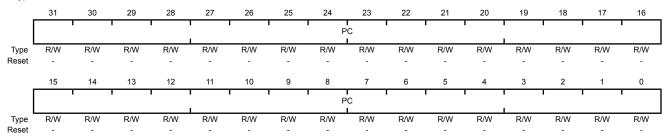
31:0 LINK R/W 0xFFF.FFF This field is the return address.

Register 16: Program Counter (PC)

The **Program Counter (PC)** is register R15, and it contains the current program address. On reset, the processor loads the **PC** with the value of the reset vector, which is at address 0x0000.0004. Bit 0 of the reset vector is loaded into the THUMB bit of the **EPSR** at reset and must be 1. The **PC** register can be accessed in either privileged or unprivileged mode.

Program Counter (PC)





Bit/Field	Name	Type	Reset	Description
31:0	PC	R/W	-	This field is the current program address

Register 17: Program Status Register (PSR)

Note: This register is also referred to as **xPSR**.

The **Program Status Register (PSR)** has three functions, and the register bits are assigned to the different functions:

- Application Program Status Register (APSR), bits 31:27,
- Execution Program Status Register (EPSR), bits 26:24, 15:10
- Interrupt Program Status Register (IPSR), bits 5:0

The **PSR**, **IPSR**, and **EPSR** registers can only be accessed in privileged mode; the **APSR** register can be accessed in either privileged or unprivileged mode.

APSR contains the current state of the condition flags from previous instruction executions.

EPSR contains the Thumb state bit and the execution state bits for the If-Then (IT) instruction or the Interruptible-Continuable Instruction (ICI) field for an interrupted load multiple or store multiple instruction. Attempts to read the **EPSR** directly through application software using the MSR instruction always return zero. Attempts to write the **EPSR** using the MSR instruction in application software are always ignored. Fault handlers can examine the **EPSR** value in the stacked **PSR** to determine the operation that faulted (see "Exception Entry and Return" on page 82).

IPSR contains the exception type number of the current Interrupt Service Routine (ISR).

These registers can be accessed individually or as a combination of any two or all three registers, using the register name as an argument to the MSR or MRS instructions. For example, all of the registers can be read using **PSR** with the MRS instruction, or **APSR** only can be written to using **APSR** with the MSR instruction. page 60 shows the possible register combinations for the **PSR**. See the MRS and MSR instruction descriptions in the $Cortex^{TM}$ -M3/M4 Instruction Set Technical User's Manual for more information about how to access the program status registers.

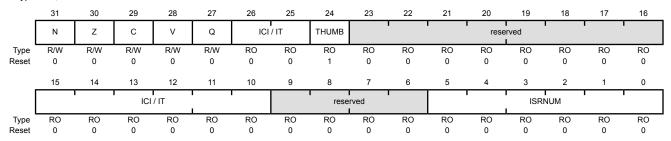
Table 2-3. PSR Register Combinations

Register	Туре	Combination
PSR	R/W ^{a, b}	APSR, EPSR, and IPSR
IEPSR	RO	EPSR and IPSR
IAPSR	R/W ^a	APSR and IPSR
EAPSR	R/W ^b	APSR and EPSR

a. The processor ignores writes to the IPSR bits.

Program Status Register (PSR)

Type R/W, reset 0x0100.0000



b. Reads of the EPSR bits return zero, and the processor ignores writes to these bits.

Bit/Field	Name	Туре	Reset	Description
31	N	R/W	0	APSR Negative or Less Flag
				Value Description
				1 The previous operation result was negative or less than.
				0 The previous operation result was positive, zero, greater than, or equal.
				The value of this bit is only meaningful when accessing PSR or APSR .
30	Z	R/W	0	APSR Zero Flag
				Value Description
				1 The previous operation result was zero.
				The previous operation result was non-zero.
				The value of this bit is only meaningful when accessing PSR or APSR .
29	С	R/W	0	APSR Carry or Borrow Flag
				Value Description
				The previous add operation resulted in a carry bit or the previous subtract operation did not result in a borrow bit.
				The previous add operation did not result in a carry bit or the previous subtract operation resulted in a borrow bit.
				The value of this bit is only meaningful when accessing PSR or APSR .
28	V	R/W	0	APSR Overflow Flag
				Value Description
				1 The previous operation resulted in an overflow.
				0 The previous operation did not result in an overflow.
				The value of this bit is only meaningful when accessing PSR or APSR .
27	Q	R/W	0	APSR DSP Overflow and Saturation Flag
				Value Description
				1 DSP Overflow or saturation has occurred.
				0 DSP overflow or saturation has not occurred since reset or since the bit was last cleared.
				The value of this bit is only meaningful when accessing PSR or APSR .
				This bit is cleared by software using an MRS instruction.

July 15, 2014 61

Bit/Field	Name	Туре	Reset	Description
26:25	ICI / IT	RO	0x0	EPSR ICI / IT status These bits, along with bits 15:10, contain the Interruptible-Continuable Instruction (ICI) field for an interrupted load multiple or store multiple instruction or the execution state bits of the IT instruction. When EPSR holds the ICI execution state, bits 26:25 are zero. The If-Then block contains up to four instructions following an IT instruction. Each instruction in the block is conditional. The conditions for the instructions are either all the same, or some can be the inverse of others. See the Cortex™-M3/M4 Instruction Set Technical User's Manual for more information. The value of this field is only meaningful when accessing PSR or EPSR.
24	THUMB	RO	1	EPSR Thumb State This bit indicates the Thumb state and should always be set. The following can clear the THUMB bit: The BLX, BX and POP{PC} instructions Restoration from the stacked xPSR value on an exception return Bit 0 of the vector value on an exception entry or reset Attempting to execute instructions when this bit is clear results in a fault or lockup. See "Lockup" on page 86 for more information. The value of this bit is only meaningful when accessing PSR or EPSR.
23:16	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:10	ICI / IT	RO	0x0	These bits, along with bits 26:25, contain the Interruptible-Continuable Instruction (ICI) field for an interrupted load multiple or store multiple instruction or the execution state bits of the IT instruction. When an interrupt occurs during the execution of an LDM, STM, PUSH or POP instruction, the processor stops the load multiple or store multiple instruction operation temporarily and stores the next register operand in the multiple operation to bits 15:12. After servicing the interrupt, the processor returns to the register pointed to by bits 15:12 and resumes execution of the multiple load or store instruction. When EPSR holds the ICI execution state, bits 11:10 are zero. The If-Then block contains up to four instructions following a 16-bit IT instruction. Each instruction in the block is conditional. The conditions for the instructions are either all the same, or some can be the inverse of others. See the Cortex™-M3/M4 Instruction Set Technical User's Manual for more information. The value of this field is only meaningful when accessing PSR or EPSR.
9:6	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description	
5:0	ISRNUM	RO	0x00	IPSR ISR No This field co Service Rou	ntains the exception type number of the current Interrupt
				Value	Description
				0x00	Thread mode
				0x01	Reserved
				0x02	NMI
				0x03	Hard fault
				0x04	Memory management fault
				0x05	Bus fault
				0x06	Usage fault
				0x07-0x0A	Reserved
				0x0B	SVCall
				0x0C	Reserved for Debug
				0x0D	Reserved
				0x0E	PendSV
				0x0F	SysTick
				0x10	Interrupt Vector 0
				0x11	Interrupt Vector 1
				0x3B	Interrupt Vector 43
				0x3C-0x3F	Reserved
				Caa "Evaant	ilaa Turaa" ah aasa 77 fan mana information

See "Exception Types" on page 77 for more information.

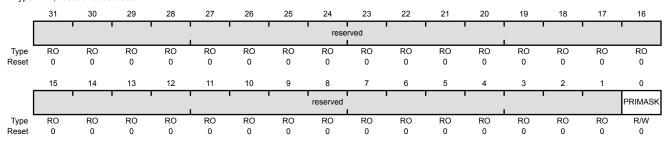
The value of this field is only meaningful when accessing **PSR** or **IPSR**.

Register 18: Priority Mask Register (PRIMASK)

The **PRIMASK** register prevents activation of all exceptions with programmable priority. Reset, non-maskable interrupt (NMI), and hard fault are the only exceptions with fixed priority. Exceptions should be disabled when they might impact the timing of critical tasks. This register is only accessible in privileged mode. The MSR and MRS instructions are used to access the **PRIMASK** register, and the CPS instruction may be used to change the value of the **PRIMASK** register. See the Cortex™-M3/M4 Instruction Set Technical User's Manual for more information on these instructions. For more information on exception priority levels, see "Exception Types" on page 77.

Priority Mask Register (PRIMASK)

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	PRIMASK	R/W	0	Priority Mask

Value Description

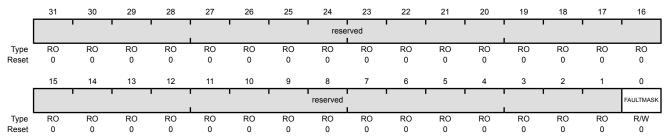
- Prevents the activation of all exceptions with configurable priority.
- 0 No effect.

Register 19: Fault Mask Register (FAULTMASK)

The **FAULTMASK** register prevents activation of all exceptions except for the Non-Maskable Interrupt (NMI). Exceptions should be disabled when they might impact the timing of critical tasks. This register is only accessible in privileged mode. The MSR and MRS instructions are used to access the **FAULTMASK** register, and the CPS instruction may be used to change the value of the **FAULTMASK** register. See the *Cortex™-M3/M4 Instruction Set Technical User's Manual* for more information on these instructions. For more information on exception priority levels, see "Exception Types" on page 77.

Fault Mask Register (FAULTMASK)

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	FAULTMASK	R/W	0	Fault Mask

Value Description

- 1 Prevents the activation of all exceptions except for NMI.
- 0 No effect.

The processor clears the ${\tt FAULTMASK}$ bit on exit from any exception handler except the NMI handler.

Register 20: Base Priority Mask Register (BASEPRI)

The **BASEPRI** register defines the minimum priority for exception processing. When **BASEPRI** is set to a nonzero value, it prevents the activation of all exceptions with the same or lower priority level as the **BASEPRI** value. Exceptions should be disabled when they might impact the timing of critical tasks. This register is only accessible in privileged mode. For more information on exception priority levels, see "Exception Types" on page 77.

Base Priority Mask Register (BASEPRI)

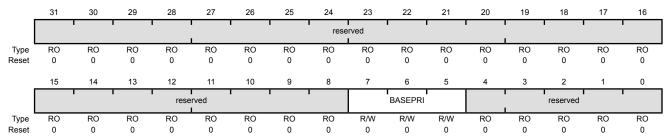


4:0

reserved

RO

0x0



bit/riei	u Name	туре	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:5	BASEPRI	R/W	0x0	Base Priority

Any exception that has a programmable priority level with the same or lower priority as the value of this field is masked. The **PRIMASK** register can be used to mask all exceptions with programmable priority levels. Higher priority exceptions have lower priority levels.

Value Description 0x0 All exceptions are unmasked. 0x1 All exceptions with priority level 1-7 are masked. 0x2 All exceptions with priority level 2-7 are masked. 0x3 All exceptions with priority level 3-7 are masked. All exceptions with priority level 4-7 are masked. 0x4 All exceptions with priority level 5-7 are masked. 0x5 All exceptions with priority level 6-7 are masked. 0x60x7 All exceptions with priority level 7 are masked.

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 21: Control Register (CONTROL)

The **CONTROL** register controls the stack used and the privilege level for software execution when the processor is in Thread mode. This register is only accessible in privileged mode.

Handler mode always uses **MSP**, so the processor ignores explicit writes to the ASP bit of the **CONTROL** register when in Handler mode. The exception entry and return mechanisms automatically update the **CONTROL** register based on the EXC_RETURN value (see Table 2-10 on page 84). In an OS environment, threads running in Thread mode should use the process stack and the kernel and exception handlers should use the main stack. By default, Thread mode uses **MSP**. To switch the stack pointer used in Thread mode to **PSP**, either use the MSR instruction to set the ASP bit, as detailed in the *Cortex*[™]-*M3/M4 Instruction Set Technical User's Manual*, or perform an exception return to Thread mode with the appropriate EXC_RETURN value, as shown in Table 2-10 on page 84.

Note: When changing the stack pointer, software must use an ISB instruction immediately after the MSR instruction, ensuring that instructions after the ISB execute use the new stack pointer. See the *Cortex*TM-*M3/M4 Instruction Set Technical User's Manual*.

Control Register (CONTROL)

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	ASP	R/W	0	Active Stack Pointer
				Value Description
				1 PSP is the current stack pointer.
				0 MSP is the current stack pointer
				In Handler mode, this bit reads as zero and ignores writes. The Cortex-M3 updates this bit automatically on exception return.
0	TMPL	R/W	0	Thread Mode Privilege Level
				Value Description

Value Description

- 1 Unprivileged software can be executed in Thread mode.
- Only privileged software can be executed in Thread mode.

2.3.5 Exceptions and Interrupts

The Cortex-M3 processor supports interrupts and system exceptions. The processor and the Nested Vectored Interrupt Controller (NVIC) prioritize and handle all exceptions. An exception changes the normal flow of software control. The processor uses Handler mode to handle all exceptions except for reset. See "Exception Entry and Return" on page 82 for more information.

The NVIC registers control interrupt handling. See "Nested Vectored Interrupt Controller (NVIC)" on page 92 for more information.

2.3.6 Data Types

The Cortex-M3 supports 32-bit words, 16-bit halfwords, and 8-bit bytes. The processor also supports 64-bit data transfer instructions. All instruction and data memory accesses are little endian. See "Memory Regions, Types and Attributes" on page 69 for more information.

2.4 Memory Model

This section describes the processor memory map, the behavior of memory accesses, and the bit-banding features. The processor has a fixed memory map that provides up to 4 GB of addressable memory.

The memory map for the LM3S6432 controller is provided in Table 2-4 on page 68. In this manual, register addresses are given as a hexadecimal increment, relative to the module's base address as shown in the memory map.

The regions for SRAM and peripherals include bit-band regions. Bit-banding provides atomic operations to bit data (see "Bit-Banding" on page 72).

The processor reserves regions of the Private peripheral bus (PPB) address range for core peripheral registers (see "Cortex-M3 Peripherals" on page 91).

Note: Within the memory map, all reserved space returns a bus fault when read or written.

Table 2-4. Memory Map

Start	End	Description	For details, see page
Memory			
0x0000.0000	0x0001.7FFF	On-chip Flash	237
0x0001.8000	0x1FFF.FFFF	Reserved	-
0x2000.0000	0x2000.7FFF	Bit-banded on-chip SRAM	231
0x2000.8000	0x21FF.FFFF	Reserved	-
0x2200.0000	0x220F.FFFF	Bit-band alias of bit-banded on-chip SRAM starting at 0x2000.0000	231
0x2210.0000	0x3FFF.FFFF	Reserved	-
FiRM Peripherals			•
0x4000.0000	0x4000.0FFF	Watchdog timer 0	344
0x4000.1000	0x4000.3FFF	Reserved	-
0x4000.4000	0x4000.4FFF	GPIO Port A	270
0x4000.5000	0x4000.5FFF	GPIO Port B	270
0x4000.6000	0x4000.6FFF	GPIO Port C	270
0x4000.7000	0x4000.7FFF	GPIO Port D	270
0x4000.8000	0x4000.8FFF	SSI0	455

Table 2-4. Memory Map (continued)

Start	End	Description	For details, see page
0x4000.9000	0x4000.BFFF	Reserved	-
0x4000.C000	0x4000.CFFF	UART0	409
0x4000.D000	0x4000.DFFF	UART1	409
0x4000.E000	0x4001.FFFF	Reserved	-
Peripherals			•
0x4002.0000	0x4002.0FFF	I ² C 0	496
0x4002.1000	0x4002.3FFF	Reserved	-
0x4002.4000	0x4002.4FFF	GPIO Port E	270
0x4002.5000	0x4002.5FFF	GPIO Port F	270
0x4002.6000	0x4002.6FFF	GPIO Port G	270
0x4002.7000	0x4002.7FFF	Reserved	-
0x4002.8000	0x4002.8FFF	PWM	586
0x4002.9000	0x4002.FFFF	Reserved	-
0x4003.0000	0x4003.0FFF	Timer 0	316
0x4003.1000	0x4003.1FFF	Timer 1	316
0x4003.2000	0x4003.2FFF	Timer 2	316
0x4003.3000	0x4003.7FFF	Reserved	-
0x4003.8000	0x4003.8FFF	ADC0	374
0x4003.9000	0x4003.BFFF	Reserved	-
0x4003.C000	0x4003.CFFF	Analog Comparators	566
0x4003.D000	0x400F.CFFF	Reserved	-
0x400F.D000	0x400F.DFFF	Flash memory control	237
0x400F.E000	0x400F.EFFF	System control	181
0x400F.F000	0x41FF.FFFF	Reserved	-
0x4200.0000	0x43FF.FFFF	Bit-banded alias of 0x4000.0000 through 0x400F.FFFF	-
0x4400.0000	0xDFFF.FFFF	Reserved	-
Private Peripheral Bus			•
0xE000.0000	0xE000.0FFF	Instrumentation Trace Macrocell (ITM)	51
0xE000.1000	0xE000.1FFF	Data Watchpoint and Trace (DWT)	51
0xE000.2000	0xE000.2FFF	Flash Patch and Breakpoint (FPB)	51
0xE000.3000	0xE000.DFFF	Reserved	-
0xE000.E000	0xE000.EFFF	Cortex-M3 Peripherals (SysTick, NVIC, MPU and SCB)	99
0xE000.F000	0xE003.FFFF	Reserved	-
0xE004.0000	0xE004.0FFF	Trace Port Interface Unit (TPIU)	52
0xE004.1000	0xFFFF.FFFF	Reserved	-

2.4.1 Memory Regions, Types and Attributes

The memory map and the programming of the MPU split the memory map into regions. Each region has a defined memory type, and some regions have additional memory attributes. The memory type and attributes determine the behavior of accesses to the region.

The memory types are:

- Normal: The processor can re-order transactions for efficiency and perform speculative reads.
- Device: The processor preserves transaction order relative to other transactions to Device or Strongly Ordered memory.
- Strongly Ordered: The processor preserves transaction order relative to all other transactions.

The different ordering requirements for Device and Strongly Ordered memory mean that the memory system can buffer a write to Device memory but must not buffer a write to Strongly Ordered memory.

An additional memory attribute is Execute Never (XN), which means the processor prevents instruction accesses. A fault exception is generated only on execution of an instruction executed from an XN region.

2.4.2 Memory System Ordering of Memory Accesses

For most memory accesses caused by explicit memory access instructions, the memory system does not guarantee that the order in which the accesses complete matches the program order of the instructions, providing the order does not affect the behavior of the instruction sequence. Normally, if correct program execution depends on two memory accesses completing in program order, software must insert a memory barrier instruction between the memory access instructions (see "Software Ordering of Memory Accesses" on page 71).

However, the memory system does guarantee ordering of accesses to Device and Strongly Ordered memory. For two memory access instructions A1 and A2, if both A1 and A2 are accesses to either Device or Strongly Ordered memory, and if A1 occurs before A2 in program order, A1 is always observed before A2.

2.4.3 Behavior of Memory Accesses

Table 2-5 on page 70 shows the behavior of accesses to each region in the memory map. See "Memory Regions, Types and Attributes" on page 69 for more information on memory types and the XN attribute. Stellaris devices may have reserved memory areas within the address ranges shown below (refer to Table 2-4 on page 68 for more information).

Table 2-5. Memory Access Behavior

Address Range	Memory Region	Memory Type	Execute Never (XN)	Description
0x0000.0000 - 0x1FFF.FFF	Code	Normal	-	This executable region is for program code. Data can also be stored here.
0x2000.0000 - 0x3FFF.FFFF	SRAM	Normal	-	This executable region is for data. Code can also be stored here. This region includes bit band and bit band alias areas (see Table 2-6 on page 72).
0x4000.0000 - 0x5FFF.FFFF	Peripheral	Device	XN	This region includes bit band and bit band alias areas (see Table 2-7 on page 72).
0x6000.0000 - 0x9FFF.FFFF	External RAM	Normal	-	This executable region is for data.
0xA000.0000 - 0xDFFF.FFFF	External device	Device	XN	This region is for external device memory.
0xE000.0000- 0xE00F.FFFF	Private peripheral bus	Strongly Ordered	XN	This region includes the NVIC, system timer, and system control block.
0xE010.0000- 0xFFFF.FFF	Reserved	-	-	-

The Code, SRAM, and external RAM regions can hold programs. However, it is recommended that programs always use the Code region because the Cortex-M3 has separate buses that can perform instruction fetches and data accesses simultaneously.

The MPU can override the default memory access behavior described in this section. For more information, see "Memory Protection Unit (MPU)" on page 94.

The Cortex-M3 prefetches instructions ahead of execution and speculatively prefetches from branch target addresses.

2.4.4 Software Ordering of Memory Accesses

The order of instructions in the program flow does not always guarantee the order of the corresponding memory transactions for the following reasons:

- The processor can reorder some memory accesses to improve efficiency, providing this does not affect the behavior of the instruction sequence.
- The processor has multiple bus interfaces.
- Memory or devices in the memory map have different wait states.
- Some memory accesses are buffered or speculative.

"Memory System Ordering of Memory Accesses" on page 70 describes the cases where the memory system guarantees the order of memory accesses. Otherwise, if the order of memory accesses is critical, software must include memory barrier instructions to force that ordering. The Cortex-M3 has the following memory barrier instructions:

- The Data Memory Barrier (DMB) instruction ensures that outstanding memory transactions complete before subsequent memory transactions.
- The Data Synchronization Barrier (DSB) instruction ensures that outstanding memory transactions complete before subsequent instructions execute.
- The Instruction Synchronization Barrier (ISB) instruction ensures that the effect of all completed memory transactions is recognizable by subsequent instructions.

Memory barrier instructions can be used in the following situations:

- MPU programming
 - If the MPU settings are changed and the change must be effective on the very next instruction, use a DSB instruction to ensure the effect of the MPU takes place immediately at the end of context switching.
 - Use an ISB instruction to ensure the new MPU setting takes effect immediately after programming the MPU region or regions, if the MPU configuration code was accessed using a branch or call. If the MPU configuration code is entered using exception mechanisms, then an ISB instruction is not required.

Vector table

If the program changes an entry in the vector table and then enables the corresponding exception, use a DMB instruction between the operations. The DMB instruction ensures that if the exception is taken immediately after being enabled, the processor uses the new exception vector.

Self-modifying code

If a program contains self-modifying code, use an ISB instruction immediately after the code modification in the program. The ISB instruction ensures subsequent instruction execution uses the updated program.

Memory map switching

If the system contains a memory map switching mechanism, use a DSB instruction after switching the memory map in the program. The DSB instruction ensures subsequent instruction execution uses the updated memory map.

Dynamic exception priority change

When an exception priority has to change when the exception is pending or active, use DSB instructions after the change. The change then takes effect on completion of the DSB instruction.

Memory accesses to Strongly Ordered memory, such as the System Control Block, do not require the use of DMB instructions.

For more information on the memory barrier instructions, see the *Cortex*™-*M3/M4 Instruction Set Technical User's Manual*.

2.4.5 Bit-Banding

A bit-band region maps each word in a bit-band alias region to a single bit in the bit-band region. The bit-band regions occupy the lowest 1 MB of the SRAM and peripheral memory regions. Accesses to the 32-MB SRAM alias region map to the 1-MB SRAM bit-band region, as shown in Table 2-6 on page 72. Accesses to the 32-MB peripheral alias region map to the 1-MB peripheral bit-band region, as shown in Table 2-7 on page 72. For the specific address range of the bit-band regions, see Table 2-4 on page 68.

Note: A word access to the SRAM or the peripheral bit-band alias region maps to a single bit in the SRAM or peripheral bit-band region.

A word access to a bit band address results in a word access to the underlying memory, and similarly for halfword and byte accesses. This allows bit band accesses to match the access requirements of the underlying peripheral.

Table 2-6. SRAM Memory Bit-Banding Regions

Address Range		Memory Region	Instruction and Data Accesses	
Start	End	Memory Region	instruction and Data Accesses	
0x2000.0000	0x2000.7FFF		Direct accesses to this memory range behave as SRAM memory accesses, but this region is also bit addressable through bit-band alias.	
0x2200.0000	0x220F.FFFF	SRAM bit-band alias	Data accesses to this region are remapped to bit band region. A write operation is performed as read-modify-write. Instruction accesses are not remapped.	

Table 2-7. Peripheral Memory Bit-Banding Regions

Address Range		Memory Region	Instruction and Data Accesses	
Start	End	Welliory Region	Instruction and Data Accesses	
0x4000.0000	0x400F.FFFF	region	Direct accesses to this memory range behave as peripheral memory accesses, but this region is also bit addressable through bit-band alias.	

Table 2-7. Peripheral Memory Bit-Banding Regions (continued)

Address Range		Memory Region	Instruction and Data Accesses	
Start End		Memory Region		
0x4200.0000	0x43FF.FFFF	'	Data accesses to this region are remapped to bit band region. A write operation is performed as read-modify-write. Instruction accesses are not permitted.	

The following formula shows how the alias region maps onto the bit-band region:

```
bit_word_offset = (byte_offset x 32) + (bit_number x 4)
bit_word_addr = bit_band_base + bit_word_offset
```

where:

bit word offset

The position of the target bit in the bit-band memory region.

bit word addr

The address of the word in the alias memory region that maps to the targeted bit.

bit band base

The starting address of the alias region.

byte_offset

The number of the byte in the bit-band region that contains the targeted bit.

bit number

The bit position, 0-7, of the targeted bit.

Figure 2-4 on page 74 shows examples of bit-band mapping between the SRAM bit-band alias region and the SRAM bit-band region:

■ The alias word at 0x23FF.FFE0 maps to bit 0 of the bit-band byte at 0x200F.FFFF:

```
0x23FF.FFE0 = 0x2200.0000 + (0x000F.FFFF*32) + (0*4)
```

■ The alias word at 0x23FF.FFFC maps to bit 7 of the bit-band byte at 0x200F.FFFF:

```
0x23FF.FFFC = 0x2200.0000 + (0x000F.FFFF*32) + (7*4)
```

■ The alias word at 0x2200.0000 maps to bit 0 of the bit-band byte at 0x2000.0000:

```
0x2200.0000 = 0x2200.0000 + (0*32) + (0*4)
```

■ The alias word at 0x2200.001C maps to bit 7 of the bit-band byte at 0x2000.0000:

```
0x2200.001C = 0x2200.0000 + (0*32) + (7*4)
```

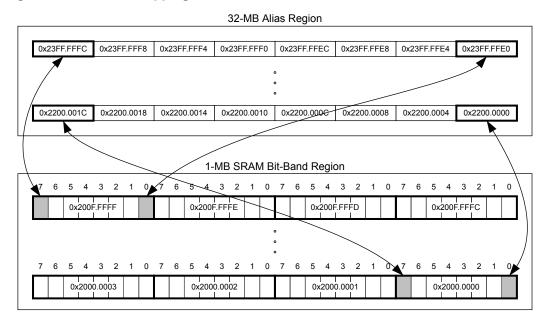


Figure 2-4. Bit-Band Mapping

2.4.5.1 Directly Accessing an Alias Region

Writing to a word in the alias region updates a single bit in the bit-band region.

Bit 0 of the value written to a word in the alias region determines the value written to the targeted bit in the bit-band region. Writing a value with bit 0 set writes a 1 to the bit-band bit, and writing a value with bit 0 clear writes a 0 to the bit-band bit.

Bits 31:1 of the alias word have no effect on the bit-band bit. Writing 0x01 has the same effect as writing 0xFF. Writing 0x00 has the same effect as writing 0x0E.

When reading a word in the alias region, 0x0000.0000 indicates that the targeted bit in the bit-band region is clear and 0x0000.0001 indicates that the targeted bit in the bit-band region is set.

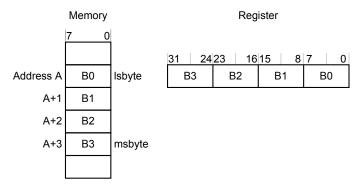
2.4.5.2 Directly Accessing a Bit-Band Region

"Behavior of Memory Accesses" on page 70 describes the behavior of direct byte, halfword, or word accesses to the bit-band regions.

2.4.6 Data Storage

The processor views memory as a linear collection of bytes numbered in ascending order from zero. For example, bytes 0-3 hold the first stored word, and bytes 4-7 hold the second stored word. Data is stored in little-endian format, with the least-significant byte (Isbyte) of a word stored at the lowest-numbered byte, and the most-significant byte (msbyte) stored at the highest-numbered byte. Figure 2-5 on page 75 illustrates how data is stored.

Figure 2-5. Data Storage



2.4.7 Synchronization Primitives

The Cortex-M3 instruction set includes pairs of synchronization primitives which provide a non-blocking mechanism that a thread or process can use to obtain exclusive access to a memory location. Software can use these primitives to perform a guaranteed read-modify-write memory update sequence or for a semaphore mechanism.

A pair of synchronization primitives consists of:

- A Load-Exclusive instruction, which is used to read the value of a memory location and requests exclusive access to that location.
- A Store-Exclusive instruction, which is used to attempt to write to the same memory location and returns a status bit to a register. If this status bit is clear, it indicates that the thread or process gained exclusive access to the memory and the write succeeds; if this status bit is set, it indicates that the thread or process did not gain exclusive access to the memory and no write was performed.

The pairs of Load-Exclusive and Store-Exclusive instructions are:

- The word instructions LDREX and STREX
- The halfword instructions LDREXH and STREXH
- The byte instructions LDREXB and STREXB

Software must use a Load-Exclusive instruction with the corresponding Store-Exclusive instruction.

To perform an exclusive read-modify-write of a memory location, software must:

- 1. Use a Load-Exclusive instruction to read the value of the location.
- **2.** Modify the value, as required.
- **3.** Use a Store-Exclusive instruction to attempt to write the new value back to the memory location.
- 4. Test the returned status bit.

If the status bit is clear, the read-modify-write completed successfully. If the status bit is set, no write was performed, which indicates that the value returned at step 1 might be out of date. The software must retry the entire read-modify-write sequence.

Software can use the synchronization primitives to implement a semaphore as follows:

- **1.** Use a Load-Exclusive instruction to read from the semaphore address to check whether the semaphore is free.
- 2. If the semaphore is free, use a Store-Exclusive to write the claim value to the semaphore address.
- **3.** If the returned status bit from step 2 indicates that the Store-Exclusive succeeded, then the software has claimed the semaphore. However, if the Store-Exclusive failed, another process might have claimed the semaphore after the software performed step 1.

The Cortex-M3 includes an exclusive access monitor that tags the fact that the processor has executed a Load-Exclusive instruction. The processor removes its exclusive access tag if:

- It executes a CLREX instruction.
- It executes a Store-Exclusive instruction, regardless of whether the write succeeds.
- An exception occurs, which means the processor can resolve semaphore conflicts between different threads.

For more information about the synchronization primitive instructions, see the *Cortex™-M3/M4 Instruction Set Technical User's Manual.*

2.5 Exception Model

The ARM Cortex-M3 processor and the Nested Vectored Interrupt Controller (NVIC) prioritize and handle all exceptions in Handler Mode. The processor state is automatically stored to the stack on an exception and automatically restored from the stack at the end of the Interrupt Service Routine (ISR). The vector is fetched in parallel to the state saving, enabling efficient interrupt entry. The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoration.

Table 2-8 on page 78 lists all exception types. Software can set eight priority levels on seven of these exceptions (system handlers) as well as on 29 interrupts (listed in Table 2-9 on page 79).

Priorities on the system handlers are set with the NVIC **System Handler Priority n (SYSPRIn)** registers. Interrupts are enabled through the NVIC **Interrupt Set Enable n (ENn)** register and prioritized with the NVIC **Interrupt Priority n (PRIn)** registers. Priorities can be grouped by splitting priority levels into preemption priorities and subpriorities. All the interrupt registers are described in "Nested Vectored Interrupt Controller (NVIC)" on page 92.

Internally, the highest user-programmable priority (0) is treated as fourth priority, after a Reset, Non-Maskable Interrupt (NMI), and a Hard Fault, in that order. Note that 0 is the default priority for all the programmable priorities.

Important: After a write to clear an interrupt source, it may take several processor cycles for the NVIC to see the interrupt source de-assert. Thus if the interrupt clear is done as the last action in an interrupt handler, it is possible for the interrupt handler to complete while the NVIC sees the interrupt as still asserted, causing the interrupt handler to be re-entered errantly. This situation can be avoided by either clearing the interrupt source at the beginning of the interrupt handler or by performing a read or write after the write to clear the interrupt source (and flush the write buffer).

See "Nested Vectored Interrupt Controller (NVIC)" on page 92 for more information on exceptions and interrupts.

2.5.1 Exception States

Each exception is in one of the following states:

- Inactive. The exception is not active and not pending.
- **Pending.** The exception is waiting to be serviced by the processor. An interrupt request from a peripheral or from software can change the state of the corresponding interrupt to pending.
- **Active.** An exception that is being serviced by the processor but has not completed.

Note: An exception handler can interrupt the execution of another exception handler. In this case, both exceptions are in the active state.

■ **Active and Pending.** The exception is being serviced by the processor, and there is a pending exception from the same source.

2.5.2 Exception Types

The exception types are:

- Reset. Reset is invoked on power up or a warm reset. The exception model treats reset as a special form of exception. When reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is deasserted, execution restarts from the address provided by the reset entry in the vector table. Execution restarts as privileged execution in Thread mode.
- NMI. A non-maskable Interrupt (NMI) can be signaled using the NMI signal or triggered by software using the Interrupt Control and State (INTCTRL) register. This exception has the highest priority other than reset. NMI is permanently enabled and has a fixed priority of -2. NMIs cannot be masked or prevented from activation by any other exception or preempted by any exception other than reset.
- Hard Fault. A hard fault is an exception that occurs because of an error during exception processing, or because an exception cannot be managed by any other exception mechanism. Hard faults have a fixed priority of -1, meaning they have higher priority than any exception with configurable priority.
- Memory Management Fault. A memory management fault is an exception that occurs because of a memory protection related fault, including access violation and no match. The MPU or the fixed memory protection constraints determine this fault, for both instruction and data memory transactions. This fault is used to abort instruction accesses to Execute Never (XN) memory regions, even if the MPU is disabled.
- **Bus Fault.** A bus fault is an exception that occurs because of a memory-related fault for an instruction or data memory transaction such as a prefetch fault or a memory access fault. This fault can be enabled or disabled.
- **Usage Fault.** A usage fault is an exception that occurs because of a fault related to instruction execution, such as:
 - An undefined instruction
 - An illegal unaligned access
 - Invalid state on instruction execution

An error on exception return

An unaligned address on a word or halfword memory access or division by zero can cause a usage fault when the core is properly configured.

- **SVCall.** A supervisor call (SVC) is an exception that is triggered by the SVC instruction. In an OS environment, applications can use SVC instructions to access OS kernel functions and device drivers.
- **Debug Monitor.** This exception is caused by the debug monitor (when not halting). This exception is only active when enabled. This exception does not activate if it is a lower priority than the current activation.
- **PendSV.** PendSV is a pendable, interrupt-driven request for system-level service. In an OS environment, use PendSV for context switching when no other exception is active. PendSV is triggered using the **Interrupt Control and State (INTCTRL)** register.
- SysTick. A SysTick exception is an exception that the system timer generates when it reaches zero when it is enabled to generate an interrupt. Software can also generate a SysTick exception using the Interrupt Control and State (INTCTRL) register. In an OS environment, the processor can use this exception as system tick.
- Interrupt (IRQ). An interrupt, or IRQ, is an exception signaled by a peripheral or generated by a software request and fed through the NVIC (prioritized). All interrupts are asynchronous to instruction execution. In the system, peripherals use interrupts to communicate with the processor. Table 2-9 on page 79 lists the interrupts on the LM3S6432 controller.

For an asynchronous exception, other than reset, the processor can execute another instruction between when the exception is triggered and when the processor enters the exception handler.

Privileged software can disable the exceptions that Table 2-8 on page 78 shows as having configurable priority (see the **SYSHNDCTRL** register on page 133 and the **DIS0** register on page 108).

For more information about hard faults, memory management faults, bus faults, and usage faults, see "Fault Handling" on page 84.

Table 2-8. Exception Types

Exception Type	Vector Number	Priority ^a	Vector Address or Offset ^b	Activation
-	0	-	0x0000.0000	Stack top is loaded from the first entry of the vector table on reset.
Reset	1	-3 (highest)	0x0000.0004	Asynchronous
Non-Maskable Interrupt (NMI)	2	-2	0x0000.0008	Asynchronous
Hard Fault	3	-1	0x0000.000C	-
Memory Management	4	programmable ^c	0x0000.0010	Synchronous
Bus Fault	5	programmable ^c	0x0000.0014	Synchronous when precise and asynchronous when imprecise
Usage Fault	6	programmable ^c	0x0000.0018	Synchronous
-	7-10	-	-	Reserved
SVCall	11	programmable ^c	0x0000.002C	Synchronous
Debug Monitor	12	programmable ^c	0x0000.0030	Synchronous
-	13	-	-	Reserved

Table 2-8. Exception Types (continued)

Exception Type	Vector Number	Priority ^a	Vector Address or Offset ^b	Activation
PendSV	14	programmable ^c	0x0000.0038	Asynchronous
SysTick	15	programmable ^c	0x0000.003C	Asynchronous
Interrupts	16 and above	programmable ^d	0x0000.0040 and above	Asynchronous

a. 0 is the default priority for all the programmable priorities.

Table 2-9. Interrupts

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description
0-15	-	0x0000.0000 - 0x0000.003C	Processor exceptions
16	0	0x0000.0040	GPIO Port A
17	1	0x0000.0044	GPIO Port B
18	2	0x0000.0048	GPIO Port C
19	3	0x0000.004C	GPIO Port D
20	4	0x0000.0050	GPIO Port E
21	5	0x0000.0054	UART0
22	6	0x0000.0058	UART1
23	7	0x0000.005C	SSI0
24	8	0x0000.0060	I ² C0
25	9	0x0000.0064	PWM Fault
26	10	0x0000.0068	PWM Generator 0
27-29	11-13	-	Reserved
30	14	0x0000.0078	ADC0 Sequence 0
31	15	0x0000.007C	ADC0 Sequence 1
32	16	0x0000.0080	ADC0 Sequence 2
33	17	0x0000.0084	ADC0 Sequence 3
34	18	0x0000.0088	Watchdog Timer 0
35	19	0x0000.008C	Timer 0A
36	20	0x0000.0090	Timer 0B
37	21	0x0000.0094	Timer 1A
38	22	0x0000.0098	Timer 1B
39	23	0x0000.009C	Timer 2A
40	24	0x0000.00A0	Timer 2B
41	25	0x0000.00A4	Analog Comparator 0
42	26	0x0000.00A8	Analog Comparator 1
43	27	-	Reserved
44	28	0x0000.00B0	System Control
45	29	0x0000.00B4	Flash Memory Control
46	30	0x0000.00B8	GPIO Port F

b. See "Vector Table" on page 80.

c. See SYSPRI1 on page 130.

d. See **PRIn** registers on page 116.

Table 2-9. Interrupts (continued)

Vector Number	Interrupt Number (Bit in Interrupt Registers)		Description
47	31	0x0000.00BC	GPIO Port G
48-57	32-41	-	Reserved
58	42	0x0000.00E8	Ethernet Controller
59	43	-	Reserved

2.5.3 Exception Handlers

The processor handles exceptions using:

- Interrupt Service Routines (ISRs). Interrupts (IRQx) are the exceptions handled by ISRs.
- Fault Handlers. Hard fault, memory management fault, usage fault, and bus fault are fault exceptions handled by the fault handlers.
- **System Handlers.** NMI, PendSV, SVCall, SysTick, and the fault exceptions are all system exceptions that are handled by system handlers.

2.5.4 Vector Table

The vector table contains the reset value of the stack pointer and the start addresses, also called exception vectors, for all exception handlers. The vector table is constructed using the vector address or offset shown in Table 2-8 on page 78. Figure 2-6 on page 81 shows the order of the exception vectors in the vector table. The least-significant bit of each vector must be 1, indicating that the exception handler is Thumb code

Figure 2-6. Vector Table

Exception number	IRQ number	Offset	Vector
59	43	0x00EC	IRQ43
18 17 16 15 14 13 12 11 10 9 8 7	2 1 0 -1 -2	0x004C 0x004C 0x0048 0x0044 0x0040 0x003C 0x003B	IRQ2 IRQ1 IRQ0 Systick PendSV Reserved Reserved for Debug SVCall
6	-10	0x0018	Usage fault
5	-11	0x0010	Bus fault
4	-12	0x0014	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
			Initial SP value
		0x0000	miliar or value

On system reset, the vector table is fixed at address 0x0000.0000. Privileged software can write to the **Vector Table Offset (VTABLE)** register to relocate the vector table start address to a different memory location, in the range 0x0000.0100 to 0x3FFF.FF00 (see "Vector Table" on page 80). Note that when configuring the **VTABLE** register, the offset must be aligned on a 256-byte boundary.

2.5.5 Exception Priorities

As Table 2-8 on page 78 shows, all exceptions have an associated priority, with a lower priority value indicating a higher priority and configurable priorities for all exceptions except Reset, Hard fault, and NMI. If software does not configure any priorities, then all exceptions with a configurable priority have a priority of 0. For information about configuring exception priorities, see page 130 and page 116.

Note: Configurable priority values for the Stellaris implementation are in the range 0-7. This means that the Reset, Hard fault, and NMI exceptions, with fixed negative priority values, always have higher priority than any other exception.

For example, assigning a higher priority value to IRQ[0] and a lower priority value to IRQ[1] means that IRQ[1] has higher priority than IRQ[0]. If both IRQ[1] and IRQ[0] are asserted, IRQ[1] is processed before IRQ[0].

If multiple pending exceptions have the same priority, the pending exception with the lowest exception number takes precedence. For example, if both IRQ[0] and IRQ[1] are pending and have the same priority, then IRQ[0] is processed before IRQ[1].

When the processor is executing an exception handler, the exception handler is preempted if a higher priority exception occurs. If an exception occurs with the same priority as the exception being handled, the handler is not preempted, irrespective of the exception number. However, the status of the new interrupt changes to pending.

2.5.6 Interrupt Priority Grouping

To increase priority control in systems with interrupts, the NVIC supports priority grouping. This grouping divides each interrupt priority register entry into two fields:

- An upper field that defines the group priority
- A lower field that defines a subpriority within the group

Only the group priority determines preemption of interrupt exceptions. When the processor is executing an interrupt exception handler, another interrupt with the same group priority as the interrupt being handled does not preempt the handler.

If multiple pending interrupts have the same group priority, the subpriority field determines the order in which they are processed. If multiple pending interrupts have the same group priority and subpriority, the interrupt with the lowest IRQ number is processed first.

For information about splitting the interrupt priority fields into group priority and subpriority, see page 124.

2.5.7 Exception Entry and Return

Descriptions of exception handling use the following terms:

- **Preemption.** When the processor is executing an exception handler, an exception can preempt the exception handler if its priority is higher than the priority of the exception being handled. See "Interrupt Priority Grouping" on page 82 for more information about preemption by an interrupt. When one exception preempts another, the exceptions are called nested exceptions. See "Exception Entry" on page 83 more information.
- **Return.** Return occurs when the exception handler is completed, and there is no pending exception with sufficient priority to be serviced and the completed exception handler was not handling a late-arriving exception. The processor pops the stack and restores the processor state to the state it had before the interrupt occurred. See "Exception Return" on page 84 for more information.
- **Tail-Chaining.** This mechanism speeds up exception servicing. On completion of an exception handler, if there is a pending exception that meets the requirements for exception entry, the stack pop is skipped and control transfers to the new exception handler.
- Late-Arriving. This mechanism speeds up preemption. If a higher priority exception occurs during state saving for a previous exception, the processor switches to handle the higher priority exception and initiates the vector fetch for that exception. State saving is not affected by late arrival because the state saved is the same for both exceptions. Therefore, the state saving continues uninterrupted. The processor can accept a late arriving exception until the first instruction of the exception handler of the original exception enters the execute stage of the processor. On

return from the exception handler of the late-arriving exception, the normal tail-chaining rules apply.

2.5.7.1 Exception Entry

Exception entry occurs when there is a pending exception with sufficient priority and either the processor is in Thread mode or the new exception is of higher priority than the exception being handled, in which case the new exception preempts the original exception.

When one exception preempts another, the exceptions are nested.

Sufficient priority means the exception has more priority than any limits set by the mask registers (see **PRIMASK** on page 64, **FAULTMASK** on page 65, and **BASEPRI** on page 66). An exception with less priority than this is pending but is not handled by the processor.

When the processor takes an exception, unless the exception is a tail-chained or a late-arriving exception, the processor pushes information onto the current stack. This operation is referred to as *stacking* and the structure of eight data words is referred to as *stack frame*.

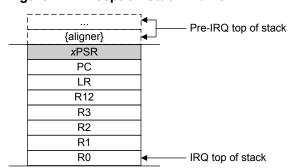


Figure 2-7. Exception Stack Frame

Immediately after stacking, the stack pointer indicates the lowest address in the stack frame. Unless stack alignment is disabled, the stack frame is aligned to a double-word address. If the STKALIGN bit of the **Configuration Control (CCR)** register is set, stack align adjustment is performed during stacking.

The stack frame includes the return address, which is the address of the next instruction in the interrupted program. This value is restored to the **PC** at exception return so that the interrupted program resumes.

In parallel to the stacking operation, the processor performs a vector fetch that reads the exception handler start address from the vector table. When stacking is complete, the processor starts executing the exception handler. At the same time, the processor writes an EXC_RETURN value to the **LR**, indicating which stack pointer corresponds to the stack frame and what operation mode the processor was in before the entry occurred.

If no higher-priority exception occurs during exception entry, the processor starts executing the exception handler and automatically changes the status of the corresponding pending interrupt to active.

If another higher-priority exception occurs during exception entry, known as late arrival, the processor starts executing the exception handler for this exception and does not change the pending status of the earlier exception.

2.5.7.2 Exception Return

Exception return occurs when the processor is in Handler mode and executes one of the following instructions to load the EXC_RETURN value into the **PC**:

- An LDM or POP instruction that loads the PC
- A BX instruction using any register
- An LDR instruction with the **PC** as the destination

EXC_RETURN is the value loaded into the **LR** on exception entry. The exception mechanism relies on this value to detect when the processor has completed an exception handler. The lowest four bits of this value provide information on the return stack and processor mode. Table 2-10 on page 84 shows the EXC_RETURN values with a description of the exception return behavior.

EXC_RETURN bits 31:4 are all set. When this value is loaded into the **PC**, it indicates to the processor that the exception is complete, and the processor initiates the appropriate exception return sequence.

Table 2-10. Exception Return Behavior

EXC_RETURN[31:0]	Description
0xFFFF.FFF0	Reserved
0xFFFF.FFF1	Return to Handler mode.
	Exception return uses state from MSP.
	Execution uses MSP after return.
0xFFFF.FFF2 - 0xFFFF.FFF8	Reserved
0xFFFF.FFF9	Return to Thread mode.
	Exception return uses state from MSP.
	Execution uses MSP after return.
0xFFFF.FFFA - 0xFFFF.FFFC	Reserved
0xFFFF.FFFD	Return to Thread mode.
	Exception return uses state from PSP.
	Execution uses PSP after return.
0xFFFF.FFFE - 0xFFFF.FFFF	Reserved

2.6 Fault Handling

Faults are a subset of the exceptions (see "Exception Model" on page 76). The following conditions generate a fault:

- A bus error on an instruction fetch or vector table load or a data access.
- An internally detected error such as an undefined instruction or an attempt to change state with a BX instruction.
- Attempting to execute an instruction from a memory region marked as Non-Executable (XN).
- An MPU fault because of a privilege violation or an attempt to access an unmanaged region.

2.6.1 Fault Types

Table 2-11 on page 85 shows the types of fault, the handler used for the fault, the corresponding fault status register, and the register bit that indicates the fault has occurred. See page 137 for more information about the fault status registers.

Table 2-11. Faults

Fault	Handler	Fault Status Register	Bit Name
Bus error on a vector read	Hard fault	Hard Fault Status (HFAULTSTAT)	VECT
Fault escalated to a hard fault	Hard fault	Hard Fault Status (HFAULTSTAT)	FORCED
MPU or default memory mismatch on instruction access	Memory management fault	Memory Management Fault Status (MFAULTSTAT)	IERR ^a
MPU or default memory mismatch on data access	Memory management fault	Memory Management Fault Status (MFAULTSTAT)	DERR
MPU or default memory mismatch on exception stacking	Memory management fault	Memory Management Fault Status (MFAULTSTAT)	MSTKE
MPU or default memory mismatch on exception unstacking	Memory management fault	Memory Management Fault Status (MFAULTSTAT)	MUSTKE
Bus error during exception stacking	Bus fault	Bus Fault Status (BFAULTSTAT)	BSTKE
Bus error during exception unstacking	Bus fault	Bus Fault Status (BFAULTSTAT)	BUSTKE
Bus error during instruction prefetch	Bus fault	Bus Fault Status (BFAULTSTAT)	IBUS
Precise data bus error	Bus fault	Bus Fault Status (BFAULTSTAT)	PRECISE
Imprecise data bus error	Bus fault	Bus Fault Status (BFAULTSTAT)	IMPRE
Attempt to access a coprocessor	Usage fault	Usage Fault Status (UFAULTSTAT)	NOCP
Undefined instruction	Usage fault	Usage Fault Status (UFAULTSTAT)	UNDEF
Attempt to enter an invalid instruction set state ^b	Usage fault	Usage Fault Status (UFAULTSTAT)	INVSTAT
Invalid EXC_RETURN value	Usage fault	Usage Fault Status (UFAULTSTAT)	INVPC
Illegal unaligned load or store	Usage fault	Usage Fault Status (UFAULTSTAT)	UNALIGN
Divide by 0	Usage fault	Usage Fault Status (UFAULTSTAT)	DIV0

a. Occurs on an access to an XN region even if the MPU is disabled.

2.6.2 Fault Escalation and Hard Faults

All fault exceptions except for hard fault have configurable exception priority (see **SYSPRI1** on page 130). Software can disable execution of the handlers for these faults (see **SYSHNDCTRL** on page 133).

Usually, the exception priority, together with the values of the exception mask registers, determines whether the processor enters the fault handler, and whether a fault handler can preempt another fault handler as described in "Exception Model" on page 76.

In some situations, a fault with configurable priority is treated as a hard fault. This process is called priority escalation, and the fault is described as *escalated to hard fault*. Escalation to hard fault occurs when:

A fault handler causes the same kind of fault as the one it is servicing. This escalation to hard fault occurs because a fault handler cannot preempt itself because it must have the same priority as the current priority level.

b. Attempting to use an instruction set other than the Thumb instruction set, or returning to a non load-store-multiple instruction with ICI continuation.

- A fault handler causes a fault with the same or lower priority as the fault it is servicing. This situation happens because the handler for the new fault cannot preempt the currently executing fault handler.
- An exception handler causes a fault for which the priority is the same as or lower than the currently executing exception.
- A fault occurs and the handler for that fault is not enabled.

If a bus fault occurs during a stack push when entering a bus fault handler, the bus fault does not escalate to a hard fault. Thus if a corrupted stack causes a fault, the fault handler executes even though the stack push for the handler failed. The fault handler operates but the stack contents are corrupted.

Note: Only Reset and NMI can preempt the fixed priority hard fault. A hard fault can preempt any exception other than Reset, NMI, or another hard fault.

2.6.3 Fault Status Registers and Fault Address Registers

The fault status registers indicate the cause of a fault. For bus faults and memory management faults, the fault address register indicates the address accessed by the operation that caused the fault, as shown in Table 2-12 on page 86.

Table 2-12. Fault Status and Fault Address Registers

Handler	Status Register Name	Address Register Name	Register Description
Hard fault	Hard Fault Status (HFAULTSTAT)	-	page 143
, ,	Memory Management Fault Status	Memory Management Fault	page 137
fault	(MFAULTSTAT)	Address (MMADDR)	page 144
Bus fault	Bus Fault Status (BFAULTSTAT)	Bus Fault Address	page 137
		(FAULTADDR)	page 145
Usage fault	Usage Fault Status (UFAULTSTAT)	-	page 137

2.6.4 **Lockup**

The processor enters a lockup state if a hard fault occurs when executing the NMI or hard fault handlers. When the processor is in the lockup state, it does not execute any instructions. The processor remains in lockup state until it is reset, an NMI occurs, or it is halted by a debugger.

Note: If the lockup state occurs from the NMI handler, a subsequent NMI does not cause the processor to leave the lockup state.

2.7 Power Management

The Cortex-M3 processor sleep modes reduce power consumption:

- Sleep mode stops the processor clock.
- Deep-sleep mode stops the system clock and switches off the PLL and Flash memory.

The SLEEPDEEP bit of the **System Control (SYSCTRL)** register selects which sleep mode is used (see page 126). For more information about the behavior of the sleep modes, see "System Control" on page 179.

This section describes the mechanisms for entering sleep mode and the conditions for waking up from sleep mode, both of which apply to Sleep mode and Deep-sleep mode.

2.7.1 Entering Sleep Modes

This section describes the mechanisms software can use to put the processor into one of the sleep modes.

The system can generate spurious wake-up events, for example a debug operation wakes up the processor. Therefore, software must be able to put the processor back into sleep mode after such an event. A program might have an idle loop to put the processor back to sleep mode.

2.7.1.1 Wait for Interrupt

The wait for interrupt instruction, WFI, causes immediate entry to sleep mode unless the wake-up condition is true (see "Wake Up from WFI or Sleep-on-Exit" on page 87). When the processor executes a WFI instruction, it stops executing instructions and enters sleep mode. See the Cortex™-M3/M4 Instruction Set Technical User's Manual for more information.

2.7.1.2 Wait for Event

The wait for event instruction, WFE, causes entry to sleep mode conditional on the value of a one-bit event register. When the processor executes a WFE instruction, it checks the event register. If the register is 0, the processor stops executing instructions and enters sleep mode. If the register is 1, the processor clears the register and continues executing instructions without entering sleep mode.

If the event register is 1, the processor must not enter sleep mode on execution of a WFE instruction. Typically, this situation occurs if an SEV instruction has been executed. Software cannot access this register directly.

See the Cortex™-M3/M4 Instruction Set Technical User's Manual for more information.

2.7.1.3 Sleep-on-Exit

If the SLEEPEXIT bit of the **SYSCTRL** register is set, when the processor completes the execution of all exception handlers, it returns to Thread mode and immediately enters sleep mode. This mechanism can be used in applications that only require the processor to run when an exception occurs.

2.7.2 Wake Up from Sleep Mode

The conditions for the processor to wake up depend on the mechanism that cause it to enter sleep mode.

2.7.2.1 Wake Up from WFI or Sleep-on-Exit

Normally, the processor wakes up only when the NVIC detects an exception with sufficient priority to cause exception entry. Some embedded systems might have to execute system restore tasks after the processor wakes up and before executing an interrupt handler. Entry to the interrupt handler can be delayed by setting the PRIMASK bit and clearing the FAULTMASK bit. If an interrupt arrives that is enabled and has a higher priority than current exception priority, the processor wakes up but does not execute the interrupt handler until the processor clears PRIMASK. For more information about **PRIMASK** and **FAULTMASK**, see page 64 and page 65.

2.7.2.2 Wake Up from WFE

The processor wakes up if it detects an exception with sufficient priority to cause exception entry.

In addition, if the SEVONPEND bit in the **SYSCTRL** register is set, any new pending interrupt triggers an event and wakes up the processor, even if the interrupt is disabled or has insufficient priority to cause exception entry. For more information about **SYSCTRL**, see page 126.

2.8 Instruction Set Summary

The processor implements a version of the Thumb instruction set. Table 2-13 on page 88 lists the supported instructions.

Note: In Table 2-13 on page 88:

- Angle brackets, <>, enclose alternative forms of the operand
- Braces, {}, enclose optional operands
- The Operands column is not exhaustive
- Op2 is a flexible second operand that can be either a register or a constant
- Most instructions can use an optional condition code suffix

For more information on the instructions and operands, see the instruction descriptions in the *Cortex™-M3/M4 Instruction Set Technical User's Manual*.

Table 2-13. Cortex-M3 Instruction Summary

Mnemonic	Operands	Brief Description	Flags
ADC, ADCS	{Rd,} Rn, Op2	Add with carry	N,Z,C,V
ADD, ADDS	{Rd,} Rn, Op2	Add	N,Z,C,V
ADD, ADDW	{Rd,} Rn , #imm12	Add	N,Z,C,V
ADR	Rd, label	Load PC-relative address	-
AND, ANDS	{Rd,} Rn, Op2	Logical AND	N,Z,C
ASR, ASRS	Rd, Rm, <rs #n></rs #n>	Arithmetic shift right	N,Z,C
В	label	Branch	-
BFC	Rd, #lsb, #width	Bit field clear	-
BFI	Rd, Rn, #lsb, #width	Bit field insert	-
BIC, BICS	{Rd,} Rn, Op2	Bit clear	N,Z,C
BKPT	#imm	Breakpoint	-
BL	label	Branch with link	-
BLX	Rm	Branch indirect with link	-
BX	Rm	Branch indirect	-
CBNZ	Rn, label	Compare and branch if non-zero	-
CBZ	Rn, label	Compare and branch if zero	-
CLREX	-	Clear exclusive	-
CLZ	Rd, Rm	Count leading zeros	-
CMN	Rn, Op2	Compare negative	N,Z,C,V
CMP	Rn, Op2	Compare	N,Z,C,V
CPSID	i	Change processor state, disable interrupts	-
CPSIE	i	Change processor state, enable interrupts	-
DMB	-	Data memory barrier	-
DSB	-	Data synchronization barrier	-

Table 2-13. Cortex-M3 Instruction Summary (continued)

Mnemonic	Operands	Brief Description	Flags
EOR, EORS	{Rd,} Rn, Op2	Exclusive OR	N,Z,C
ISB	-	Instruction synchronization barrier	-
IT	-	If-Then condition block	-
LDM	Rn{!}, reglist	Load multiple registers, increment after	-
LDMDB, LDMEA	Rn{!}, reglist	Load multiple registers, decrement before	-
LDMFD, LDMIA	Rn{!}, reglist	Load multiple registers, increment after	-
LDR	Rt, [Rn, #offset]	Load register with word	-
LDRB, LDRBT	Rt, [Rn, #offset]	Load register with byte	-
LDRD	Rt, Rt2, [Rn, #offset]	Load register with two bytes	-
LDREX	Rt, [Rn, #offset]	Load register exclusive	-
LDREXB	Rt, [Rn]	Load register exclusive with byte	-
LDREXH	Rt, [Rn]	Load register exclusive with halfword	-
LDRH, LDRHT	Rt, [Rn, #offset]	Load register with halfword	-
LDRSB, LDRSBT	Rt, [Rn, #offset]	Load register with signed byte	-
LDRSH, LDRSHT	Rt, [Rn, #offset]	Load register with signed halfword	-
LDRT	Rt, [Rn, #offset]	Load register with word	-
LSL, LSLS	Rd, Rm, <rs #n></rs #n>	Logical shift left	N,Z,C
LSR, LSRS	Rd, Rm, <rs #n></rs #n>	Logical shift right	N,Z,C
MLA	Rd, Rn, Rm, Ra	Multiply with accumulate, 32-bit result	-
MLS	Rd, Rn, Rm, Ra	Multiply and subtract, 32-bit result	-
MOV, MOVS	Rd, Op2	Move	N,Z,C
MOV, MOVW	Rd, #imm16	Move 16-bit constant	N,Z,C
TVOM	Rd, #imm16	Move top	-
MRS	Rd, spec_reg	Move from special register to general register	-
MSR	spec_reg, Rm	Move from general register to special register	N,Z,C,V
MUL, MULS	{Rd,} Rn, Rm	Multiply, 32-bit result	N,Z
MVN, MVNS	Rd, Op2	Move NOT	N,Z,C
10P	-	No operation	-
ORN, ORNS	{Rd,} Rn, Op2	Logical OR NOT	N,Z,C
ORR, ORRS	{Rd,} Rn, Op2	Logical OR	N,Z,C
POP	reglist	Pop registers from stack	-
PUSH	reglist	Push registers onto stack	-
RBIT	Rd, Rn	Reverse bits	-
REV	Rd, Rn	Reverse byte order in a word	-
REV16	Rd, Rn	Reverse byte order in each halfword	-
REVSH	Rd, Rn	Reverse byte order in bottom halfword and sign extend	-
ROR, RORS	Rd, Rm, <rs #n></rs #n>	Rotate right	N,Z,C
RRX, RRXS	Rd, Rm	Rotate right with extend	N,Z,C

Table 2-13. Cortex-M3 Instruction Summary (continued)

Mnemonic	Operands	Brief Description	Flags
RSB, RSBS	{Rd,} Rn, Op2	Reverse subtract	N,Z,C,V
SBC, SBCS	{Rd,} Rn, Op2	Subtract with carry	N,Z,C,V
SBFX	Rd, Rn, #lsb, #width	Signed bit field extract	-
SDIV	{Rd,} Rn, Rm	Signed divide	-
SEV	-	Send event	-
SMLAL	RdLo, RdHi, Rn, Rm	Signed multiply with accumulate (32x32+64), 64-bit result	-
SMULL	RdLo, RdHi, Rn, Rm	Signed multiply (32x32), 64-bit result	-
SSAT	Rd, #n, Rm {,shift #s}	Signed saturate	Q
STM	Rn{!}, reglist	Store multiple registers, increment after	-
STMDB, STMEA	Rn{!}, reglist	Store multiple registers, decrement before	-
STMFD, STMIA	Rn{!}, reglist	Store multiple registers, increment after	-
STR	Rt, [Rn {, #offset}]	Store register word	-
STRB, STRBT	Rt, [Rn {, #offset}]	Store register byte	-
STRD	Rt, Rt2, [Rn {, #offset}]	Store register two words	-
STREX	Rt, Rt, [Rn {, #offset}]	Store register exclusive	-
STREXB	Rd, Rt, [Rn]	Store register exclusive byte	-
STREXH	Rd, Rt, [Rn]	Store register exclusive halfword	-
STRH, STRHT	Rt, [Rn {, #offset}]	Store register halfword	-
STRSB, STRSBT	Rt, [Rn {, #offset}]	Store register signed byte	-
STRSH, STRSHT	Rt, [Rn {, #offset}]	Store register signed halfword	-
STRT	Rt, [Rn {, #offset}]	Store register word	-
SUB, SUBS	{Rd,} Rn, Op2	Subtract	N,Z,C,V
SUB, SUBW	{Rd,} Rn, #imm12	Subtract 12-bit constant	N,Z,C,V
SVC	#imm	Supervisor call	-
SXTB	{Rd,} Rm {,ROR #n}	Sign extend a byte	-
SXTH	{Rd,} Rm {,ROR #n}	Sign extend a halfword	-
TBB	[Rn, Rm]	Table branch byte	-
ТВН	[Rn, Rm, LSL #1]	Table branch halfword	-
TEQ	Rn, Op2	Test equivalence	N,Z,C
TST	Rn, Op2	Test	N,Z,C
UBFX	Rd, Rn, #lsb, #width	Unsigned bit field extract	-
UDIV	{Rd,} Rn, Rm	Unsigned divide	-
UMLAL	RdLo, RdHi, Rn, Rm	Unsigned multiply with accumulate (32x32+32+32), 64-bit result	-
UMULL	RdLo, RdHi, Rn, Rm	Unsigned multiply (32x 2), 64-bit result	-
USAT	Rd, #n, Rm {,shift #s}	Unsigned Saturate	Q
UXTB	{Rd,} Rm, {,ROR #n}	Zero extend a Byte	-
UXTH	{Rd,} Rm, {,ROR #n}	Zero extend a Halfword	-
WFE	-	Wait for event	-
WFI	-	Wait for interrupt	-

3 Cortex-M3 Peripherals

This chapter provides information on the Stellaris[®] implementation of the Cortex-M3 processor peripherals, including:

■ SysTick (see page 91)

Provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism.

- Nested Vectored Interrupt Controller (NVIC) (see page 92)
 - Facilitates low-latency exception and interrupt handling
 - Controls power management
 - Implements system control registers
- System Control Block (SCB) (see page 94)

Provides system implementation information and system control, including configuration, control, and reporting of system exceptions.

■ Memory Protection Unit (MPU) (see page 94)

Supports the standard ARMv7 Protected Memory System Architecture (PMSA) model. The MPU provides full support for protection regions, overlapping protection regions, access permissions, and exporting memory attributes to the system.

Table 3-1 on page 91 shows the address map of the Private Peripheral Bus (PPB). Some peripheral register regions are split into two address regions, as indicated by two addresses listed.

Table 3-1. Core Peripheral Register Regions

Address	Core Peripheral	Description (see page)
0xE000.E010-0xE000.E01F	System Timer	91
0xE000.E100-0xE000.E4EF	Nested Vectored Interrupt Controller	92
0xE000.EF00-0xE000.EF03		
0xE000.ED00-0xE000.ED3F	System Control Block	94
0xE000.ED90-0xE000.EDB8	Memory Protection Unit	94

3.1 Functional Description

This chapter provides information on the Stellaris implementation of the Cortex-M3 processor peripherals: SysTick, NVIC, SCB and MPU.

3.1.1 System Timer (SysTick)

Cortex-M3 includes an integrated system timer, SysTick, which provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example as:

- An RTOS tick timer that fires at a programmable rate (for example, 100 Hz) and invokes a SysTick routine.
- A high-speed alarm timer using the system clock.

- A variable rate alarm or signal timer—the duration is range-dependent on the reference clock used and the dynamic range of the counter.
- A simple counter used to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNT bit in the STCTRL control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

The timer consists of three registers:

- SysTick Control and Status (STCTRL): A control and status counter to configure its clock, enable the counter, enable the SysTick interrupt, and determine counter status.
- SysTick Reload Value (STRELOAD): The reload value for the counter, used to provide the counter's wrap value.
- SysTick Current Value (STCURRENT): The current value of the counter.

When enabled, the timer counts down on each clock from the reload value to zero, reloads (wraps) to the value in the **STRELOAD** register on the next clock edge, then decrements on subsequent clocks. Clearing the **STRELOAD** register disables the counter on the next wrap. When the counter reaches zero, the COUNT status bit is set. The COUNT bit clears on reads.

Writing to the **STCURRENT** register clears the register and the COUNT status bit. The write does not trigger the SysTick exception logic. On a read, the current value is the value of the register at the time the register is accessed.

The SysTick counter runs on the system clock. If this clock signal is stopped for low power mode, the SysTick counter stops. Ensure software uses aligned word accesses to access the SysTick registers.

Note: When the processor is halted for debugging, the counter does not decrement.

3.1.2 Nested Vectored Interrupt Controller (NVIC)

This section describes the Nested Vectored Interrupt Controller (NVIC) and the registers it uses. The NVIC supports:

- 29 interrupts.
- A programmable priority level of 0-7 for each interrupt. A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Low-latency exception and interrupt handling.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and subpriority fields.
- Interrupt tail-chaining.
- An external Non-maskable interrupt (NMI).

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead, providing low latency exception handling.

3.1.2.1 Level-Sensitive and Pulse Interrupts

The processor supports both level-sensitive and pulse interrupts. Pulse interrupts are also described as edge-triggered interrupts.

A level-sensitive interrupt is held asserted until the peripheral deasserts the interrupt signal. Typically this happens because the ISR accesses the peripheral, causing it to clear the interrupt request. A pulse interrupt is an interrupt signal sampled synchronously on the rising edge of the processor clock. To ensure the NVIC detects the interrupt, the peripheral must assert the interrupt signal for at least one clock cycle, during which the NVIC detects the pulse and latches the interrupt.

When the processor enters the ISR, it automatically removes the pending state from the interrupt (see "Hardware and Software Control of Interrupts" on page 93 for more information). For a level-sensitive interrupt, if the signal is not deasserted before the processor returns from the ISR, the interrupt becomes pending again, and the processor must execute its ISR again. As a result, the peripheral can hold the interrupt signal asserted until it no longer needs servicing.

3.1.2.2 Hardware and Software Control of Interrupts

The Cortex-M3 latches all interrupts. A peripheral interrupt becomes pending for one of the following reasons:

- The NVIC detects that the interrupt signal is High and the interrupt is not active.
- The NVIC detects a rising edge on the interrupt signal.
- Software writes to the corresponding interrupt set-pending register bit, or to the **Software Trigger Interrupt (SWTRIG)** register to make a Software-Generated Interrupt pending. See the INT bit in the **PEND0** register on page 110 or **SWTRIG** on page 118.

A pending interrupt remains pending until one of the following:

- The processor enters the ISR for the interrupt, changing the state of the interrupt from pending to active. Then:
 - For a level-sensitive interrupt, when the processor returns from the ISR, the NVIC samples
 the interrupt signal. If the signal is asserted, the state of the interrupt changes to pending,
 which might cause the processor to immediately re-enter the ISR. Otherwise, the state of the
 interrupt changes to inactive.
 - For a pulse interrupt, the NVIC continues to monitor the interrupt signal, and if this is pulsed
 the state of the interrupt changes to pending and active. In this case, when the processor
 returns from the ISR the state of the interrupt changes to pending, which might cause the
 processor to immediately re-enter the ISR.
 - If the interrupt signal is not pulsed while the processor is in the ISR, when the processor returns from the ISR the state of the interrupt changes to inactive.
- Software writes to the corresponding interrupt clear-pending register bit
 - For a level-sensitive interrupt, if the interrupt signal is still asserted, the state of the interrupt does not change. Otherwise, the state of the interrupt changes to inactive.

For a pulse interrupt, the state of the interrupt changes to inactive, if the state was pending
or to active, if the state was active and pending.

3.1.3 System Control Block (SCB)

The System Control Block (SCB) provides system implementation information and system control, including configuration, control, and reporting of the system exceptions.

3.1.4 Memory Protection Unit (MPU)

This section describes the Memory protection unit (MPU). The MPU divides the memory map into a number of regions and defines the location, size, access permissions, and memory attributes of each region. The MPU supports independent attribute settings for each region, overlapping regions, and export of memory attributes to the system.

The memory attributes affect the behavior of memory accesses to the region. The Cortex-M3 MPU defines eight separate memory regions, 0-7, and a background region.

When memory regions overlap, a memory access is affected by the attributes of the region with the highest number. For example, the attributes for region 7 take precedence over the attributes of any region that overlaps region 7.

The background region has the same memory access attributes as the default memory map, but is accessible from privileged software only.

The Cortex-M3 MPU memory map is unified, meaning that instruction accesses and data accesses have the same region settings.

If a program accesses a memory location that is prohibited by the MPU, the processor generates a memory management fault, causing a fault exception and possibly causing termination of the process in an OS environment. In an OS environment, the kernel can update the MPU region setting dynamically based on the process to be executed. Typically, an embedded OS uses the MPU for memory protection.

Configuration of MPU regions is based on memory types (see "Memory Regions, Types and Attributes" on page 69 for more information).

Table 3-2 on page 94 shows the possible MPU region attributes. See the section called "MPU Configuration for a Stellaris Microcontroller" on page 98 for guidelines for programming a microcontroller implementation.

Table 3-2. Memory Attributes Summary

Memory Type	Description
Strongly Ordered	All accesses to Strongly Ordered memory occur in program order.
Device	Memory-mapped peripherals
Normal	Normal memory

To avoid unexpected behavior, disable the interrupts before updating the attributes of a region that the interrupt handlers might access.

Ensure software uses aligned accesses of the correct size to access MPU registers:

- Except for the MPU Region Attribute and Size (MPUATTR) register, all MPU registers must be accessed with aligned word accesses.
- The MPUATTR register can be accessed with byte or aligned halfword or word accesses.

The processor does not support unaligned accesses to MPU registers.

When setting up the MPU, and if the MPU has previously been programmed, disable unused regions to prevent any previous region settings from affecting the new MPU setup.

3.1.4.1 Updating an MPU Region

To update the attributes for an MPU region, the MPU Region Number (MPUNUMBER), MPU Region Base Address (MPUBASE) and MPUATTR registers must be updated. Each register can be programmed separately or with a multiple-word write to program all of these registers. You can use the MPUBASEx and MPUATTRx aliases to program up to four regions simultaneously using an STM instruction.

Updating an MPU Region Using Separate Words

This example simple code configures one region:

Disable a region before writing new region settings to the MPU if you have previously enabled the region being changed. For example:

```
; R1 = region number
; R2 = size/enable
; R3 = attributes
; R4 = address
                        ; 0xE000ED98, MPU region number register ; Region Number
LDR R0,=MPUNUMBER
STR R1, [R0, #0x0]
BIC R2, R2, #1
                          ; Disable
STRH R2, [R0, #0x8]
STR R4, [R0, #0x4]
                          ; Region Size and Enable
STR R4, [R0, #0x4]
                          ; Region Base Address
STRH R3, [R0, #0xA]
                          ; Region Attribute
ORR R2, #1
                           ; Enable
STRH R2, [R0, #0x8]
                           ; Region Size and Enable
```

Software must use memory barrier instructions:

- Before MPU setup, if there might be outstanding memory transfers, such as buffered writes, that might be affected by the change in MPU settings.
- After MPU setup, if it includes memory transfers that must use the new MPU settings.

However, memory barrier instructions are not required if the MPU setup process starts by entering an exception handler, or is followed by an exception return, because the exception entry and exception return mechanism cause memory barrier behavior.

Software does not need any memory barrier instructions during MPU setup, because it accesses the MPU through the Private Peripheral Bus (PPB), which is a Strongly Ordered memory region.

For example, if all of the memory access behavior is intended to take effect immediately after the programming sequence, then a DSB instruction and an ISB instruction should be used. A DSB is required after changing MPU settings, such as at the end of context switch. An ISB is required if the code that programs the MPU region or regions is entered using a branch or call. If the programming sequence is entered using a return from exception, or by taking an exception, then an ISB is not required.

Updating an MPU Region Using Multi-Word Writes

The MPU can be programmed directly using multi-word writes, depending how the information is divided. Consider the following reprogramming:

```
; R1 = region number
; R2 = address
; R3 = size, attributes in one
LDR R0, =MPUNUMBER ; 0xE000ED98, MPU region number register
STR R1, [R0, #0x0] ; Region Number
STR R2, [R0, #0x4] ; Region Base Address
STR R3, [R0, #0x8] ; Region Attribute, Size and Enable
```

An STM instruction can be used to optimize this:

```
; R1 = region number
; R2 = address
; R3 = size, attributes in one
LDR R0, =MPUNUMBER ; 0xE000ED98, MPU region number register
STM R0, {R1-R3} ; Region number, address, attribute, size and enable
```

This operation can be done in two words for pre-packed information, meaning that the **MPU Region Base Address (MPUBASE)** register (see page 150) contains the required region number and has the VALID bit set. This method can be used when the data is statically packed, for example in a boot loader:

Subregions

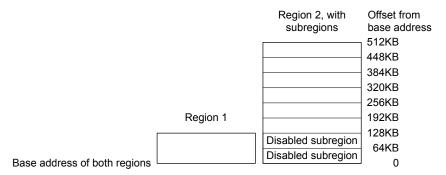
Regions of 256 bytes or more are divided into eight equal-sized subregions. Set the corresponding bit in the SRD field of the **MPU Region Attribute and Size (MPUATTR)** register (see page 152) to disable a subregion. The least-significant bit of the SRD field controls the first subregion, and the most-significant bit controls the last subregion. Disabling a subregion means another region overlapping the disabled range matches instead. If no other enabled region overlaps the disabled subregion, the MPU issues a fault.

Regions of 32, 64, and 128 bytes do not support subregions. With regions of these sizes, the SRD field must be configured to 0×0.0 , otherwise the MPU behavior is unpredictable.

Example of SRD Use

Two regions with the same base address overlap. Region one is 128 KB, and region two is 512 KB. To ensure the attributes from region one apply to the first 128 KB region, configure the SRD field for region two to 0x03 to disable the first two subregions, as Figure 3-1 on page 97 shows.

Figure 3-1. SRD Use Example



3.1.4.2 MPU Access Permission Attributes

The access permission bits, TEX, S, C, B, AP, and XN of the **MPUATTR** register, control access to the corresponding memory region. If an access is made to an area of memory without the required permissions, then the MPU generates a permission fault.

Table 3-3 on page 97 shows the encodings for the \mathtt{TEX} , \mathtt{C} , \mathtt{B} , and \mathtt{S} access permission bits. All encodings are shown for completeness, however the current implementation of the Cortex-M3 does not support the concept of cacheability or shareability. Refer to the section called "MPU Configuration for a Stellaris Microcontroller" on page 98 for information on programming the MPU for Stellaris implementations.

Table 3-3. TEX, S, C, and B Bit Field Encoding

TEX	S	С	В	Memory Type	Shareability	Other Attributes
000b	x ^a	0	0	Strongly Ordered	Shareable	-
000	x ^a	0	1	Device	Shareable	-
000	0	1	0	Normal	Not shareable	
000	1	1	0	Normal	Shareable	Outer and inner
000	0	1	1	Normal	Not shareable	write-through. No write allocate.
000	1	1	1	Normal	Shareable	
001	0	0	0	Normal	Not shareable	Outer and inner
001	1	0	0	Normal	Shareable	noncacheable.
001	x ^a	0	1	Reserved encoding	-	-
001	x ^a	1	0	Reserved encoding	-	-
001	0	1	1	Normal	Not shareable	Outer and inner
001	1	1	1	Normal	Shareable	write-back. Write and read allocate.
010	x ^a	0	0	Device	Not shareable	Nonshared Device.
010	x ^a	0	1	Reserved encoding	-	-
010	x ^a	1	x ^a	Reserved encoding	-	-

Table 3-3. TEX, S, C, and B Bit Field Encoding (continued)

TEX	s	С	В	Memory Type	Shareability	Other Attributes
1BB	0	Α	Α	Normal	Not shareable	Cached memory (BB =
1BB	1	А	А	Normal	Shareable	outer policy, AA = inner policy).
						See Table 3-4 for the encoding of the AA and BB bits.

a. The MPU ignores the value of this bit.

Table 3-4 on page 98 shows the cache policy for memory attribute encodings with a TEX value in the range of 0x4-0x7.

Table 3-4. Cache Policy for Memory Attribute Encoding

Encoding, AA or BB	Corresponding Cache Policy
00	Non-cacheable
01	Write back, write and read allocate
10	Write through, no write allocate
11	Write back, no write allocate

Table 3-5 on page 98 shows the AP encodings in the **MPUATTR** register that define the access permissions for privileged and unprivileged software.

Table 3-5. AP Bit Field Encoding

AP Bit Field	Privileged Permissions	Unprivileged Permissions	Description
000	No access	No access	All accesses generate a permission fault.
001	R/W	No access	Access from privileged software only.
010	R/W	RO	Writes by unprivileged software generate a permission fault.
011	R/W	R/W	Full access.
100	Unpredictable	Unpredictable	Reserved.
101	RO	No access	Reads by privileged software only.
110	RO	RO	Read-only, by privileged or unprivileged software.
111	RO	RO	Read-only, by privileged or unprivileged software.

MPU Configuration for a Stellaris Microcontroller

Stellaris microcontrollers have only a single processor and no caches. As a result, the MPU should be programmed as shown in Table 3-6 on page 98.

Table 3-6. Memory Region Attributes for Stellaris Microcontrollers

Memory Region	TEX	S	С	В	Memory Type and Attributes
Flash memory	000b	0	1	0	Normal memory, non-shareable, write-through
Internal SRAM	000b	1	1	0	Normal memory, shareable, write-through
External SRAM	000b	1	1	1	Normal memory, shareable, write-back, write-allocate
Peripherals	000b	1	0	1	Device memory, shareable

In current Stellaris microcontroller implementations, the shareability and cache policy attributes do not affect the system behavior. However, using these settings for the MPU regions can make the application code more portable. The values given are for typical situations.

3.1.4.3 MPU Mismatch

When an access violates the MPU permissions, the processor generates a memory management fault (see "Exceptions and Interrupts" on page 68 for more information). The **MFAULTSTAT** register indicates the cause of the fault. See page 137 for more information.

3.2 Register Map

Table 3-7 on page 99 lists the Cortex-M3 Peripheral SysTick, NVIC, MPU and SCB registers. The offset listed is a hexadecimal increment to the register's address, relative to the Core Peripherals base address of 0xE000.E000.

Note: Register spaces that are not used are reserved for future or internal use. Software should not modify any reserved memory address.

Table 3-7. Peripherals Register Map

Offset	Name	Туре	Reset	Description	See page						
System Timer (SysTick) Registers											
0x010	STCTRL	R/W	0x0000.0000	SysTick Control and Status Register	102						
0x014	STRELOAD	R/W	0x0000.0000	SysTick Reload Value Register	104						
0x018	STCURRENT	R/WC	0x0000.0000	SysTick Current Value Register	105						
Nested V	ectored Interrupt Control	ler (NVIC)	Registers		-						
0x100	EN0	R/W	0x0000.0000	Interrupt 0-31 Set Enable	106						
0x104	EN1	R/W	0x0000.0000	Interrupt 32-43 Set Enable	107						
0x180	DIS0	R/W	0x0000.0000	Interrupt 0-31 Clear Enable	108						
0x184	DIS1	R/W	0x0000.0000	Interrupt 32-43 Clear Enable	109						
0x200	PEND0	R/W	0x0000.0000	Interrupt 0-31 Set Pending	110						
0x204	PEND1	R/W	0x0000.0000	Interrupt 32-43 Set Pending	111						
0x280	UNPEND0	R/W	0x0000.0000	Interrupt 0-31 Clear Pending	112						
0x284	UNPEND1	R/W	0x0000.0000	Interrupt 32-43 Clear Pending	113						
0x300	ACTIVE0	RO	0x0000.0000	Interrupt 0-31 Active Bit	114						
0x304	ACTIVE1	RO	0x0000.0000	Interrupt 32-43 Active Bit	115						
0x400	PRI0	R/W	0x0000.0000	Interrupt 0-3 Priority	116						
0x404	PRI1	R/W	0x0000.0000	Interrupt 4-7 Priority	116						
0x408	PRI2	R/W	0x0000.0000	Interrupt 8-11 Priority	116						
0x40C	PRI3	R/W	0x0000.0000	Interrupt 12-15 Priority	116						
0x410	PRI4	R/W	0x0000.0000	Interrupt 16-19 Priority	116						

Table 3-7. Peripherals Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0x414	PRI5	R/W	0x0000.0000	Interrupt 20-23 Priority	116
0x418	PRI6	R/W	0x0000.0000	Interrupt 24-27 Priority	116
0x41C	PRI7	R/W	0x0000.0000	Interrupt 28-31 Priority	116
0x420	PRI8	R/W	0x0000.0000	Interrupt 32-35 Priority	116
0x424	PRI9	R/W	0x0000.0000	Interrupt 36-39 Priority	116
0x428	PRI10	R/W	0x0000.0000	Interrupt 40-43 Priority	116
0xF00	SWTRIG	WO	0x0000.0000	Software Trigger Interrupt	118
System C	ontrol Block (SCB) Regis	sters			
0xD00	CPUID	RO	0x411F.C231	CPU ID Base	119
0xD04	INTCTRL	R/W	0x0000.0000	Interrupt Control and State	120
0xD08	VTABLE	R/W	0x0000.0000	Vector Table Offset	123
0xD0C	APINT	R/W	0xFA05.0000	Application Interrupt and Reset Control	124
0xD10	SYSCTRL	R/W	0x0000.0000	System Control	126
0xD14	CFGCTRL	R/W	0x0000.0000	Configuration and Control	128
0xD18	SYSPRI1	R/W	0x0000.0000	System Handler Priority 1	130
0xD1C	SYSPRI2	R/W	0x0000.0000	System Handler Priority 2	131
0xD20	SYSPRI3	R/W	0x0000.0000	System Handler Priority 3	132
0xD24	SYSHNDCTRL	R/W	0x0000.0000	System Handler Control and State	133
0xD28	FAULTSTAT	R/W1C	0x0000.0000	Configurable Fault Status	137
0xD2C	HFAULTSTAT	R/W1C	0x0000.0000	Hard Fault Status	143
0xD34	MMADDR	R/W	-	Memory Management Fault Address	144
0xD38	FAULTADDR	R/W	-	Bus Fault Address	145
Memory F	Protection Unit (MPU) Reg	gisters			<u>'</u>
0xD90	MPUTYPE	RO	0x0000.0800	MPU Type	146
0xD94	MPUCTRL	R/W	0x0000.0000	MPU Control	147
0xD98	MPUNUMBER	R/W	0x0000.0000	MPU Region Number	149
0xD9C	MPUBASE	R/W	0x0000.0000	MPU Region Base Address	150
0xDA0	MPUATTR	R/W	0x0000.0000	MPU Region Attribute and Size	152
0xDA4	MPUBASE1	R/W	0x0000.0000	MPU Region Base Address Alias 1	150
0xDA8	MPUATTR1	R/W	0x0000.0000	MPU Region Attribute and Size Alias 1	152
0xDAC	MPUBASE2	R/W	0x0000.0000	MPU Region Base Address Alias 2	150
0xDB0	MPUATTR2	R/W	0x0000.0000	MPU Region Attribute and Size Alias 2	152

Table 3-7. Peripherals Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0xDB4	MPUBASE3	R/W	0x0000.0000	MPU Region Base Address Alias 3	150
0xDB8	MPUATTR3	R/W	0x0000.0000	MPU Region Attribute and Size Alias 3	152

3.3 System Timer (SysTick) Register Descriptions

This section lists and describes the System Timer registers, in numerical order by address offset.

Register 1: SysTick Control and Status Register (STCTRL), offset 0x010

Note: This register can only be accessed from privileged mode.

The SysTick **STCTRL** register enables the SysTick features.

SysTick Control and Status Register (STCTRL)

Base 0xE000.E000 Offset 0x010 Type R/W, reset 0x0000.0000

.,,,,	,															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			'		' '	!		reserved		'	•	•		•		COUNT
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
110001	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ſ	15	14	13	12	'''	10	T reserved	0				-		CLK_SRC	INTEN	ENABLE
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	sit/Field		Nam	ne		Туре		et Description								
	31:17		reserv	/ed	R	0	0x000	Software should not rely on the value of a reserved bit. To p compatibility with future products, the value of a reserved bit preserved across a read-modify-write operation.				•				
	16		COU	NT	R	0	0	Cou	nt Flag							
								Valu	ue	Descrip	otion					
								0 The SysTick timer has not counted to 0 this bit was read.			ed to 0 sir	nce the I	ast time			
								1 The SysTick timer has counted to 0 since this bit was read.				e the las	t time			
										ared by h any va		the regis	ster or if	the STCU	IRRENT	register
								If read by the debugger using the DAP, this bit is cleared only MasterType bit in the AHB-AP Control Register is clear. Of the COUNT bit is not changed by the debugger read. See the A Debug Interface V5 Architecture Specification for more inform MasterType.					clear. Ot ee the A	herwise, I <i>RM</i> ®		
	15:3		reserv	/ed	R	0	0x000	Software should not rely on the value of a reserved bit compatibility with future products, the value of a reserv preserved across a read-modify-write operation.								
	2		CLK_S	SRC	R/	W	0	Cloc	ck Sourc	е						
								Valu	ue Desc	cription						
								0		rnal refe ocontroll		ock. (Not	implem	ented for	most S	tellaris
								1	Syste	em clock	(

Because an external reference clock is not implemented, this bit must

be set in order for SysTick to operate.

Bit/Field	Name	Туре	Reset	Description	on
1	INTEN	R/W	0	Interrupt Enable	
				Value	Description
				0	Interrupt generation is disabled. Software can use the COUNT bit to determine if the counter has ever reached 0.
				1	An interrupt is generated to the NVIC when SysTick counts to 0.
0	ENABLE	R/W	0	Enable	
				Value	Description
				0	The counter is disabled.
				1	Enables SysTick to operate in a multi-shot way. That is, the counter loads the RELOAD value and begins counting down. On reaching 0, the COUNT bit is set and an interrupt is generated if enabled by INTEN. The counter then loads the RELOAD value again and begins counting.

Register 2: SysTick Reload Value Register (STRELOAD), offset 0x014

Note: This register can only be accessed from privileged mode.

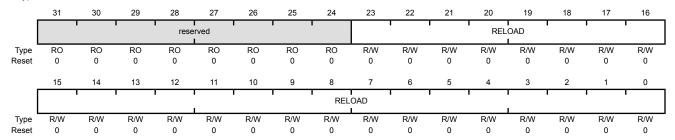
The **STRELOAD** register specifies the start value to load into the **SysTick Current Value** (**STCURRENT**) register when the counter reaches 0. The start value can be between 0x1 and 0x00FF.FFFF. A start value of 0 is possible but has no effect because the SysTick interrupt and the COUNT bit are activated when counting from 1 to 0.

SysTick can be configured as a multi-shot timer, repeated over and over, firing every N+1 clock pulses, where N is any value from 1 to 0x00FF.FFFF. For example, if a tick interrupt is required every 100 clock pulses, 99 must be written into the RELOAD field.

SysTick Reload Value Register (STRELOAD)

Base 0xE000.E000

Offset 0x014 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:24	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
23:0	RELOAD	R/W	0x00.0000	Reload Value

Value to load into the $\mbox{\bf SysTick}$ Current Value (STCURRENT) register when the counter reaches 0.

Register 3: SysTick Current Value Register (STCURRENT), offset 0x018

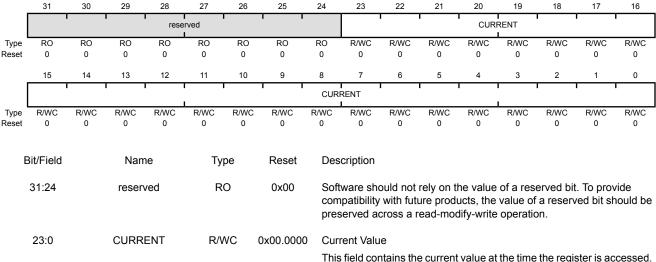
Note: This register can only be accessed from privileged mode.

The **STCURRENT** register contains the current value of the SysTick counter.

SysTick Current Value Register (STCURRENT)

Base 0xE000.E000 Offset 0x018

Type R/WC, reset 0x0000.0000



No read-modify-write protection is provided, so change with care.

This register is write-clear. Writing to it with any value clears the register.

This register is write-clear. Writing to it with any value clears the register. Clearing this register also clears the COUNT bit of the **STCTRL** register.

3.4 NVIC Register Descriptions

This section lists and describes the NVIC registers, in numerical order by address offset.

The NVIC registers can only be fully accessed from privileged mode, but interrupts can be pended while in unprivileged mode by enabling the **Configuration and Control (CFGCTRL)** register. Any other unprivileged mode access causes a bus fault.

Ensure software uses correctly aligned register accesses. The processor does not support unaligned accesses to NVIC registers.

An interrupt can enter the pending state even if it is disabled.

Before programming the **VTABLE** register to relocate the vector table, ensure the vector table entries of the new vector table are set up for fault handlers, NMI, and all enabled exceptions such as interrupts. For more information, see page 123.

Register 4: Interrupt 0-31 Set Enable (EN0), offset 0x100

Note: This register can only be accessed from privileged mode.

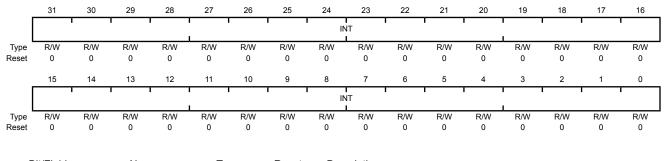
The **EN0** register enables interrupts and shows which interrupts are enabled. Bit 0 corresponds to Interrupt 0; bit 31 corresponds to Interrupt 31.

See Table 2-9 on page 79 for interrupt assignments.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

Interrupt 0-31 Set Enable (EN0)

Base 0xE000.E000 Offset 0x100 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:0	INT	R/W	0x0000.0000	Interrupt Enable

Value	Description
0	On a read, indicates the interrupt is disabled.
	On a write, no effect.
1	On a read, indicates the interrupt is enabled.
	On a write, enables the interrupt.

A bit can only be cleared by setting the corresponding ${\tt INT[n]}$ bit in the **DISn** register.

Register 5: Interrupt 32-43 Set Enable (EN1), offset 0x104

Note: This register can only be accessed from privileged mode.

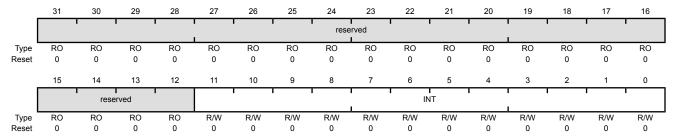
The **EN1** register enables interrupts and shows which interrupts are enabled. Bit 0 corresponds to Interrupt 32; bit 11 corresponds to Interrupt 43. See Table 2-9 on page 79 for interrupt assignments.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

Interrupt 32-43 Set Enable (EN1)

Base 0xE000.E000 Offset 0x104

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	INT	R/W	0x000	Interrupt Enable

value	Description
0	On a read, indicates the interrupt is disabled.
	On a write, no effect.
1	On a read, indicates the interrupt is enabled.
	On a write, enables the interrupt.

A bit can only be cleared by setting the corresponding ${\tt INT[n]}$ bit in the **DIS1** register.

Register 6: Interrupt 0-31 Clear Enable (DIS0), offset 0x180

Note: This register can only be accessed from privileged mode.

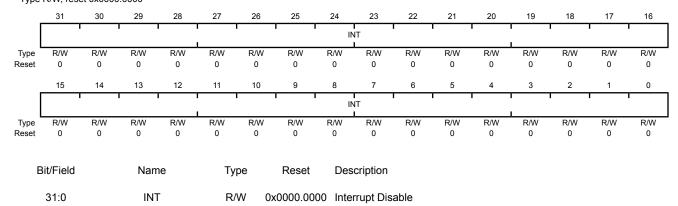
The **DIS0** register disables interrupts. Bit 0 corresponds to Interrupt 0; bit 31 corresponds to Interrupt 31.

See Table 2-9 on page 79 for interrupt assignments.

Interrupt 0-31 Clear Enable (DIS0)

Base 0xE000.E000 Offset 0x180

Type R/W, reset 0x0000.0000



Value Description

0 On a read, indicates the interrupt is disabled.

On a write, no effect.

1 On a read, indicates the interrupt is enabled.

On a write, clears the corresponding ${\tt INT[n]}$ bit in the EN0 register, disabling interrupt [n].

Register 7: Interrupt 32-43 Clear Enable (DIS1), offset 0x184

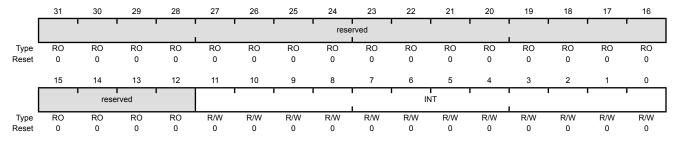
Note: This register can only be accessed from privileged mode.

The **DIS1** register disables interrupts. Bit 0 corresponds to Interrupt 32; bit 11 corresponds to Interrupt 43. See Table 2-9 on page 79 for interrupt assignments.

Interrupt 32-43 Clear Enable (DIS1)

Base 0xE000.E000

Offset 0x184
Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	INT	R/W	0x000	Interrupt Disable

- On a read, indicates the interrupt is disabled. On a write, no effect.
 - On a read, indicates the interrupt is enabled. On a write, clears the corresponding INT[n] bit in the EN1 register, disabling interrupt [n].

Register 8: Interrupt 0-31 Set Pending (PEND0), offset 0x200

Note: This register can only be accessed from privileged mode.

The **PEND0** register forces interrupts into the pending state and shows which interrupts are pending. Bit 0 corresponds to Interrupt 0; bit 31 corresponds to Interrupt 31.

See Table 2-9 on page 79 for interrupt assignments.

R/W

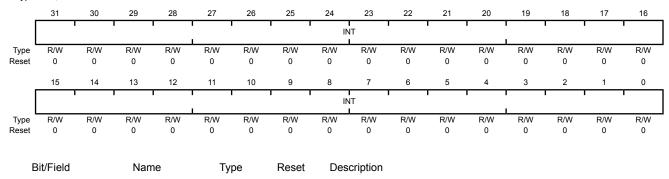
Interrupt 0-31 Set Pending (PEND0)

INT

Base 0xE000.E000 Offset 0x200

31.0

Type R/W, reset 0x0000.0000



00	 	0,,0000,000		or chamig
			Value	Description
			0	On a read, indicates that the interrupt is not pending.
				On a write, no effect.

0x0000 0000 Interrupt Set Pending

On a read, indicates that the interrupt is pending.
 On a write, the corresponding interrupt is set to pending even if it is disabled.

If the corresponding interrupt is already pending, setting a bit has no effect

A bit can only be cleared by setting the corresponding ${\tt INT[n]}$ bit in the <code>UNPENDO</code> register.

Register 9: Interrupt 32-43 Set Pending (PEND1), offset 0x204

Note: This register can only be accessed from privileged mode.

The **PEND1** register forces interrupts into the pending state and shows which interrupts are pending. Bit 0 corresponds to Interrupt 32; bit 11 corresponds to Interrupt 43. See Table 2-9 on page 79 for interrupt assignments.

Interrupt 32-43 Set Pending (PEND1)

Base 0xE000.E000 Offset 0x204

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11.0	INT	R/W	0x000	Interrupt Set Pending

Value	Description
0	On a read, indicates that the interrupt is not pending.
	On a write, no effect.
1	On a read, indicates that the interrupt is pending.
	On a write, the corresponding interrupt is set to pending even if it is disabled.

If the corresponding interrupt is already pending, setting a bit has no effect.

A bit can only be cleared by setting the corresponding ${\tt INT[n]}$ bit in the <code>UNPEND1</code> register.

Register 10: Interrupt 0-31 Clear Pending (UNPEND0), offset 0x280

Note: This register can only be accessed from privileged mode.

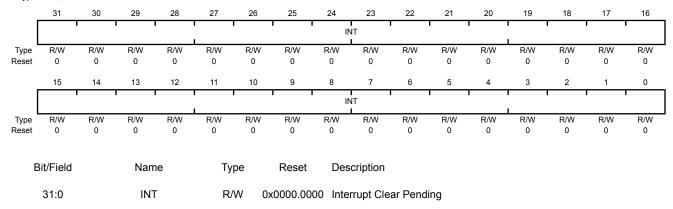
The **UNPEND0** register shows which interrupts are pending and removes the pending state from interrupts. Bit 0 corresponds to Interrupt 0; bit 31 corresponds to Interrupt 31.

See Table 2-9 on page 79 for interrupt assignments.

Interrupt 0-31 Clear Pending (UNPEND0)

Base 0xE000.E000 Offset 0x280

Type R/W, reset 0x0000.0000



- On a read, indicates that the interrupt is not pending. On a write, no effect.
- On a read, indicates that the interrupt is pending.

 On a write, clears the corresponding INT[n] bit in the **PEND0** register, so that interrupt [n] is no longer pending.

 Setting a bit does not affect the active state of the corresponding interrupt.

Register 11: Interrupt 32-43 Clear Pending (UNPEND1), offset 0x284

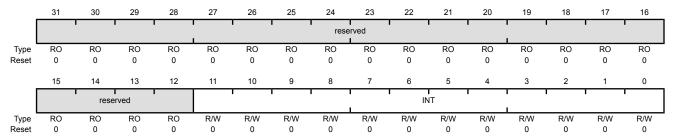
Note: This register can only be accessed from privileged mode.

The **UNPEND1** register shows which interrupts are pending and removes the pending state from interrupts. Bit 0 corresponds to Interrupt 32; bit 11 corresponds to Interrupt 43. See Table 2-9 on page 79 for interrupt assignments.

Interrupt 32-43 Clear Pending (UNPEND1)

Base 0xE000.E000 Offset 0x284

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	INT	R/W	0x000	Interrupt Clear Pending

- On a read, indicates that the interrupt is not pending. On a write, no effect.
- On a read, indicates that the interrupt is pending.

 On a write, clears the corresponding INT[n] bit in the **PEND1** register, so that interrupt [n] is no longer pending.

 Setting a bit does not affect the active state of the corresponding interrupt.

Register 12: Interrupt 0-31 Active Bit (ACTIVE0), offset 0x300

Note: This register can only be accessed from privileged mode.

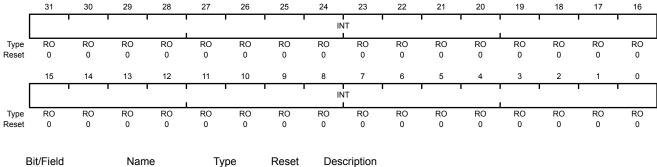
The ACTIVEO register indicates which interrupts are active. Bit 0 corresponds to Interrupt 0; bit 31 corresponds to Interrupt 31.

See Table 2-9 on page 79 for interrupt assignments.

Caution – Do not manually set or clear the bits in this register.

Interrupt 0-31 Active Bit (ACTIVE0)

Base 0xE000.E000 Offset 0x300 Type RO, reset 0x0000.0000



INT 31:0 RO 0x0000.0000 Interrupt Active

- 0 The corresponding interrupt is not active.
- The corresponding interrupt is active, or active and pending.

Register 13: Interrupt 32-43 Active Bit (ACTIVE1), offset 0x304

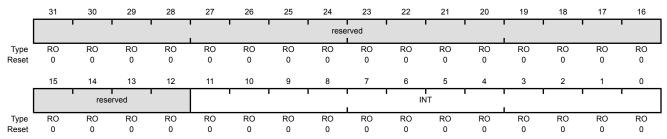
Note: This register can only be accessed from privileged mode.

The ACTIVE1 register indicates which interrupts are active. Bit 0 corresponds to Interrupt 32; bit 11 corresponds to Interrupt 43. See Table 2-9 on page 79 for interrupt assignments.

Caution – Do not manually set or clear the bits in this register.

Interrupt 32-43 Active Bit (ACTIVE1)

Base 0xE000.E000 Offset 0x304 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	INT	RO	0x000	Interrupt Active

- 0 The corresponding interrupt is not active.
- 1 The corresponding interrupt is active, or active and pending.

Register 14: Interrupt 0-3 Priority (PRI0), offset 0x400

Register 15: Interrupt 4-7 Priority (PRI1), offset 0x404

Register 16: Interrupt 8-11 Priority (PRI2), offset 0x408

Register 17: Interrupt 12-15 Priority (PRI3), offset 0x40C

Register 18: Interrupt 16-19 Priority (PRI4), offset 0x410

Register 19: Interrupt 20-23 Priority (PRI5), offset 0x414

Register 20: Interrupt 24-27 Priority (PRI6), offset 0x418

Register 21: Interrupt 28-31 Priority (PRI7), offset 0x41C

Register 22: Interrupt 32-35 Priority (PRI8), offset 0x420

Register 23: Interrupt 36-39 Priority (PRI9), offset 0x424

Register 24: Interrupt 40-43 Priority (PRI10), offset 0x428

Note: This register can only be accessed from privileged mode.

The **PRIn** registers provide 3-bit priority fields for each interrupt. These registers are byte accessible. Each register holds four priority fields that are assigned to interrupts as follows:

PRIn Register Bit Field	Interrupt
Bits 31:29	Interrupt [4n+3]
Bits 23:21	Interrupt [4n+2]
Bits 15:13	Interrupt [4n+1]
Bits 7:5	Interrupt [4n]

See Table 2-9 on page 79 for interrupt assignments.

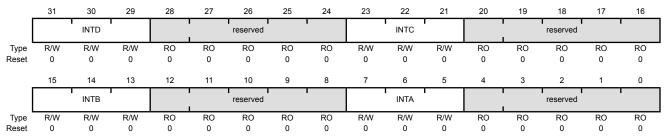
Each priority level can be split into separate group priority and subpriority fields. The PRIGROUP field in the **Application Interrupt and Reset Control (APINT)** register (see page 124) indicates the position of the binary point that splits the priority and subpriority fields.

These registers can only be accessed from privileged mode.

Interrupt 0-3 Priority (PRI0)

Base 0xE000.E000 Offset 0x400

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:29	INTD	R/W	0x0	Interrupt Priority for Interrupt [4n+3] This field holds a priority value, 0-7, for the interrupt with the number [4n+3], where n is the number of the Interrupt Priority register (n=0 for PRI0 , and so on). The lower the value, the greater the priority of the corresponding interrupt.
28:24	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
23:21	INTC	R/W	0x0	Interrupt Priority for Interrupt [4n+2] This field holds a priority value, 0-7, for the interrupt with the number [4n+2], where n is the number of the Interrupt Priority register (n=0 for PRIO , and so on). The lower the value, the greater the priority of the corresponding interrupt.
20:16	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:13	INTB	R/W	0x0	Interrupt Priority for Interrupt [4n+1] This field holds a priority value, 0-7, for the interrupt with the number [4n+1], where n is the number of the Interrupt Priority register (n=0 for PRIO , and so on). The lower the value, the greater the priority of the corresponding interrupt.
12:8	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:5	INTA	R/W	0x0	Interrupt Priority for Interrupt [4n] This field holds a priority value, 0-7, for the interrupt with the number [4n], where n is the number of the Interrupt Priority register (n=0 for PRIO , and so on). The lower the value, the greater the priority of the corresponding interrupt.
4:0	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 25: Software Trigger Interrupt (SWTRIG), offset 0xF00

Note: Only privileged software can enable unprivileged access to the SWTRIG register.

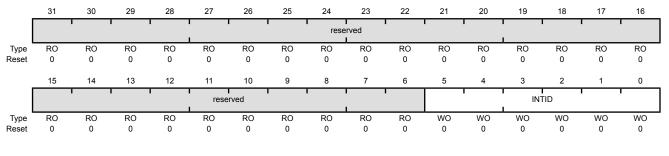
Writing an interrupt number to the **SWTRIG** register generates a Software Generated Interrupt (SGI). See Table 2-9 on page 79 for interrupt assignments.

When the MAINPEND bit in the **Configuration and Control (CFGCTRL)** register (see page 128) is set, unprivileged software can access the **SWTRIG** register.

Software Trigger Interrupt (SWTRIG)

Base 0xE000.E000 Offset 0xF00

Type WO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	INTID	WO	0x00	Interrupt ID

This field holds the interrupt ID of the required SGI. For example, a value of 0x3 generates an interrupt on IRQ3.

3.5 System Control Block (SCB) Register Descriptions

This section lists and describes the System Control Block (SCB) registers, in numerical order by address offset. The SCB registers can only be accessed from privileged mode.

All registers must be accessed with aligned word accesses except for the **FAULTSTAT** and **SYSPRI1-SYSPRI3** registers, which can be accessed with byte or aligned halfword or word accesses. The processor does not support unaligned accesses to system control block registers.

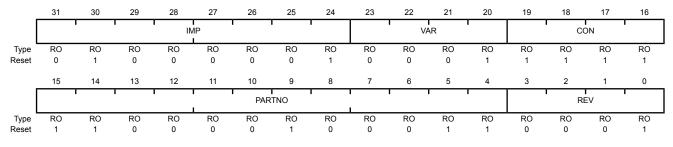
Register 26: CPU ID Base (CPUID), offset 0xD00

Note: This register can only be accessed from privileged mode.

The **CPUID** register contains the ARM® Cortex™-M3 processor part number, version, and implementation information.

CPU ID Base (CPUID)

Base 0xE000.E000 Offset 0xD00 Type RO, reset 0x411F.C231



Bit/Field	Name	Туре	Reset	Description
31:24	IMP	RO	0x41	Implementer Code
				Value Description 0x41 ARM
23:20	VAR	RO	0x1	Variant Number Value Description
				Ox1 The rn value in the rnpn product revision identifier, for example, the 1 in r1p1.
19:16	CON	RO	0xF	Constant
				Value Description 0xF Always reads as 0xF.
				OXF Always reads as OXF.
15:4	PARTNO	RO	0xC23	Part Number
				Value Description
				0xC23 Cortex-M3 processor.
3:0	REV	RO	0x1	Revision Number
				Value Description

The pn value in the rnpn product revision identifier, for example, the 1 in r1p1.

Register 27: Interrupt Control and State (INTCTRL), offset 0xD04

Note: This register can only be accessed from privileged mode.

The **INCTRL** register provides a set-pending bit for the NMI exception, and set-pending and clear-pending bits for the PendSV and SysTick exceptions. In addition, bits in this register indicate the exception number of the exception being processed, whether there are preempted active exceptions, the exception number of the highest priority pending exception, and whether any interrupts are pending.

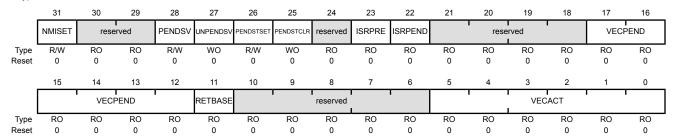
When writing to **INCTRL**, the effect is unpredictable when writing a 1 to both the PENDSV and UNPENDSV bits, or writing a 1 to both the PENDSTSET and PENDSTCLR bits.

Interrupt Control and State (INTCTRL)

Base 0xE000.E000 Offset 0xD04

28

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description	
31	NMISET	R/W	0	NMI Set Pendir	ıq

R/W

n

Value Description

- On a read, indicates an NMI exception is not pending. On a write, no effect.
- On a read, indicates an NMI exception is pending.
 On a write, changes the NMI exception state to pending.

Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it registers the setting of this bit, and clears this bit on entering the interrupt handler. A read of this bit by the NMI exception handler returns 1 only if the ${\tt NMI}$ signal is reasserted while the processor is executing that handler.

30:29	reserved	RO	0x0

PENDSV

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

PendSV Set Pending

Value Description

- On a read, indicates a PendSV exception is not pending.
 On a write, no effect.
- On a read, indicates a PendSV exception is pending.On a write, changes the PendSV exception state to pending.

Setting this bit is the only way to set the PendSV exception state to pending. This bit is cleared by writing a 1 to the ${\tt UNPENDSV}$ bit.

Bit/Field	Name	Туре	Reset	Description
27	UNPENDSV	WO	0	PendSV Clear Pending
				Value Description On a write, no effect. On a write, removes the pending state from the PendSV exception.
				This bit is write only; on a register read, its value is unknown.
26	PENDSTSET	R/W	0	SysTick Set Pending
				 Value Description On a read, indicates a SysTick exception is not pending. On a write, no effect. On a read, indicates a SysTick exception is pending. On a write, changes the SysTick exception state to pending. This bit is cleared by writing a 1 to the PENDSTCLR bit.
25	PENDSTCLR	WO	0	SysTick Clear Pending
				 Value Description On a write, no effect. On a write, removes the pending state from the SysTick exception.
				This bit is write only; on a register read, its value is unknown.
24	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
23	ISRPRE	RO	0	Debug Interrupt Handling
				Value Description The release from halt does not take an interrupt. The release from halt takes an interrupt. This bit is only meaningful in Debug mode and reads as zero when the processor is not in Debug mode.
22	ISRPEND	RO	0	Interrupt Pending
				Value Description 0 No interrupt is pending. 1 An interrupt is pending. This bit provides status for all interrupts excluding NMI and Faults.
21:18	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

July 15, 2014 121

Bit/Field	Name	Туре	Reset	Description
17:12	VECPEND	RO	0x00	Interrupt Pending Vector Number This field contains the exception number of the highest priority pending enabled exception. The value indicated by this field includes the effect of the BASEPRI and FAULTMASK registers, but not any effect of the PRIMASK register.
				Value Description
				0x00 No exceptions are pending
				0x01 Reserved
				0x02 NMI
				0x03 Hard fault
				0x04 Memory management fault
				0x05 Bus fault
				0x06 Usage fault
				0x07-0x0A Reserved
				0x0B SVCall
				0x0C Reserved for Debug
				0x0D Reserved
				0x0E PendSV
				0x0F SysTick
				0x10 Interrupt Vector 0
				· · · · · · · · · · · · · · · · · · ·
				'
				•
				0x3C-0x3F Reserved
11	RETBASE	RO	0	Return to Base
				Value Description
				O There are preempted active exceptions to execute.
				1 There are no active exceptions, or the currently executing exception is the only active exception.
				This bit provides status for all interrupts excluding NMI and Faults. This bit only has meaning if the processor is currently executing an ISR (the Interrupt Program Status (IPSR) register is non-zero).
10:6	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	VECACT	RO	0x00	Interrupt Pending Vector Number
				This field contains the active exception number. The exception numbers can be found in the description for the VECPEND field. If this field is clear, the processor is in Thread mode. This field contains the same value as the ISRNUM field in the IPSR register.
				Subtract 16 from this value to obtain the IRQ number required to index into the Interrupt Set Enable (ENn), Interrupt Clear Enable (DISn), Interrupt Set Pending (PENDn), Interrupt Clear Pending (UNPENDn), and Interrupt Priority (PRIn) registers (see page 60).

Register 28: Vector Table Offset (VTABLE), offset 0xD08

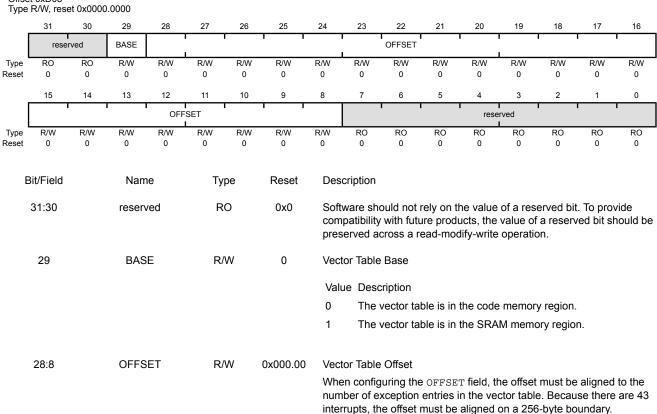
Note: This register can only be accessed from privileged mode.

The **VTABLE** register indicates the offset of the vector table base address from memory address 0x0000.0000.

Vector Table Offset (VTABLE)

Base 0xE000.E000 Offset 0xD08

7:0



RO

reserved

0x00

Software should not rely on the value of a reserved bit. To provide

preserved across a read-modify-write operation.

compatibility with future products, the value of a reserved bit should be

Register 29: Application Interrupt and Reset Control (APINT), offset 0xD0C

Note: This register can only be accessed from privileged mode.

The **APINT** register provides priority grouping control for the exception model, endian status for data accesses, and reset control of the system. To write to this register, 0x05FA must be written to the VECTKEY field, otherwise the write is ignored.

The PRIGROUP field indicates the position of the binary point that splits the INTx fields in the Interrupt Priority (PRIx) registers into separate group priority and subpriority fields. Table 3-8 on page 124 shows how the PRIGROUP value controls this split. The bit numbers in the Group Priority Field and Subpriority Field columns in the table refer to the bits in the INTA field. For the INTB field, the corresponding bits are 15:13; for INTC, 23:21; and for INTD, 31:29.

Note: Determining preemption of an exception uses only the group priority field.

Table 3-8. Interrupt Priority Levels

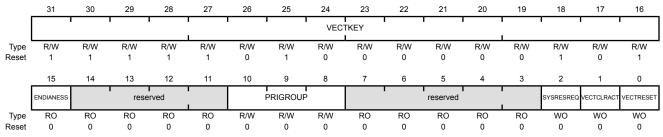
PRIGROUP Bit Field	Binary Point ^a	Group Priority Field		Group Priorities	Subpriorities
0x0 - 0x4	bxxx.	[7:5]	None	8	1
0x5	bxx.y	[7:6]	[5]	4	2
0x6	bx.yy	[7]	[6:5]	2	4
0x7	b.yyy	None	[7:5]	1	8

a. INTx field showing the binary point. An x denotes a group priority field bit, and a y denotes a subpriority field bit.

Application Interrupt and Reset Control (APINT)

Base 0xE000.E000 Offset 0xD0C

Type R/W, reset 0xFA05.0000



Bit/Field	Name	Type	Reset	Description
31:16	VECTKEY	R/W	0xFA05	Register Key
				This field is used to guard against accidental writes to this register. 0x05FA must be written to this field in order to change the bits in this register. On a read, 0xFA05 is returned.
15	ENDIANESS	RO	0	Data Endianess
				The Stellaris implementation uses only little-endian mode so this is cleared to 0.
14:11	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
10:8	PRIGROUP	R/W	0x0	Interrupt Priority Grouping This field determines the split of group priority from subpriority (see Table 3-8 on page 124 for more information).
7:3	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	SYSRESREQ	WO	0	System Reset Request
				Value Description
				0 No effect.
				1 Resets the core and all on-chip peripherals except the Debug interface.
				This bit is automatically cleared during the reset of the core and reads as 0.
1	VECTCLRACT	WO	0	Clear Active NMI / Fault
				This bit is reserved for Debug use and reads as 0. This bit must be written as a 0, otherwise behavior is unpredictable.
0	VECTRESET	WO	0	System Reset
				This bit is reserved for Debug use and reads as 0. This bit must be written as a 0, otherwise behavior is unpredictable.

Register 30: System Control (SYSCTRL), offset 0xD10

Note: This register can only be accessed from privileged mode.

The SYSCTRL register controls features of entry to and exit from low-power state.

System Control (SYSCTRL)

Base 0xE000.E000

Offset 0xD10
Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		•	1		 			rese	rved							1
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		'	'			reserved						SEVONPEND	reserved	SLEEPDEEP	SLEEPEXIT	reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	R/W	R/W	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Е	Bit/Field		Nam	ne	Ty	ре	Reset	Des	cription							

31:5	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SEVONPEND	R/W	0	Wake Up on Pending

Value Description

- Only enabled interrupts or events can wake up the processor; disabled interrupts are excluded.
- Enabled events and all interrupts, including disabled interrupts, can wake up the processor.

When an event or interrupt enters the pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE.

The processor also wakes up on execution of a SEV instruction or an external event.

3	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	SLEEPDEEP	R/W	0	Deep Sleep Enable

Value Description

- Use Sleep mode as the low power mode.
- Use Deep-sleep mode as the low power mode.

Bit/Field	Name	Туре	Reset	Description
1	SLEEPEXIT	R/W	0	Sleep on ISR Exit
				Value Description
				When returning from Handler mode to Thread mode, do not sleep when returning to Thread mode.
				When returning from Handler mode to Thread mode, enter sleep or deep sleep on return from an ISR.
				Setting this bit enables an interrupt-driven application to avoid returning to an empty main application.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 31: Configuration and Control (CFGCTRL), offset 0xD14

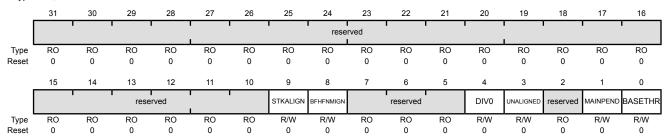
Note: This register can only be accessed from privileged mode.

The **CFGCTRL** register controls entry to Thread mode and enables: the handlers for NMI, hard fault and faults escalated by the **FAULTMASK** register to ignore bus faults; trapping of divide by zero and unaligned accesses; and access to the **SWTRIG** register by unprivileged software (see page 118).

Configuration and Control (CFGCTRL)

Base 0xE000.E000 Offset 0xD14

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:10	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9	STKALIGN	R/W	0	Stack Alignment on Exception Entry
				Value Description
				0 The stack is 4-byte aligned.
				1 The stack is 8-byte aligned.
				On exception entry, the processor uses bit 9 of the stacked PSR to indicate the stack alignment. On return from the exception, it uses this stacked bit to restore the correct stack alignment.
8	BFHFNMIGN	R/W	0	Ignore Bus Fault in NMI and Fault
				This bit enables handlers with priority -1 or -2 to ignore data bus faults caused by load and store instructions. The setting of this bit applies to the hard fault, NMI, and FAULTMASK escalated handlers.
				Value Description
				O Data bus faults caused by load and store instructions cause a lock-up.
				1 Handlers running at priority -1 and -2 ignore data bus faults caused by load and store instructions.
				Set this bit only when the handler and its data are in absolutely safe memory. The normal use of this bit is to probe system devices and bridges to detect control path problems and fix them.
7:5	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
4	DIVO	R/W	0	Trap on Divide by 0 This bit enables faulting or halting when the processor executes an SDIV or UDIV instruction with a divisor of 0. Value Description 0 Do not trap on divide by 0. A divide by zero returns a quotient of 0. 1 Trap on divide by 0.
3	UNALIGNED	R/W	0	Trap on Unaligned Access Value Description 0 Do not trap on unaligned halfword and word accesses. 1 Trap on unaligned halfword and word accesses. An unaligned access generates a usage fault. Unaligned LDM, STM, LDRD, and STRD instructions always fault regardless of whether UNALIGNED is set.
2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	MAINPEND	R/W	0	Allow Main Interrupt Trigger Value Description Disables unprivileged software access to the SWTRIG register. Enables unprivileged software access to the SWTRIG register (see page 118).
0	BASETHR	R/W	0	Thread State Control Value Description 0 The processor can enter Thread mode only when no exception is active. 1 The processor can enter Thread mode from any level under the control of an EXC_RETURN value (see "Exception Return" on page 84 for more information).

Register 32: System Handler Priority 1 (SYSPRI1), offset 0xD18

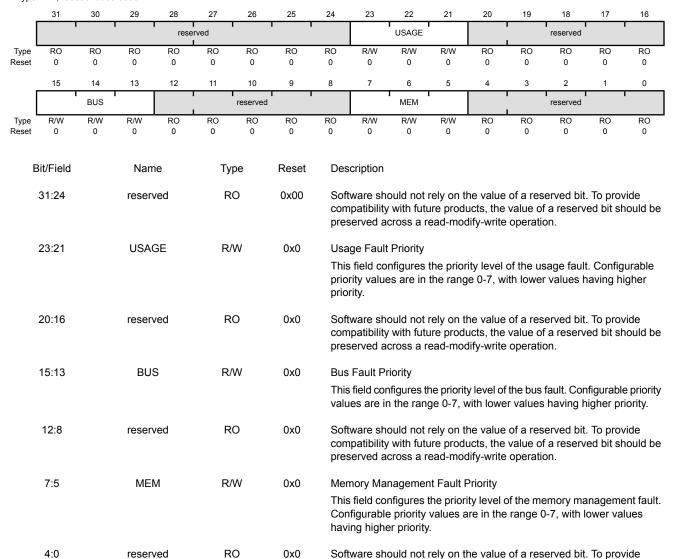
Note: This register can only be accessed from privileged mode.

The **SYSPRI1** register configures the priority level, 0 to 7 of the usage fault, bus fault, and memory management fault exception handlers. This register is byte-accessible.

System Handler Priority 1 (SYSPRI1)

Base 0xE000.E000 Offset 0xD18

Type R/W, reset 0x0000.0000



compatibility with future products, the value of a reserved bit should be

preserved across a read-modify-write operation.

Register 33: System Handler Priority 2 (SYSPRI2), offset 0xD1C

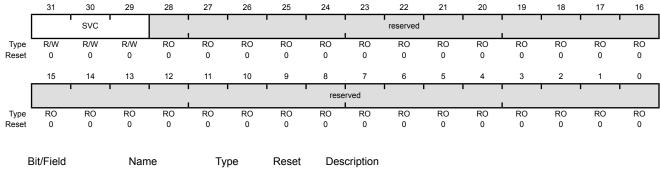
Note: This register can only be accessed from privileged mode.

The SYSPRI2 register configures the priority level, 0 to 7 of the SVCall handler. This register is byte-accessible.

System Handler Priority 2 (SYSPRI2)

Base 0xE000.E000

Offset 0xD1C Type R/W, reset 0x0000.0000



Bit/Field	name	туре	Reset	Description
31:29	SVC	R/W	0x0	SVCall Priority
				This field configures the priority level of SVCall. Configurable priority values are in the range 0-7, with lower values having higher priority.

28:0 RO 0x000.0000 Software should not rely on the value of a reserved bit. To provide reserved

compatibility with future products, the value of a reserved bit should be

preserved across a read-modify-write operation.

Register 34: System Handler Priority 3 (SYSPRI3), offset 0xD20

Note: This register can only be accessed from privileged mode.

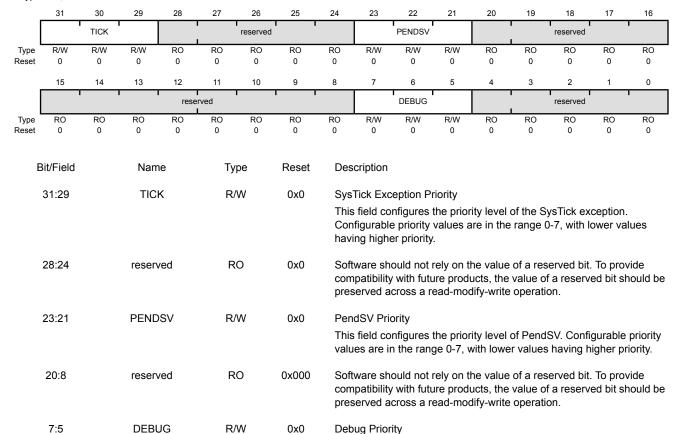
The **SYSPRI3** register configures the priority level, 0 to 7 of the SysTick exception and PendSV handlers. This register is byte-accessible.

System Handler Priority 3 (SYSPRI3)

Base 0xE000.E000 Offset 0xD20

4:0

Type R/W, reset 0x0000.0000



RO

reserved

0x0.0000

This field configures the priority level of Debug. Configurable priority values are in the range 0-7, with lower values having higher priority.

Software should not rely on the value of a reserved bit. To provide

preserved across a read-modify-write operation.

compatibility with future products, the value of a reserved bit should be

be

Register 35: System Handler Control and State (SYSHNDCTRL), offset 0xD24

Note: This register can only be accessed from privileged mode.

The **SYSHNDCTRL** register enables the system handlers, and indicates the pending status of the usage fault, bus fault, memory management fault, and SVC exceptions as well as the active status of the system handlers.

If a system handler is disabled and the corresponding fault occurs, the processor treats the fault as a hard fault.

This register can be modified to change the pending or active status of system exceptions. An OS kernel can write to the active bits to perform a context switch that changes the current exception type.

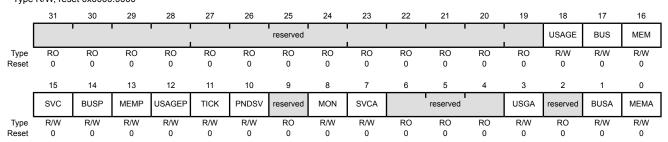
Caution – Software that changes the value of an active bit in this register without correct adjustment to the stacked content can cause the processor to generate a fault exception. Ensure software that writes to this register retains and subsequently restores the current active status.

If the value of a bit in this register must be modified after enabling the system handlers, a read-modify-write procedure must be used to ensure that only the required bit is modified.

System Handler Control and State (SYSHNDCTRL)

Base 0xE000.E000 Offset 0xD24

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:19	reserved	RO	0x000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
18	USAGE	R/W	0	Usage Fault Enable
				Value Description
				0 Disables the usage fault exception.
				1 Enables the usage fault exception.
17	BUS	R/W	0	Bus Fault Enable
				Value Description
				0 Disables the bus fault exception.
	31:19	31:19 reserved 18 USAGE	31:19 reserved RO 18 USAGE R/W	31:19 reserved RO 0x000 18 USAGE R/W 0

Enables the bus fault exception.

Bit/Field	Name	Туре	Reset	Description
16	MEM	R/W	0	Memory Management Fault Enable
				Value Description Disables the memory management fault exception. Enables the memory management fault exception.
15	SVC	R/W	0	SVC Call Pending Value Description O An SVC call exception is not pending.
				 An SVC call exception is not pending. An SVC call exception is pending. This bit can be modified to change the pending status of the SVC call exception.
14	BUSP	R/W	0	Bus Fault Pending
				Value Description O A bus fault exception is not pending. A bus fault exception is pending.
				This bit can be modified to change the pending status of the bus fault exception.
13	MEMP	R/W	0	Memory Management Fault Pending
				Value Description O A memory management fault exception is not pending. A memory management fault exception is pending.
				This bit can be modified to change the pending status of the memory management fault exception.
12	USAGEP	R/W	0	Usage Fault Pending
				 Value Description A usage fault exception is not pending. A usage fault exception is pending.
				This bit can be modified to change the pending status of the usage fault exception.
11	TICK	R/W	0	SysTick Exception Active
				Value Description O A SysTick exception is not active. A SysTick exception is active.
				This bit can be modified to change the active status of the SysTick exception, however, see the Caution above before setting this bit.

Bit/Field	Name	Туре	Reset	Description
10	PNDSV	R/W	0	PendSV Exception Active
				Value Description
				0 A PendSV exception is not active.
				1 A PendSV exception is active.
				This bit can be modified to change the active status of the PendSV exception, however, see the Caution above before setting this bit.
9	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
8	MON	R/W	0	Debug Monitor Active
				Value Description
				0 The Debug monitor is not active.
				1 The Debug monitor is active.
7	SVCA	R/W	0	SVC Call Active
				Value Description
				0 SVC call is not active.
				1 SVC call is active.
				This bit can be modified to change the active status of the SVC call exception, however, see the Caution above before setting this bit.
6:4	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	USGA	R/W	0	Usage Fault Active
				Value Description
				0 Usage fault is not active.
				1 Usage fault is active.
				This bit can be modified to change the active status of the usage fault exception, however, see the Caution above before setting this bit.
2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	BUSA	R/W	0	Bus Fault Active
				Value Description
				0 Bus fault is not active.
				1 Bus fault is active.
				This bit can be modified to change the active status of the bus fault exception, however, see the Caution above before setting this bit.

July 15, 2014 135

Bit/Field	Name	Туре	Reset	Description
0	MEMA	R/W	0	Memory Management Fault Active
				Value Description 0 Memory management fault is not active. 1 Memory management fault is active. This bit can be modified to change the active status of the memory management fault exception, however, see the Caution above before setting this bit.

Register 36: Configurable Fault Status (FAULTSTAT), offset 0xD28

Note: This register can only be accessed from privileged mode.

The **FAULTSTAT** register indicates the cause of a memory management fault, bus fault, or usage fault. Each of these functions is assigned to a subregister as follows:

- Usage Fault Status (UFAULTSTAT), bits 31:16
- Bus Fault Status (BFAULTSTAT), bits 15:8
- Memory Management Fault Status (MFAULTSTAT), bits 7:0

FAULTSTAT is byte accessible. **FAULTSTAT** or its subregisters can be accessed as follows:

- The complete **FAULTSTAT** register, with a word access to offset 0xD28
- The **MFAULTSTAT**, with a byte access to offset 0xD28
- The MFAULTSTAT and BFAULTSTAT, with a halfword access to offset 0xD28
- The **BFAULTSTAT**, with a byte access to offset 0xD29
- The **UFAULTSTAT**, with a halfword access to offset 0xD2A

Bits are cleared by writing a 1 to them.

In a fault handler, the true faulting address can be determined by:

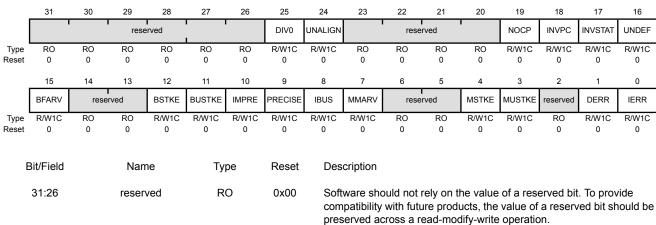
- Read and save the Memory Management Fault Address (MMADDR) or Bus Fault Address (FAULTADDR) value.
- 2. Read the MMARV bit in **MFAULTSTAT**, or the BFARV bit in **BFAULTSTAT** to determine if the **MMADDR** or **FAULTADDR** contents are valid.

Software must follow this sequence because another higher priority exception might change the **MMADDR** or **FAULTADDR** value. For example, if a higher priority handler preempts the current fault handler, the other fault might change the **MMADDR** or **FAULTADDR** value.

Configurable Fault Status (FAULTSTAT)

Base 0xE000.E000 Offset 0xD28

Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
25	DIV0	R/W1C	0	Divide-by-Zero Usage Fault
				Value Description
				No divide-by-zero fault has occurred, or divide-by-zero trapping is not enabled.
				1 The processor has executed an SDIV or UDIV instruction with a divisor of 0.
				When this bit is set, the PC value stacked for the exception return points to the instruction that performed the divide by zero.
				Trapping on divide-by-zero is enabled by setting the DIVO bit in the Configuration and Control (CFGCTRL) register (see page 128).
				This bit is cleared by writing a 1 to it.
24	UNALIGN	R/W1C	0	Unaligned Access Usage Fault
				Value Description
				No unaligned access fault has occurred, or unaligned access trapping is not enabled.
				1 The processor has made an unaligned memory access.
				Unaligned LDM, STM, LDRD, and STRD instructions always fault regardless of the configuration of this bit.
				Trapping on unaligned access is enabled by setting the UNALIGNED bit in the CFGCTRL register (see page 128).
				This bit is cleared by writing a 1 to it.
23:20	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
19	NOCP	R/W1C	0	No Coprocessor Usage Fault
				Value Description
				O A usage fault has not been caused by attempting to access a coprocessor.
				1 The processor has attempted to access a coprocessor.
				This bit is cleared by writing a 1 to it.
18	INVPC	R/W1C	0	Invalid PC Load Usage Fault
				Value Description
				O A usage fault has not been caused by attempting to load an invalid PC value.
				The processor has attempted an illegal load of EXC_RETURN to the PC as a result of an invalid context or an invalid EXC_RETURN value.
				When this bit is set, the PC value stacked for the exception return points to the instruction that tried to perform the illegal load of the PC .
				This bit is cleared by writing a 1 to it.

Bit/Field	Name	Туре	Reset	Description
17	INVSTAT	R/W1C	0	Invalid State Usage Fault
				Value Description
				O A usage fault has not been caused by an invalid state.
				1 The processor has attempted to execute an instruction that makes illegal use of the EPSR register.
				When this bit is set, the PC value stacked for the exception return points to the instruction that attempted the illegal use of the Execution Program Status Register (EPSR) register.
				This bit is not set if an undefined instruction uses the EPSR register.
				This bit is cleared by writing a 1 to it.
16	UNDEF	R/W1C	0	Undefined Instruction Usage Fault
				Value Description
				0 A usage fault has not been caused by an undefined instruction.
				1 The processor has attempted to execute an undefined instruction.
				When this bit is set, the PC value stacked for the exception return points to the undefined instruction.
				An undefined instruction is an instruction that the processor cannot decode.
				This bit is cleared by writing a 1 to it.
15	BFARV	R/W1C	0	Bus Fault Address Register Valid
				Value Description
				The value in the Bus Fault Address (FAULTADDR) register is not a valid fault address.
				1 The FAULTADDR register is holding a valid fault address.
				This bit is set after a bus fault, where the address is known. Other faults can clear this bit, such as a memory management fault occurring later.
				If a bus fault occurs and is escalated to a hard fault because of priority, the hard fault handler must clear this bit. This action prevents problems if returning to a stacked active bus fault handler whose FAULTADDR register value has been overwritten.
				This bit is cleared by writing a 1 to it.
14:13	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
12	BSTKE	R/W1C	0	Stack Bus Fault
				Value Description
				0 No bus fault has occurred on stacking for exception entry.
				Stacking for an exception entry has caused one or more bus faults.
				When this bit is set, the SP is still adjusted but the values in the context area on the stack might be incorrect. A fault address is not written to the FAULTADDR register.
				This bit is cleared by writing a 1 to it.
11	BUSTKE	R/W1C	0	Unstack Bus Fault
				Value Description
				No bus fault has occurred on unstacking for a return from exception.
				1 Unstacking for a return from exception has caused one or more bus faults.
				This fault is chained to the handler. Thus, when this bit is set, the original return stack is still present. The SP is not adjusted from the failing return, a new save is not performed, and a fault address is not written to the FAULTADDR register.
				This bit is cleared by writing a 1 to it.
10	IMPRE	R/W1C	0	Imprecise Data Bus Error
				Value Description
				O An imprecise data bus error has not occurred.
				A data bus error has occurred, but the return address in the stack frame is not related to the instruction that caused the error.
				When this bit is set, a fault address is not written to the FAULTADDR register.
				This fault is asynchronous. Therefore, if the fault is detected when the priority of the current process is higher than the bus fault priority, the bus fault becomes pending and becomes active only when the processor returns from all higher-priority processes. If a precise fault occurs before the processor enters the handler for the imprecise bus fault, the handler detects that both the IMPRE bit is set and one of the precise fault status bits is set.
				This bit is cleared by writing a 1 to it.
9	PRECISE	R/W1C	0	Precise Data Bus Error
				Value Description
				O A precise data bus error has not occurred.
				A data bus error has occurred, and the PC value stacked for the exception return points to the instruction that caused the fault.
				When this bit is set, the fault address is written to the FAULTADDR register.
				This bit is alsowed by continue a 4 to it

July 15, 2014

This bit is cleared by writing a 1 to it.

Bit/Field	Name	Туре	Reset	Description				
8	IBUS	R/W1C	0	Instruction Bus Error				
				Value Description				
				O An instruction bus error has not occurred.				
				1 An instruction bus error has occurred.				
				The processor detects the instruction bus error on prefetching an instruction, but sets this bit only if it attempts to issue the faulting instruction.				
				When this bit is set, a fault address is not written to the FAULTADDR register.				
				This bit is cleared by writing a 1 to it.				
7	MMARV	R/W1C	0	Memory Management Fault Address Register Valid				
				Value Description				
				0 The value in the Memory Management Fault Address (MMADDR) register is not a valid fault address.				
				1 The MMADDR register is holding a valid fault address.				
				If a memory management fault occurs and is escalated to a hard fault because of priority, the hard fault handler must clear this bit. This action prevents problems if returning to a stacked active memory management fault handler whose MMADDR register value has been overwritten.				
				This bit is cleared by writing a 1 to it.				
6:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.				
4	MSTKE	R/W1C	0	Stack Access Violation				
				Value Description				
				No memory management fault has occurred on stacking for exception entry.				
				Stacking for an exception entry has caused one or more access violations.				
				When this bit is set, the SP is still adjusted but the values in the context area on the stack might be incorrect. A fault address is not written to the MMADDR register.				

This bit is cleared by writing a 1 to it.

Bit/Field	Name	Туре	Reset	Description
3	MUSTKE	R/W1C	0	Unstack Access Violation
				Value Description
				No memory management fault has occurred on unstacking for a return from exception.
				1 Unstacking for a return from exception has caused one or more access violations.
				This fault is chained to the handler. Thus, when this bit is set, the original return stack is still present. The SP is not adjusted from the failing return, a new save is not performed, and a fault address is not written to the MMADDR register.
				This bit is cleared by writing a 1 to it.
2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	DERR	R/W1C	0	Data Access Violation
				Value Description
				0 A data access violation has not occurred.
				1 The processor attempted a load or store at a location that does not permit the operation.
				When this bit is set, the PC value stacked for the exception return points to the faulting instruction and the address of the attempted access is written to the MMADDR register.
				This bit is cleared by writing a 1 to it.
0	IERR	R/W1C	0	Instruction Access Violation
				Value Description
				0 An instruction access violation has not occurred.
				1 The processor attempted an instruction fetch from a location that does not permit execution.
				This fault occurs on any access to an XN region, even when the MPU is disabled or not present.
				When this bit is set, the PC value stacked for the exception return points to the faulting instruction and the address of the attempted access is not written to the MMADDR register.

142 July 15, 2014

This bit is cleared by writing a 1 to it.

Register 37: Hard Fault Status (HFAULTSTAT), offset 0xD2C

Note: This register can only be accessed from privileged mode.

25

24

The **HFAULTSTAT** register gives information about events that activate the hard fault handler.

23

22

21

20

Software should not rely on the value of a reserved bit. To provide

preserved across a read-modify-write operation.

compatibility with future products, the value of a reserved bit should be

19

18

17

16

Bits are cleared by writing a 1 to them.

27

26

Hard Fault Status (HFAULTSTAT)

Base 0xE000.E000

31

0

Offset 0xD2C Type R/W1C, reset 0x0000.0000

30

29

28

	DBG	FORCED					1 1		rese	rved		i i	ľ		Í	Î
Type L	R/W1C	R/W1C	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Г	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					1		reser		1						VECT	reserved
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	R/W1C 0	RO 0
P	sit/Field		Nam	ne	Ту	ne	Reset	Des	cription							
_					. ,	,	110001		·							
31 DBG		G	R/W	/1C	0		ug Even									
							This bit is reserved for Debug use. This bit must be written as a otherwise behavior is unpredictable.							a 0,		
30 FORCED				R/W	/1C	0	Ford	ed Hard	Fault							
								Valı	ue Desc	ription						
								0	No fo	rced ha	rd fault h	as occur	red.			
								1	with o	configura	ble prior	s been g ity that ca it is disal	nnot be	•		
												fault han		st read t	he other	fault
									bit is cle							
					_	_				_	_					
	29:2		reser	/ed	R	O	0x00	com	patibility	with futu	ure produ	he value ucts, the dify-write	value of	a reser		
	1		VEC	т	R/W	/1C	0	Vec	tor Table	Read F	ault					
								Valı	ue Desc	ription						
								0	No b	us fault h	nas occu	rred on a	vector	table re	ad.	
								1	A bus	s fault o	curred o	on a vecto	or table i	read.		
								This	error is	always h	nandled	by the ha	rd fault l	handler.		
								Whe	en this bit	is set, th	ne PC va	lue stack empted	ed for th	е ехсер	tion retu	n points

RO

reserved

0

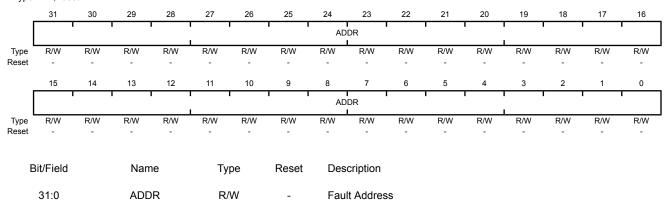
Register 38: Memory Management Fault Address (MMADDR), offset 0xD34

Note: This register can only be accessed from privileged mode.

The MMADDR register contains the address of the location that generated a memory management fault. When an unaligned access faults, the address in the MMADDR register is the actual address that faulted. Because a single read or write instruction can be split into multiple aligned accesses, the fault address can be any address in the range of the requested access size. Bits in the Memory Management Fault Status (MFAULTSTAT) register indicate the cause of the fault and whether the value in the MMADDR register is valid (see page 137).

Memory Management Fault Address (MMADDR)

Base 0xE000.E000 Offset 0xD34 Type R/W, reset -



When the MMARV bit of **MFAULTSTAT** is set, this field holds the address of the location that generated the memory management fault.

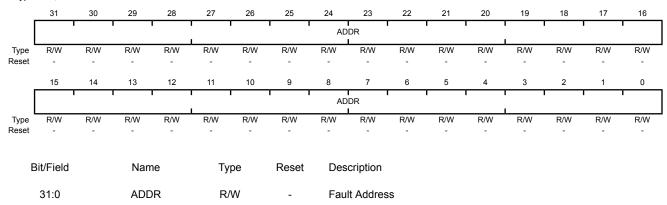
Register 39: Bus Fault Address (FAULTADDR), offset 0xD38

Note: This register can only be accessed from privileged mode.

The **FAULTADDR** register contains the address of the location that generated a bus fault. When an unaligned access faults, the address in the **FAULTADDR** register is the one requested by the instruction, even if it is not the address of the fault. Bits in the **Bus Fault Status (BFAULTSTAT)** register indicate the cause of the fault and whether the value in the **FAULTADDR** register is valid (see page 137).



Base 0xE000.E000 Offset 0xD38 Type R/W, reset -



When the FAULTADDRV bit of **BFAULTSTAT** is set, this field holds the address of the location that generated the bus fault.

3.6 Memory Protection Unit (MPU) Register Descriptions

This section lists and describes the Memory Protection Unit (MPU) registers, in numerical order by address offset.

The MPU registers can only be accessed from privileged mode.

Register 40: MPU Type (MPUTYPE), offset 0xD90

Note: This register can only be accessed from privileged mode.

The **MPUTYPE** register indicates whether the MPU is present, and if so, how many regions it supports.

MPU Type (MPUTYPE)

Base 0xE000.E000 Offset 0xD90

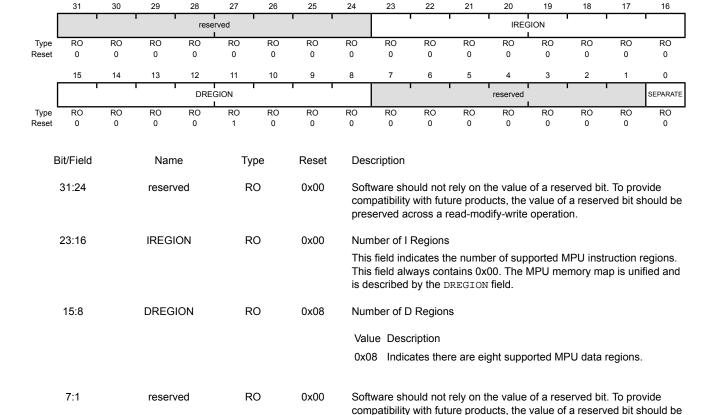
Type RO, reset 0x0000.0800

SEPARATE

0

RO

0



Value Description

Separate or Unified MPU

Indicates the MPU is unified.

preserved across a read-modify-write operation.

Register 41: MPU Control (MPUCTRL), offset 0xD94

Note: This register can only be accessed from privileged mode.

The **MPUCTRL** register enables the MPU, enables the default memory map background region, and enables use of the MPU when in the hard fault, Non-maskable Interrupt (NMI), and **Fault Mask Register (FAULTMASK)** escalated handlers.

When the ENABLE and PRIVDEFEN bits are both set:

- For privileged accesses, the default memory map is as described in "Memory Model" on page 68. Any access by privileged software that does not address an enabled memory region behaves as defined by the default memory map.
- Any access by unprivileged software that does not address an enabled memory region causes a memory management fault.

Execute Never (XN) and Strongly Ordered rules always apply to the System Control Space regardless of the value of the ENABLE bit.

When the ENABLE bit is set, at least one region of the memory map must be enabled for the system to function unless the PRIVDEFEN bit is set. If the PRIVDEFEN bit is set and no regions are enabled, then only privileged software can operate.

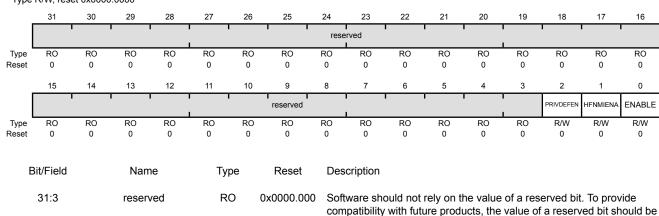
When the ENABLE bit is clear, the system uses the default memory map, which has the same memory attributes as if the MPU is not implemented (see Table 2-5 on page 70 for more information). The default memory map applies to accesses from both privileged and unprivileged software.

When the MPU is enabled, accesses to the System Control Space and vector table are always permitted. Other areas are accessible based on regions and whether PRIVDEFEN is set.

Unless HFNMIENA is set, the MPU is not enabled when the processor is executing the handler for an exception with priority -1 or -2. These priorities are only possible when handling a hard fault or NMI exception or when **FAULTMASK** is enabled. Setting the HFNMIENA bit enables the MPU when operating with these two priorities.

MPU Control (MPUCTRL)

Base 0xE000.E000 Offset 0xD94 Type R/W, reset 0x0000.0000



preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
2	PRIVDEFEN	R/W	0	MPU Default Region
				This bit enables privileged software access to the default memory map.
				Value Description
				0 If the MPU is enabled, this bit disables use of the default memory map. Any memory access to a location not covered by any enabled region causes a fault.
				1 If the MPU is enabled, this bit enables use of the default memory map as a background region for privileged software accesses.
				When this bit is set, the background region acts as if it is region number -1. Any region that is defined and enabled has priority over this default map.
				If the MPU is disabled, the processor ignores this bit.
1	HFNMIENA	R/W	0	MPU Enabled During Faults
				This bit controls the operation of the MPU during hard fault, NMI, and FAULTMASK handlers.
				Value Description
				The MPU is disabled during hard fault, NMI, and FAULTMASK handlers, regardless of the value of the ENABLE bit.
				1 The MPU is enabled during hard fault, NMI, and FAULTMASK handlers.
				When the MPU is disabled and this bit is set, the resulting behavior is unpredictable.
0	ENABLE	R/W	0	MPU Enable
				Value Description
				0 The MPU is disabled.
				1 The MPU is enabled.
				When the MPU is disabled and the ${\tt HFNMIENA}$ bit is set, the resulting behavior is unpredictable.

Register 42: MPU Region Number (MPUNUMBER), offset 0xD98

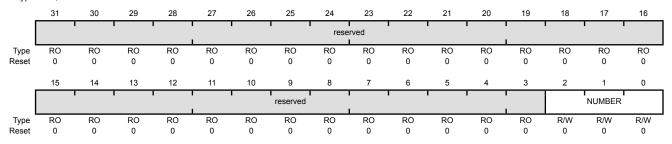
Note: This register can only be accessed from privileged mode.

The MPUNUMBER register selects which memory region is referenced by the MPU Region Base Address (MPUBASE) and MPU Region Attribute and Size (MPUATTR) registers. Normally, the required region number should be written to this register before accessing the MPUBASE or the MPUATTR register. However, the region number can be changed by writing to the MPUBASE register with the VALID bit set (see page 150). This write updates the value of the REGION field.

MPU Region Number (MPUNUMBER)

Base 0xE000.E000 Offset 0xD98

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:3	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2:0	NUMBER	R/W	0x0	MPU Region to Access

This field indicates the MPU region referenced by the **MPUBASE** and **MPUATTR** registers. The MPU supports eight memory regions.

Register 43: MPU Region Base Address (MPUBASE), offset 0xD9C

Register 44: MPU Region Base Address Alias 1 (MPUBASE1), offset 0xDA4

Register 45: MPU Region Base Address Alias 2 (MPUBASE2), offset 0xDAC

Register 46: MPU Region Base Address Alias 3 (MPUBASE3), offset 0xDB4

Note: This register can only be accessed from privileged mode.

The MPUBASE register defines the base address of the MPU region selected by the MPU Region Number (MPUNUMBER) register and can update the value of the MPUNUMBER register. To change the current region number and update the MPUNUMBER register, write the MPUBASE register with the VALID bit set.

The ADDR field is bits 31:*N* of the **MPUBASE** register. Bits (*N*-1):5 are reserved. The region size, as specified by the SIZE field in the **MPU Region Attribute and Size (MPUATTR)** register, defines the value of *N* where:

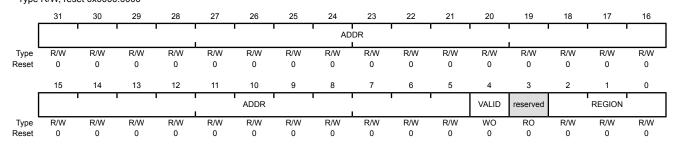
 $N = Log_2$ (Region size in bytes)

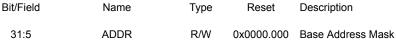
If the region size is configured to 4 GB in the **MPUATTR** register, there is no valid ADDR field. In this case, the region occupies the complete memory map, and the base address is 0x0000.0000.

The base address is aligned to the size of the region. For example, a 64-KB region must be aligned on a multiple of 64 KB, for example, at 0x0001.0000 or 0x0002.0000.

MPU Region Base Address (MPUBASE)

Base 0xE000.E000 Offset 0xD9C Type R/W, reset 0x0000.0000





Bits 31:N in this field contain the region base address. The value of N depends on the region size, as shown above. The remaining bits (N-1):5 are reserved.

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
4	VALID	WO	0	Region Number Valid
				Value Description
				The MPUNUMBER register is not changed and the processor updates the base address for the region specified in the MPUNUMBER register and ignores the value of the REGION field.
				The MPUNUMBER register is updated with the value of the REGION field and the base address is updated for the region specified in the REGION field.
				This bit is always read as 0.
3	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2:0	REGION	R/W	0x0	Region Number On a write, contains the value to be written to the MPUNUMBER register. On a read, returns the current region number in the MPUNUMBER register.

Register 47: MPU Region Attribute and Size (MPUATTR), offset 0xDA0

Register 48: MPU Region Attribute and Size Alias 1 (MPUATTR1), offset 0xDA8

Register 49: MPU Region Attribute and Size Alias 2 (MPUATTR2), offset 0xDB0

Register 50: MPU Region Attribute and Size Alias 3 (MPUATTR3), offset 0xDB8

Note: This register can only be accessed from privileged mode.

The **MPUATTR** register defines the region size and memory attributes of the MPU region specified by the **MPU Region Number (MPUNUMBER)** register and enables that region and any subregions.

The **MPUATTR** register is accessible using word or halfword accesses with the most-significant halfword holding the region attributes and the least-significant halfword holds the region size and the region and subregion enable bits.

The MPU access permission attribute bits, XN, AP, TEX, S, C, and B, control access to the corresponding memory region. If an access is made to an area of memory without the required permissions, then the MPU generates a permission fault.

The SIZE field defines the size of the MPU memory region specified by the **MPUNUMBER** register as follows:

(Region size in bytes) = $2^{(SIZE+1)}$

The smallest permitted region size is 32 bytes, corresponding to a SIZE value of 4. Table 3-9 on page 152 gives example SIZE values with the corresponding region size and value of N in the MPU Region Base Address (MPUBASE) register.

Table 3-9. Example SIZE Field Values

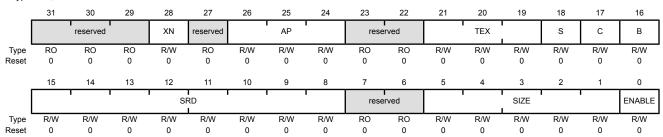
SIZE Encoding	Region Size	Value of N ^a	Note
00100b (0x4)	32 B	5	Minimum permitted size
01001b (0x9)	1 KB	10	-
10011b (0x13)	1 MB	20	-
11101b (0x1D)	1 GB	30	-
11111b (0x1F)		No valid ADDR field in MPUBASE ; the region occupies the complete memory map.	Maximum possible size

a. Refers to the N parameter in the MPUBASE register (see page 150).

MPU Region Attribute and Size (MPUATTR)

Base 0xE000.E000 Offset 0xDA0

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:29	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	XN	R/W	0	Instruction Access Disable
				Value Description
				0 Instruction fetches are enabled.
				1 Instruction fetches are disabled.
27	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
26:24	AP	R/W	0	Access Privilege
				For information on using this bit field, see Table 3-5 on page 98.
23:22	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
21:19	TEX	R/W	0x0	Type Extension Mask
				For information on using this bit field, see Table 3-3 on page 97.
18	S	R/W	0	Shareable For information on using this bit, see Table 3-3 on page 97.
17	С	R/W	0	Cacheable For information on using this bit, see Table 3-3 on page 97.
16	В	R/W	0	Bufferable
				For information on using this bit, see Table 3-3 on page 97.
15:8	SRD	R/W	0x00	Subregion Disable Bits
				Value Description
				The corresponding subregion is enabled.
				1 The corresponding subregion is disabled.
				Region sizes of 128 bytes and less do not support subregions. When writing the attributes for such a region, configure the SRD field as 0x00. See the section called "Subregions" on page 96 for more information.
7:6	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:1	SIZE	R/W	0x0	Region Size Mask
				The SIZE field defines the size of the MPU memory region specified by the MPUNUMBER register. Refer to Table 3-9 on page 152 for more information.

Bit/Field	Name	Туре	Reset	Description
0	ENABLE	R/W	0	Region Enable
				Value Description
				0 The region is disabled.
				1 The region is enabled.

4 JTAG Interface

The Joint Test Action Group (JTAG) port is an IEEE standard that defines a Test Access Port and Boundary Scan Architecture for digital integrated circuits and provides a standardized serial interface for controlling the associated test logic. The TAP, Instruction Register (IR), and Data Registers (DR) can be used to test the interconnections of assembled printed circuit boards and obtain manufacturing information on the components. The JTAG Port also provides a means of accessing and controlling design-for-test features such as I/O pin observation and control, scan testing, and debugging.

The JTAG port is comprised of five pins: TRST, TCK, TMS, TDI, and TDO. Data is transmitted serially into the controller on TDI and out of the controller on TDO. The interpretation of this data is dependent on the current state of the TAP controller. For detailed information on the operation of the JTAG port and TAP controller, please refer to the *IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture*.

The Stellaris JTAG controller works with the ARM JTAG controller built into the Cortex-M3 core. This is implemented by multiplexing the ${\tt TDO}$ outputs from both JTAG controllers. ARM JTAG instructions select the ARM ${\tt TDO}$ output while Stellaris JTAG instructions select the Stellaris ${\tt TDO}$ outputs. The multiplexer is controlled by the Stellaris JTAG controller, which has comprehensive programming for the ARM, Stellaris, and unimplemented JTAG instructions.

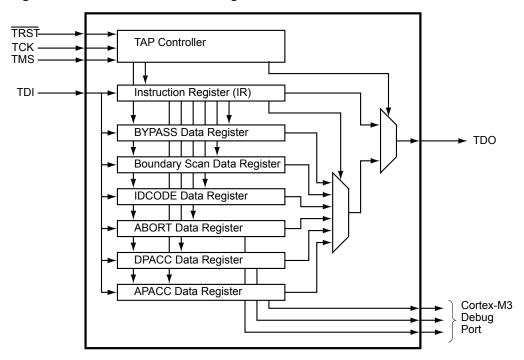
The Stellaris JTAG module has the following features:

- IEEE 1149.1-1990 compatible Test Access Port (TAP) controller
- Four-bit Instruction Register (IR) chain for storing JTAG instructions
- IEEE standard instructions: BYPASS, IDCODE, SAMPLE/PRELOAD, EXTEST and INTEST
- ARM additional instructions: APACC, DPACC and ABORT
- Integrated ARM Serial Wire Debug (SWD)

See the ARM® Debug Interface V5 Architecture Specification for more information on the ARM JTAG controller.

4.1 Block Diagram

Figure 4-1. JTAG Module Block Diagram



4.2 Signal Description

Table 4-1 on page 156 and Table 4-2 on page 157 list the external signals of the JTAG/SWD controller and describe the function of each. The JTAG/SWD controller signals are alternate functions for some GPIO signals, however note that the reset state of the pins is for the JTAG/SWD function. The JTAG/SWD controller signals are under commit protection and require a special process to be configured as GPIOs, see "Commit Control" on page 266. The column in the table below titled "Pin Assignment" lists the GPIO pin placement for the JTAG/SWD controller signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) is set to choose the JTAG/SWD function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 4-1. JTAG_SWD_SWO Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
SWCLK	80	1	TTL	JTAG/SWD CLK.
SWDIO	79	I/O	TTL	JTAG TMS and SWDIO.
SWO	77	0	TTL	JTAG TDO and SWO.
TCK	80	1	TTL	JTAG/SWD CLK.
TDI	78	1	TTL	JTAG TDI.
TDO	77	0	TTL	JTAG TDO and SWO.
TMS	79	I/O	TTL	JTAG TMS and SWDIO.
TRST	89	1	TTL	JTAG TRST.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 4-2. JTAG_SWD_SWO Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
SWCLK	A9	1	TTL	JTAG/SWD CLK.
SWDIO	В9	I/O	TTL	JTAG TMS and SWDIO.
SWO	A10	0	TTL	JTAG TDO and SWO.
TCK	A9	1	TTL	JTAG/SWD CLK.
TDI	В8	I	TTL	JTAG TDI.
TDO	A10	0	TTL	JTAG TDO and SWO.
TMS	В9	I/O	TTL	JTAG TMS and SWDIO.
TRST	A8	1	TTL	JTAG TRST.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

4.3 Functional Description

A high-level conceptual drawing of the JTAG module is shown in Figure 4-1 on page 156. The JTAG module is composed of the Test Access Port (TAP) controller and serial shift chains with parallel update registers. The TAP controller is a simple state machine controlled by the TRST, TCK and TMS inputs. The current state of the TAP controller depends on the current value of TRST and the sequence of values captured on TMS at the rising edge of TCK. The TAP controller determines when the serial shift chains capture new data, shift data from TDI towards TDO, and update the parallel load registers. The current state of the TAP controller also determines whether the Instruction Register (IR) chain or one of the Data Register (DR) chains is being accessed.

The serial shift chains with parallel load registers are comprised of a single Instruction Register (IR) chain and multiple Data Register (DR) chains. The current instruction loaded in the parallel load register determines which DR chain is captured, shifted, or updated during the sequencing of the TAP controller.

Some instructions, like EXTEST and INTEST, operate on data currently in a DR chain and do not capture, shift, or update any of the chains. Instructions that are not implemented decode to the BYPASS instruction to ensure that the serial path between TDI and TDO is always connected (see Table 4-4 on page 164 for a list of implemented instructions).

See "JTAG and Boundary Scan" on page 650 for JTAG timing diagrams.

4.3.1 JTAG Interface Pins

The JTAG interface consists of five standard pins: TRST,TCK, TMS, TDI, and TDO. These pins and their associated reset state are given in Table 4-3 on page 157. Detailed information on each pin follows.

Table 4-3. JTAG Port Pins Reset State

Pin Name	Data Direction	Internal Pull-Up	Internal Pull-Down	Drive Strength	Drive Value
TRST	Input	Enabled	Disabled	N/A	N/A
TCK	Input	Enabled	Disabled	N/A	N/A
TMS	Input	Enabled	Disabled	N/A	N/A
TDI	Input	Enabled	Disabled	N/A	N/A
TDO	Output	Enabled	Disabled	2-mA driver	High-Z

4.3.1.1 Test Reset Input (TRST)

The $\overline{\mathtt{TRST}}$ pin is an asynchronous active Low input signal for initializing and resetting the JTAG TAP controller and associated JTAG circuitry. When $\overline{\mathtt{TRST}}$ is asserted, the TAP controller resets to the Test-Logic-Reset state and remains there while $\overline{\mathtt{TRST}}$ is asserted. When the TAP controller enters the Test-Logic-Reset state, the JTAG Instruction Register (IR) resets to the default instruction, IDCODE.

By default, the internal pull-up resistor on the TRST pin is enabled after reset. Changes to the pull-up resistor settings on GPIO Port B should ensure that the internal pull-up resistor remains enabled on PB7/TRST; otherwise JTAG communication could be lost.

4.3.1.2 Test Clock Input (TCK)

The TCK pin is the clock for the JTAG module. This clock is provided so the test logic can operate independently of any other system clocks. In addition, it ensures that multiple JTAG TAP controllers that are daisy-chained together can synchronously communicate serial test data between components. During normal operation, TCK is driven by a free-running clock with a nominal 50% duty cycle. When necessary, TCK can be stopped at 0 or 1 for extended periods of time. While TCK is stopped at 0 or 1, the state of the TAP controller does not change and data in the JTAG Instruction and Data Registers is not lost.

By default, the internal pull-up resistor on the ${ t TCK}$ pin is enabled after reset. This assures that no clocking occurs if the pin is not driven from an external source. The internal pull-up and pull-down resistors can be turned off to save internal power as long as the ${ t TCK}$ pin is constantly being driven by an external source.

4.3.1.3 Test Mode Select (TMS)

The TMS pin selects the next state of the JTAG TAP controller. TMS is sampled on the rising edge of TCK. Depending on the current TAP state and the sampled value of TMS, the next state is entered. Because the TMS pin is sampled on the rising edge of TCK, the *IEEE Standard 1149.1* expects the value on TMS to change on the falling edge of TCK.

Holding TMS high for five consecutive TCK cycles drives the TAP controller state machine to the Test-Logic-Reset state. When the TAP controller enters the Test-Logic-Reset state, the JTAG Instruction Register (IR) resets to the default instruction, IDCODE. Therefore, this sequence can be used as a reset mechanism, similar to asserting TRST. The JTAG Test Access Port state machine can be seen in its entirety in Figure 4-2 on page 160.

By default, the internal pull-up resistor on the TMS pin is enabled after reset. Changes to the pull-up resistor settings on GPIO Port C should ensure that the internal pull-up resistor remains enabled on PC1/TMS: otherwise JTAG communication could be lost.

4.3.1.4 Test Data Input (TDI)

The TDI pin provides a stream of serial information to the IR chain and the DR chains. TDI is sampled on the rising edge of TCK and, depending on the current TAP state and the current instruction, presents this data to the proper shift register chain. Because the TDI pin is sampled on the rising edge of TCK, the *IEEE Standard 1149.1* expects the value on TDI to change on the falling edge of TCK.

By default, the internal pull-up resistor on the TDI pin is enabled after reset. Changes to the pull-up resistor settings on GPIO Port C should ensure that the internal pull-up resistor remains enabled on PC2/TDI; otherwise JTAG communication could be lost.

4.3.1.5 Test Data Output (TDO)

The TDO pin provides an output stream of serial information from the IR chain or the DR chains. The value of TDO depends on the current TAP state, the current instruction, and the data in the chain being accessed. In order to save power when the JTAG port is not being used, the TDO pin is placed in an inactive drive state when not actively shifting out data. Because TDO can be connected to the TDI of another controller in a daisy-chain configuration, the *IEEE Standard 1149.1* expects the value on TDO to change on the falling edge of TCK.

By default, the internal pull-up resistor on the <code>TDO</code> pin is enabled after reset. This assures that the pin remains at a constant logic level when the JTAG port is not being used. The internal pull-up and pull-down resistors can be turned off to save internal power if a High-Z output value is acceptable during certain TAP controller states.

4.3.2 JTAG TAP Controller

The JTAG TAP controller state machine is shown in Figure 4-2 on page 160. The TAP controller state machine is reset to the Test-Logic-Reset state on the assertion of a Power-On-Reset (POR) or the assertion of TRST. Asserting the correct sequence on the TMS pin allows the JTAG module to shift in new instructions, shift in data, or idle during extended testing sequences. For detailed information on the function of the TAP controller and the operations that occur in each state, please refer to *IEEE Standard 1149.1*.

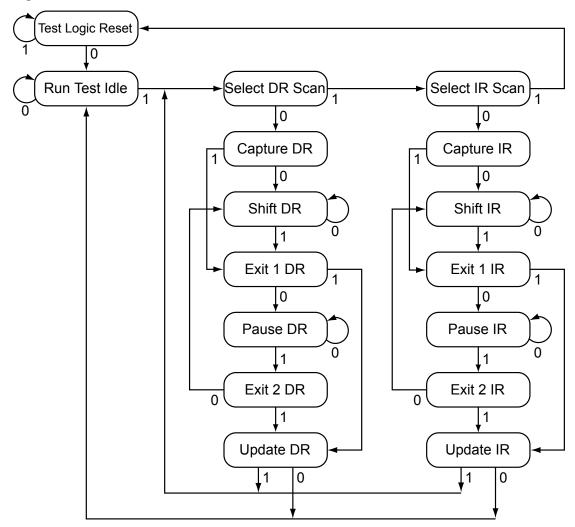


Figure 4-2. Test Access Port State Machine

4.3.3 Shift Registers

The Shift Registers consist of a serial shift register chain and a parallel load register. The serial shift register chain samples specific information during the TAP controller's CAPTURE states and allows this information to be shifted out of TDO during the TAP controller's SHIFT states. While the sampled data is being shifted out of the chain on TDO, new data is being shifted into the serial shift register on TDI. This new data is stored in the parallel load register during the TAP controller's UPDATE states. Each of the shift registers is discussed in detail in "Register Descriptions" on page 163.

4.3.4 Operational Considerations

There are certain operational considerations when using the JTAG module. Because the JTAG pins can be programmed to be GPIOs, board configuration and reset conditions on these pins must be considered. In addition, because the JTAG module has integrated ARM Serial Wire Debug, the method for switching between these two operational modes is described below.

4.3.4.1 **GPIO** Functionality

When the controller is reset with either a POR or RST, the JTAG/SWD port pins default to their JTAG/SWD configurations. The default configuration includes enabling digital functionality (setting **GPIODEN** to 1), enabling the pull-up resistors (setting **GPIOPUR** to 1), and enabling the alternate hardware function (setting **GPIOAFSEL** to 1) for the PB7 and PC[3:0] JTAG/SWD pins.

It is possible for software to configure these pins as GPIOs after reset by writing 0s to PB7 and PC[3:0] in the **GPIOAFSEL** register. If the user does not require the JTAG/SWD port for debugging or board-level testing, this provides five more GPIOs for use in the design.

Caution – It is possible to create a software sequence that prevents the debugger from connecting to the Stellaris microcontroller. If the program code loaded into flash immediately changes the JTAG pins to their GPIO functionality, the debugger may not have enough time to connect and halt the controller before the JTAG pin functionality switches. This may lock the debugger out of the part. This can be avoided with a software routine that restores JTAG functionality based on an external or software trigger.

The GPIO commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Protection is currently provided for the five JTAG/SWD pins (PB7 and PC[3:0]). Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 280) are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register (see page 290) has been unlocked and the appropriate bits of the **GPIO Commit (GPIOCR)** register (see page 291) have been set to 1.

Recovering a "Locked" Device

Note: The mass erase of the flash memory caused by the below sequence erases the entire flash memory, regardless of the settings in the **Flash Memory Protection Program Enable n** (**FMPPEn**) registers. Performing the sequence below does not affect the nonvolatile registers discussed in "Nonvolatile Register Programming" on page 235.

If software configures any of the JTAG/SWD pins as GPIO and loses the ability to communicate with the debugger, there is a debug sequence that can be used to recover the device. Performing a total of ten JTAG-to-SWD and SWD-to-JTAG switch sequences while holding the device in reset mass erases the flash memory. The sequence to recover the device is:

- 1. Assert and hold the $\overline{\mathtt{RST}}$ signal.
- 2. Apply power to the device.
- **3.** Perform the JTAG-to-SWD switch sequence.
- **4.** Perform the SWD-to-JTAG switch sequence.
- **5.** Perform the JTAG-to-SWD switch sequence.
- **6.** Perform the SWD-to-JTAG switch sequence.
- **7.** Perform the JTAG-to-SWD switch sequence.
- **8.** Perform the SWD-to-JTAG switch sequence.
- **9.** Perform the JTAG-to-SWD switch sequence.
- **10.** Perform the SWD-to-JTAG switch sequence.

- 11. Perform the JTAG-to-SWD switch sequence.
- **12.** Perform the SWD-to-JTAG switch sequence.
- **13.** Release the \overline{RST} signal.
- 14. Wait 400 ms.
- 15. Power-cycle the device.

The JTAG-to-SWD and SWD-to-JTAG switch sequences are described in "ARM Serial Wire Debug (SWD)" on page 162. When performing switch sequences for the purpose of recovering the debug capabilities of the device, only steps 1 and 2 of the switch sequence in the section called "JTAG-to-SWD Switching" on page 162 must be performed.

4.3.4.2 Communication with JTAG/SWD

Because the debug clock and the system clock can be running at different frequencies, care must be taken to maintain reliable communication with the JTAG/SWD interface. In the Capture-DR state, the result of the previous transaction, if any, is returned, together with a 3-bit ACK response. Software should check the ACK response to see if the previous operation has completed before initiating a new transaction. Alternatively, if the system clock is at least 8 times faster than the debug clock (TCK or SWCLK), the previous operation has enough time to complete and the ACK bits do not have to be checked.

4.3.4.3 ARM Serial Wire Debug (SWD)

In order to seamlessly integrate the ARM Serial Wire Debug (SWD) functionality, a serial-wire debugger must be able to connect to the Cortex-M3 core without having to perform, or have any knowledge of, JTAG cycles. This is accomplished with a SWD preamble that is issued before the SWD session begins.

The switching preamble used to enable the SWD interface of the SWJ-DP module starts with the TAP controller in the Test-Logic-Reset state. From here, the preamble sequences the TAP controller through the following states: Run Test Idle, Select DR, Select IR, Test Logic Reset, Test Logic Reset, Run Test Idle, Run Test Idle, Select DR, Select IR, Test Logic Reset, Test Logic Reset, Run Test Idle, Select DR, Select IR, and Test Logic Reset states.

Stepping through this sequences of the TAP state machine enables the SWD interface and disables the JTAG interface. For more information on this operation and the SWD interface, see the *ARM® Debug Interface V5 Architecture Specification*.

Because this sequence is a valid series of JTAG operations that could be issued, the ARM JTAG TAP controller is not fully compliant to the *IEEE Standard 1149.1*. This is the only instance where the ARM JTAG TAP controller does not meet full compliance with the specification. Due to the low probability of this sequence occurring during normal operation of the TAP controller, it should not affect normal performance of the JTAG interface.

JTAG-to-SWD Switching

To switch the operating mode of the Debug Access Port (DAP) from JTAG to SWD mode, the external debug hardware must send the switching preamble to the microcontroller. The 16-bit TMS/SWDIO command for switching to SWD mode is defined as b1110.0111.1001.1110, transmitted LSB first. This command can also be represented as 0xE79E when transmitted LSB first. The complete switch sequence should consist of the following transactions on the TCK/SWCLK and TMS/SWDIO signals:

- 1. Send at least 50 TCK/SWCLK cycles with TMS/SWDIO High to ensure that both JTAG and SWD are in their reset states.
- 2. Send the 16-bit JTAG-to-SWD switch command, 0xE79E, on TMS/SWDIO.
- 3. Send at least 50 TCK/SWCLK cycles with TMS/SWDIO High to ensure that if SWJ-DP was already in SWD mode before sending the switch sequence, the SWD goes into the line reset state.

To verify that the Debug Access Port (DAP) has switched to the Serial Wire Debug (SWD) operating mode, perform a SWD READID operation. The ID value can be compared against the device's known ID to verify the switch.

SWD-to-JTAG Switching

To switch the operating mode of the Debug Access Port (DAP) from SWD to JTAG mode, the external debug hardware must send a switch command to the microcontroller. The 16-bit TMS/SWDIO command for switching to JTAG mode is defined as b1110.0111.0011.1100, transmitted LSB first. This command can also be represented as 0xE73C when transmitted LSB first. The complete switch sequence should consist of the following transactions on the TCK/SWCLK and TMS/SWDIO signals:

- 1. Send at least 50 TCK/SWCLK cycles with TMS/SWDIO High to ensure that both JTAG and SWD are in their reset states.
- 2. Send the 16-bit SWD-to-JTAG switch command, 0xE73C, on TMS/SWDIO.
- 3. Send at least 50 TCK/SWCLK cycles with TMS/SWDIO High to ensure that if SWJ-DP was already in JTAG mode before sending the switch sequence, the JTAG goes into the Test Logic Reset state.

To verify that the Debug Access Port (DAP) has switched to the JTAG operating mode, set the JTAG Instruction Register (IR) to the IDCODE instruction and shift out the Data Register (DR). The DR value can be compared against the device's known IDCODE to verify the switch.

4.4 Initialization and Configuration

After a Power-On-Reset or an external reset (\overline{RST}), the JTAG pins are automatically configured for JTAG communication. No user-defined initialization or configuration is needed. However, if the user application changes these pins to their GPIO function, they must be configured back to their JTAG functionality before JTAG communication can be restored. This is done by enabling the five JTAG pins (PB7 and PC[3:0]) for their alternate function using the GPIOAFSEL register. In addition to enabling the alternate functions, any other changes to the GPIO pad configurations on the five JTAG pins (PB7 and PC[3:0]) should be reverted to their default settings.

4.5 Register Descriptions

There are no APB-accessible registers in the JTAG TAP Controller or Shift Register chains. The registers within the JTAG controller are all accessed serially through the TAP Controller. The registers can be broken down into two main categories: Instruction Registers and Data Registers.

4.5.1 Instruction Register (IR)

The JTAG TAP Instruction Register (IR) is a four-bit serial scan chain connected between the JTAG $\tiny{\text{TDI}}$ and $\tiny{\text{TDO}}$ pins with a parallel load register. When the TAP Controller is placed in the correct states, bits can be shifted into the Instruction Register. Once these bits have been shifted into the chain and updated, they are interpreted as the current instruction. The decode of the Instruction

Register bits is shown in Table 4-4 on page 164. A detailed explanation of each instruction, along with its associated Data Register, follows.

Table 4-4. JTAG Instruction Register Commands

IR[3:0]	Instruction	Description
0000	EXTEST	Drives the values preloaded into the Boundary Scan Chain by the SAMPLE/PRELOAD instruction onto the pads.
0001	INTEST	Drives the values preloaded into the Boundary Scan Chain by the SAMPLE/PRELOAD instruction into the controller.
0010	SAMPLE / PRELOAD	Captures the current I/O values and shifts the sampled values out of the Boundary Scan Chain while new preload data is shifted in.
1000	ABORT	Shifts data into the ARM Debug Port Abort Register.
1010	DPACC	Shifts data into and out of the ARM DP Access Register.
1011	APACC	Shifts data into and out of the ARM AC Access Register.
1110	IDCODE	Loads manufacturing information defined by the <i>IEEE Standard 1149.1</i> into the IDCODE chain and shifts it out.
1111	BYPASS	Connects TDI to TDO through a single Shift Register chain.
All Others	Reserved	Defaults to the BYPASS instruction to ensure that \mathtt{TDI} is always connected to \mathtt{TDO} .

4.5.1.1 EXTEST Instruction

The EXTEST instruction is not associated with its own Data Register chain. The EXTEST instruction uses the data that has been preloaded into the Boundary Scan Data Register using the SAMPLE/PRELOAD instruction. When the EXTEST instruction is present in the Instruction Register, the preloaded data in the Boundary Scan Data Register associated with the outputs and output enables are used to drive the GPIO pads rather than the signals coming from the core. This allows tests to be developed that drive known values out of the controller, which can be used to verify connectivity. While the EXTEST instruction is present in the Instruction Register, the Boundary Scan Data Register can be accessed to sample and shift out the current data and load new data into the Boundary Scan Data Register.

4.5.1.2 INTEST Instruction

The INTEST instruction is not associated with its own Data Register chain. The INTEST instruction uses the data that has been preloaded into the Boundary Scan Data Register using the SAMPLE/PRELOAD instruction. When the INTEST instruction is present in the Instruction Register, the preloaded data in the Boundary Scan Data Register associated with the inputs are used to drive the signals going into the core rather than the signals coming from the GPIO pads. This allows tests to be developed that drive known values into the controller, which can be used for testing. It is important to note that although the $\overline{\text{RST}}$ input pin is on the Boundary Scan Data Register chain, it is only observable. While the INTEXT instruction is present in the Instruction Register, the Boundary Scan Data Register can be accessed to sample and shift out the current data and load new data into the Boundary Scan Data Register.

4.5.1.3 SAMPLE/PRELOAD Instruction

The SAMPLE/PRELOAD instruction connects the Boundary Scan Data Register chain between TDI and TDO. This instruction samples the current state of the pad pins for observation and preloads new test data. Each GPIO pad has an associated input, output, and output enable signal. When the TAP controller enters the Capture DR state during this instruction, the input, output, and output-enable signals to each of the GPIO pads are captured. These samples are serially shifted out of TDO while

the TAP controller is in the Shift DR state and can be used for observation or comparison in various tests.

While these samples of the inputs, outputs, and output enables are being shifted out of the Boundary Scan Data Register, new data is being shifted into the Boundary Scan Data Register from TDI. Once the new data has been shifted into the Boundary Scan Data Register, the data is saved in the parallel load registers when the TAP controller enters the Update DR state. This update of the parallel load register preloads data into the Boundary Scan Data Register that is associated with each input, output, and output enable. This preloaded data can be used with the EXTEST and INTEST instructions to drive data into or out of the controller. Please see "Boundary Scan Data Register" on page 166 for more information.

4.5.1.4 ABORT Instruction

The ABORT instruction connects the associated ABORT Data Register chain between TDI and TDO. This instruction provides read and write access to the ABORT Register of the ARM Debug Access Port (DAP). Shifting the proper data into this Data Register clears various error bits or initiates a DAP abort of a previous request. Please see the "ABORT Data Register" on page 167 for more information.

4.5.1.5 DPACC Instruction

The DPACC instruction connects the associated DPACC Data Register chain between TDI and TDO. This instruction provides read and write access to the DPACC Register of the ARM Debug Access Port (DAP). Shifting the proper data into this register and reading the data output from this register allows read and write access to the ARM debug and status registers. Please see "DPACC Data Register" on page 167 for more information.

4.5.1.6 APACC Instruction

The APACC instruction connects the associated APACC Data Register chain between TDI and TDO. This instruction provides read and write access to the APACC Register of the ARM Debug Access Port (DAP). Shifting the proper data into this register and reading the data output from this register allows read and write access to internal components and buses through the Debug Port. Please see "APACC Data Register" on page 167 for more information.

4.5.1.7 IDCODE Instruction

The IDCODE instruction connects the associated IDCODE Data Register chain between <code>TDI</code> and <code>TDO</code>. This instruction provides information on the manufacturer, part number, and version of the ARM core. This information can be used by testing equipment and debuggers to automatically configure their input and output data streams. IDCODE is the default instruction that is loaded into the JTAG Instruction Register when a Power-On-Reset (POR) is asserted, <code>TRST</code> is asserted, or the Test-Logic-Reset state is entered. Please see "IDCODE Data Register" on page 166 for more information.

4.5.1.8 BYPASS Instruction

The BYPASS instruction connects the associated BYPASS Data Register chain between TDI and TDO. This instruction is used to create a minimum length serial path between the TDI and TDO ports. The BYPASS Data Register is a single-bit shift register. This instruction improves test efficiency by allowing components that are not needed for a specific test to be bypassed in the JTAG scan chain by loading them with the BYPASS instruction. Please see "BYPASS Data Register" on page 166 for more information.

4.5.2 Data Registers

The JTAG module contains six Data Registers. These include: IDCODE, BYPASS, Boundary Scan, APACC, DPACC, and ABORT serial Data Register chains. Each of these Data Registers is discussed in the following sections.

4.5.2.1 IDCODE Data Register

The format for the 32-bit IDCODE Data Register defined by the *IEEE Standard 1149.1* is shown in Figure 4-3 on page 166. The standard requires that every JTAG-compliant device implement either the IDCODE instruction or the BYPASS instruction as the default instruction. The LSB of the IDCODE Data Register is defined to be a 1 to distinguish it from the BYPASS instruction, which has an LSB of 0. This allows auto configuration test tools to determine which instruction is the default instruction.

The major uses of the JTAG port are for manufacturer testing of component assembly, and program development and debug. To facilitate the use of auto-configuration debug tools, the IDCODE instruction outputs a value of 0x3BA0.0477. This allows the debuggers to automatically configure themselves to work correctly with the Cortex-M3 during debug.

Figure 4-3. IDCODE Register Format



4.5.2.2 BYPASS Data Register

The format for the 1-bit BYPASS Data Register defined by the *IEEE Standard 1149.1* is shown in Figure 4-4 on page 166. The standard requires that every JTAG-compliant device implement either the BYPASS instruction or the IDCODE instruction as the default instruction. The LSB of the BYPASS Data Register is defined to be a 0 to distinguish it from the IDCODE instruction, which has an LSB of 1. This allows auto configuration test tools to determine which instruction is the default instruction.

Figure 4-4. BYPASS Register Format

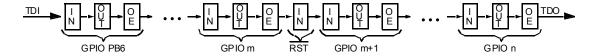
4.5.2.3 Boundary Scan Data Register

The format of the Boundary Scan Data Register is shown in Figure 4-5 on page 167. Each GPIO pin, starting with a GPIO pin next to the JTAG port pins, is included in the Boundary Scan Data Register. Each GPIO pin has three associated digital signals that are included in the chain. These signals are input, output, and output enable, and are arranged in that order as can be seen in the figure.

When the Boundary Scan Data Register is accessed with the SAMPLE/PRELOAD instruction, the input, output, and output enable from each digital pad are sampled and then shifted out of the chain to be verified. The sampling of these values occurs on the rising edge of TCK in the Capture DR state of the TAP controller. While the sampled data is being shifted out of the Boundary Scan chain in the Shift DR state of the TAP controller, new data can be preloaded into the chain for use with

the EXTEST and INTEST instructions. These instructions either force data out of the controller, with the EXTEST instruction, or into the controller, with the INTEST instruction.

Figure 4-5. Boundary Scan Register Format



4.5.2.4 APACC Data Register

The format for the 35-bit APACC Data Register defined by ARM is described in the *ARM® Debug Interface V5 Architecture Specification*.

4.5.2.5 DPACC Data Register

The format for the 35-bit DPACC Data Register defined by ARM is described in the *ARM® Debug Interface V5 Architecture Specification*.

4.5.2.6 ABORT Data Register

The format for the 35-bit ABORT Data Register defined by ARM is described in the *ARM® Debug Interface V5 Architecture Specification*.

5 System Control

System control determines the overall operation of the device. It provides information about the device, controls the clocking to the core and individual peripherals, and handles reset detection and reporting.

5.1 Signal Description

Table 5-1 on page 168 and Table 5-2 on page 168 list the external signals of the System Control module and describe the function of each. The NMI signal is the alternate function for and functions as a GPIO after reset. under commit protection and require a special process to be configured as any alternate function or to subsequently return to the GPIO function, see "Commit Control" on page 266. The column in the table below titled "Pin Assignment" lists the GPIO pin placement for the NMI signal. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the NMI function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258. The remaining signals (with the word "fixed" in the Pin Assignment column) have a fixed pin assignment and function.

Table 5-1. System Control & Clocks Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
CMOD0	65	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	76	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
osc0	48	I	Analog	Main oscillator crystal input or an external clock reference input.
OSC1	49	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
RST	64	I	TTL	System reset input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 5-2. System Control & Clocks Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
CMOD0	E11	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	B10	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
osc0	L11	I	Analog	Main oscillator crystal input or an external clock reference input.
OSC1	M11	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
RST	H11	I	TTL	System reset input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

5.2 Functional Description

The System Control module provides the following capabilities:

■ Device identification (see "Device Identification" on page 169)

- Local control, such as reset (see "Reset Control" on page 169), power (see "Power Control" on page 173) and clock control (see "Clock Control" on page 174)
- System control (Run, Sleep, and Deep-Sleep modes); see "System Control" on page 179

5.2.1 Device Identification

Several read-only registers provide software with information on the microcontroller, such as version, part number, SRAM size, flash size, and other features. See the **DID0**, **DID1**, and **DC0-DC4** registers.

5.2.2 Reset Control

This section discusses aspects of hardware functions during reset as well as system software requirements following the reset sequence.

5.2.2.1 CMOD0 and CMOD1 Test-Mode Control Pins

Two pins, CMOD0 and CMOD1, are defined for internal use for testing the microcontroller during manufacture. They have no end-user function and should not be used. The CMOD pins should be connected to ground.

5.2.2.2 Reset Sources

The controller has five sources of reset:

- **1.** External reset input pin (\overline{RST}) assertion; see "External \overline{RST} Pin" on page 170.
- 2. Power-on reset (POR); see "Power-On Reset (POR)" on page 170.
- 3. Internal brown-out (BOR) detector; see "Brown-Out Reset (BOR)" on page 171.
- **4.** Software-initiated reset (with the software reset registers); see "Software Reset" on page 172.
- 5. A watchdog timer reset condition violation; see "Watchdog Timer Reset" on page 172.

Table 5-3 provides a summary of results of the various reset operations.

Table 5-3. Reset Sources

Reset Source	Core Reset?	JTAG Reset?	On-Chip Peripherals Reset?
Power-On Reset	Yes	Yes	Yes
RST	Yes	Pin Config Only	Yes
Brown-Out Reset	Yes	No	Yes
Software System Request Reset ^a	Yes	No	Yes
Software Peripheral Reset	No	No	Yes ^b
Watchdog Reset	Yes	No	Yes

a. By using the SYSRESREQ bit in the ARM Cortex-M3 Application Interrupt and Reset Control (APINT) register

After a reset, the **Reset Cause (RESC)** register is set with the reset cause. The bits in this register are sticky and maintain their state across multiple reset sequences, except when an internal POR or an external reset is the cause, and then all the other bits in the **RESC** register are cleared except for the POR or EXT indicator.

b. Programmable on a module-by-module basis using the Software Reset Control Registers.

5.2.2.3 Power-On Reset (POR)

Note: The power-on reset also resets the JTAG controller. An external reset does not.

The internal Power-On Reset (POR) circuit monitors the power supply voltage (V_{DD}) and generates a reset signal to all of the internal logic including JTAG when the power supply ramp reaches a threshold value (V_{TH}). The microcontroller must be operating within the specified operating parameters when the on-chip power-on reset pulse is complete. The 3.3-V power supply to the microcontroller must reach 3.0 V within 10 msec of V_{DD} crossing 2.0 V to guarantee proper operation. For applications that require the use of an external reset signal to hold the microcontroller in reset longer than the internal POR, the \overline{RST} input may be used as discussed in "External \overline{RST} Pin" on page 170.

The Power-On Reset sequence is as follows:

- **1.** The microcontroller waits for internal POR to go inactive.
- The internal reset is released and the core loads from memory the initial stack pointer, the initial program counter, and the first instruction designated by the program counter, and then begins execution.

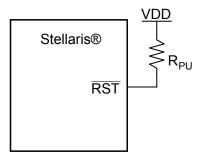
The internal POR is only active on the initial power-up of the microcontroller. The Power-On Reset timing is shown in Figure 20-6 on page 652.

5.2.2.4 External RST Pin

Note: It is recommended that the trace for the \overline{RST} signal must be kept as short as possible. Be sure to place any components connected to the \overline{RST} signal as close to the microcontroller as possible.

If the application only uses the internal POR circuit, the $\overline{\text{RST}}$ input must be connected to the power supply (V_{DD}) through an optional pull-up resistor (0 to 100K Ω) as shown in Figure 5-1 on page 170.

Figure 5-1. Basic RST Configuration



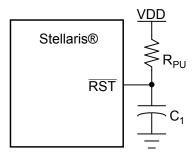
 R_{PU} = 0 to 100 k Ω

The external reset pin (RST) resets the microcontroller including the core and all the on-chip peripherals except the JTAG TAP controller (see "JTAG Interface" on page 155). The external reset sequence is as follows:

- 1. The external reset pin (\overline{RST}) is asserted for the duration specified by T_{MIN} and then de-asserted (see "Reset" on page 651).
- 2. The internal reset is released and the core loads from memory the initial stack pointer, the initial program counter, and the first instruction designated by the program counter, and then begins execution.

To improve noise immunity and/or to delay reset at power up, the $\overline{\mathtt{RST}}$ input may be connected to an RC network as shown in Figure 5-2 on page 171.

Figure 5-2. External Circuitry to Extend Power-On Reset

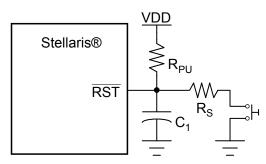


 R_{PU} = 1 k Ω to 100 k Ω

 $C_1 = 1 \text{ nF to } 10 \mu\text{F}$

If the application requires the use of an external reset switch, Figure 5-3 on page 171 shows the proper circuitry to use.

Figure 5-3. Reset Circuit Controlled by Switch



Typical $R_{PU} = 10 \text{ k}\Omega$

Typical $R_S = 470 \Omega$

 $C_1 = 10 \text{ nF}$

The R_{PIJ} and C₁ components define the power-on delay.

The external reset timing is shown in Figure 20-5 on page 652.

5.2.2.5 Brown-Out Reset (BOR)

A drop in the input voltage resulting in the assertion of the internal brown-out detector can be used to reset the controller. This is initially disabled and may be enabled by software.

The system provides a brown-out detection circuit that triggers if the power supply (V_{DD}) drops below a brown-out threshold voltage (V_{BTH}) . If a brown-out condition is detected, the system may generate a controller interrupt or a system reset.

Brown-out resets are controlled with the **Power-On and Brown-Out Reset Control (PBORCTL)** register. The BORIOR bit in the **PBORCTL** register must be set for a brown-out condition to trigger a reset.

The brown-out reset is equivalent to an assertion of the external $\overline{\mathtt{RST}}$ input and the reset is held active until the proper V_{DD} level is restored. The **RESC** register can be examined in the reset interrupt handler to determine if a Brown-Out condition was the cause of the reset, thus allowing software to determine what actions are required to recover.

The internal Brown-Out Reset timing is shown in Figure 20-7 on page 652.

5.2.2.6 Software Reset

Software can reset a specific peripheral or generate a reset to the entire system.

Peripherals can be individually reset by software via three registers that control reset signals to each peripheral (see the **SRCRn** registers). If the bit position corresponding to a peripheral is set and subsequently cleared, the peripheral is reset. The encoding of the reset registers is consistent with the encoding of the clock gating control for peripherals and on-chip functions (see "System Control" on page 179). Note that all reset signals for all clocks of the specified unit are asserted as a result of a software-initiated reset.

The entire system can be reset by software by setting the SYSRESETREQ bit in the Cortex-M3 Application Interrupt and Reset Control register resets the entire system including the core. The software-initiated system reset sequence is as follows:

- **1.** A software system reset is initiated by writing the SYSRESETREQ bit in the ARM Cortex-M3 Application Interrupt and Reset Control register.
- 2. An internal reset is asserted.
- The internal reset is deasserted and the controller loads from memory the initial stack pointer, the initial program counter, and the first instruction designated by the program counter, and then begins execution.

The software-initiated system reset timing is shown in Figure 20-8 on page 653.

5.2.2.7 Watchdog Timer Reset

The watchdog timer module's function is to prevent system hangs. The watchdog timer can be configured to generate an interrupt to the controller on its first time-out, and to generate a reset signal on its second time-out.

After the first time-out event, the 32-bit counter is reloaded with the value of the **Watchdog Timer Load (WDTLOAD)** register, and the timer resumes counting down from that value. If the timer counts down to its zero state again before the first time-out interrupt is cleared, and the reset signal has been enabled, the watchdog timer asserts its reset signal to the system. The watchdog timer reset sequence is as follows:

- 1. The watchdog timer times out for the second time without being serviced.
- 2. An internal reset is asserted.
- The internal reset is released and the controller loads from memory the initial stack pointer, the initial program counter, the first instruction designated by the program counter, and begins execution.

The watchdog reset timing is shown in Figure 20-9 on page 653.

5.2.3 Power Control

The Stellaris microcontroller provides an integrated LDO regulator that is used to provide power to the majority of the controller's internal logic. For power reduction, the LDO regulator provides software a mechanism to adjust the regulated value, in small increments (VSTEP), over the range of 2.25 V to 2.75 V (inclusive)—or 2.5 V \pm 10%. The adjustment is made by changing the value of the VADJ field in the **LDO Power Control (LDOPCTL)** register.

Figure 5-4 on page 174 shows the power architecture.

Note: On the printed circuit board, use the LDO output as the source of VDD25 input. Do not use an external regulator to supply the voltage to VDD25. In addition, the LDO requires decoupling capacitors. See "On-Chip Low Drop-Out (LDO) Regulator Characteristics" on page 646.

VDDA must be supplied with 3.3 V, or the microcontroller does not function properly. VDDA is the supply for all of the analog circuitry on the device, including the LDO and the clock circuitry.

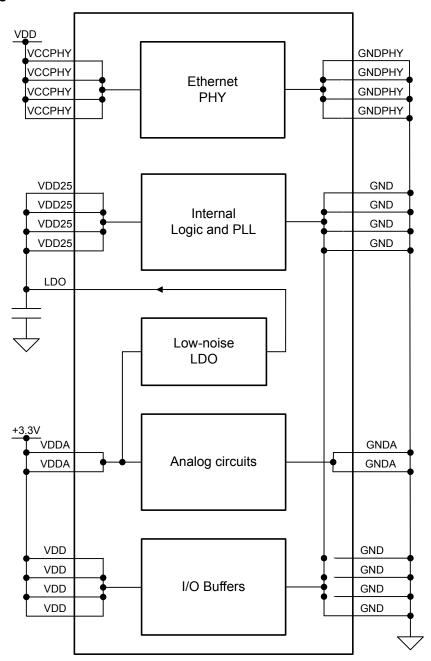


Figure 5-4. Power Architecture

5.2.4 Clock Control

System control determines the control of clocks in this part.

5.2.4.1 Fundamental Clock Sources

There are multiple clock sources for use in the device:

■ Internal Oscillator (IOSC). The internal oscillator is an on-chip clock source. It does not require the use of any external components. The frequency of the internal oscillator is 12 MHz ± 30%.

Applications that do not depend on accurate clock sources may use this clock source to reduce system cost. The internal oscillator is the clock source the device uses during and following POR. If the main oscillator is required, software must enable the main oscillator following reset and allow the main oscillator to stabilize before changing the clock reference.

- Main Oscillator (MOSC). The main oscillator provides a frequency-accurate clock source by one of two means: an external single-ended clock source is connected to the OSC0 input pin, or an external crystal is connected across the OSC0 input and OSC1 output pins. If the PLL is being used, the crystal value must be one of the supported frequencies between 3.579545 MHz through 8.192 MHz (inclusive). If the PLL is not being used, the crystal may be any one of the supported frequencies between 1 MHz and 8.192 MHz. The single-ended clock source range is from DC through the specified speed of the device. The supported crystals are listed in the XTAL bit field in the RCC register (see page 190).
- Internal 30-kHz Oscillator. The internal 30-kHz oscillator is similar to the internal oscillator, except that it provides an operational frequency of 30 kHz ± 50%. It is intended for use during Deep-Sleep power-saving modes. This power-savings mode benefits from reduced internal switching and also allows the main oscillator to be powered down.

The internal system clock (SysClk), is derived from any of the above sources plus two others: the output of the main internal PLL, and the internal oscillator divided by four (3 MHz \pm 30%). The frequency of the PLL clock reference must be in the range of 3.579545 MHz to 8.192 MHz (inclusive). Table 5-4 on page 175 shows how the various clock sources can be used in a system.

Clock Source **Drive PLL?** Used as SysClk? Internal Oscillator (12 MHz) No BYPASS = 1 Yes BYPASS = 1, OSCSRC = 0x1Internal Oscillator divide by 4 (3 No BYPASS = 1 Yes BYPASS = 1, OSCSRC = 0x2MHz) Main Oscillator BYPASS = 0, OSCSRC = Yes Yes BYPASS = 1, OSCSRC = 0x0Internal 30-kHz Oscillator No BYPASS = 1 Yes BYPASS = 1, OSCSRC = 0x3

Table 5-4. Clock Source Options

5.2.4.2 Clock Configuration

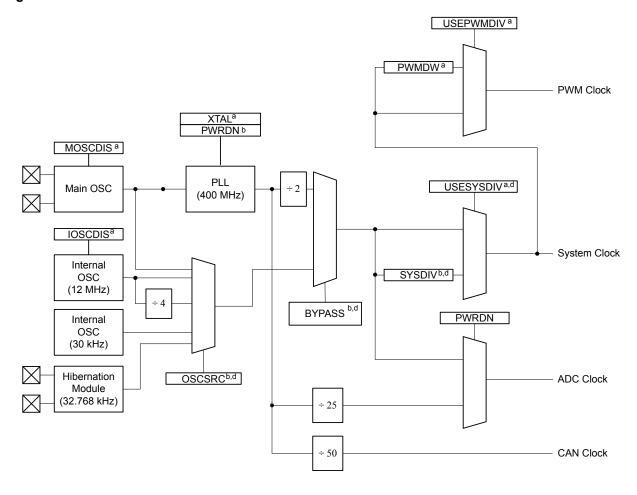
The Run-Mode Clock Configuration (RCC) and Run-Mode Clock Configuration 2 (RCC2) registers provide control for the system clock. The RCC2 register is provided to extend fields that offer additional encodings over the RCC register. When used, the RCC2 register field values are used by the logic over the corresponding field in the RCC register. In particular, RCC2 provides for a larger assortment of clock configuration options. These registers control the following clock functionality:

- Source of clocks in sleep and deep-sleep modes
- System clock derived from PLL or other clock source
- Enabling/disabling of oscillators and PLL
- Clock divisors
- Crystal input selection

Figure 5-5 on page 176 shows the logic for the main clock tree. The peripheral blocks are driven by the system clock signal and can be individually enabled/disabled. The ADC clock signal is automatically divided down to 16 MHz for proper ADC operation. The PWM clock signal is a synchronous divide of the system clock to provide the PWM circuit with more range (set with PWMDIV in **RCC**).

Note: When the ADC module is in operation, the system clock must be at least 16 MHz.

Figure 5-5. Main Clock Tree



- a. Control provided by RCC register bit/field.
- b. Control provided by RCC register bit/field or RCC2 register bit/field, if overridden with RCC2 register bit USERCC2.
- c. Control provided by RCC2 register bit/field.
- d. Also may be controlled by DSLPCLKCFG when in deep sleep mode.

Note: The figure above shows all features available on all Stellaris® Fury-class devices. Not all peripherals may be available on this device.

In the RCC register, the SYSDIV field specifies which divisor is used to generate the system clock from either the PLL output or the oscillator source (depending on how the BYPASS bit in this register is configured). When using the PLL, the VCO frequency of 400 MHz is predivided by 2 before the

divisor is applied. Table 5-5 shows how the SYSDIV encoding affects the system clock frequency, depending on whether the PLL is used (BYPASS=0) or another clock source is used (BYPASS=1). The divisor is equivalent to the SYSDIV encoding plus 1. For a list of possible clock sources, see Table 5-4 on page 175.

Table 5-5. Possible System Clock Frequencies Using the SYSDIV Field

SYSDIV	Divisor	Frequency (BYPASS=0)	Frequency (BYPASS=1)	StellarisWare Parameter ^a
0x0	/1	reserved	Clock source frequency/2	SYSCTL_SYSDIV_1b
0x1	/2	reserved	Clock source frequency/2	SYSCTL_SYSDIV_2
0x2	/3	reserved	Clock source frequency/3	SYSCTL_SYSDIV_3
0x3	/4	50 MHz	Clock source frequency/4	SYSCTL_SYSDIV_4
0x4	/5	40 MHz	Clock source frequency/5	SYSCTL_SYSDIV_5
0x5	/6	33.33 MHz	Clock source frequency/6	SYSCTL_SYSDIV_6
0x6	/7	28.57 MHz	Clock source frequency/7	SYSCTL_SYSDIV_7
0x7	/8	25 MHz	Clock source frequency/8	SYSCTL_SYSDIV_8
0x8	/9	22.22 MHz	Clock source frequency/9	SYSCTL_SYSDIV_9
0x9	/10	20 MHz	Clock source frequency/10	SYSCTL_SYSDIV_10
0xA	/11	18.18 MHz	Clock source frequency/11	SYSCTL_SYSDIV_11
0xB	/12	16.67 MHz	Clock source frequency/12	SYSCTL_SYSDIV_12
0xC	/13	15.38 MHz	Clock source frequency/13	SYSCTL_SYSDIV_13
0xD	/14	14.29 MHz	Clock source frequency/14	SYSCTL_SYSDIV_14
0xE	/15	13.33 MHz	Clock source frequency/15	SYSCTL_SYSDIV_15
0xF	/16	12.5 MHz (default)	Clock source frequency/16	SYSCTL_SYSDIV_16

 $a.\ This\ parameter\ is\ used\ in\ functions\ such\ as\ SysCtlClockSet()\ in\ the\ Stellar is\ Peripheral\ Driver\ Library.$

The SYSDIV2 field in the **RCC2** register is 2 bits wider than the SYSDIV field in the **RCC** register so that additional larger divisors up to /64 are possible, allowing a lower system clock frequency for improved Deep Sleep power consumption. When using the PLL, the VCO frequency of 400 MHz is predivided by 2 before the divisor is applied. The divisor is equivalent to the SYSDIV2 encoding plus 1. Table 5-6 shows how the SYSDIV2 encoding affects the system clock frequency, depending on whether the PLL is used (BYPASS2=0) or another clock source is used (BYPASS2=1). For a list of possible clock sources, see Table 5-4 on page 175.

Table 5-6. Examples of Possible System Clock Frequencies Using the SYSDIV2 Field

SYSDIV2	Divisor	Frequency (BYPASS2=0)	Frequency (BYPASS2=1)	StellarisWare Parameter ^a
0x00	/1	reserved	Clock source frequency/2	SYSCTL_SYSDIV_1b
0x01	/2	reserved	Clock source frequency/2	SYSCTL_SYSDIV_2
0x02	/3	reserved	Clock source frequency/3	SYSCTL_SYSDIV_3
0x03	/4	50 MHz	Clock source frequency/4	SYSCTL_SYSDIV_4
0x04	/5	40 MHz	Clock source frequency/5	SYSCTL_SYSDIV_5
0x05	/6	33.33 MHz	Clock source frequency/6	SYSCTL_SYSDIV_6
0x06	/7	28.57 MHz	Clock source frequency/7	SYSCTL_SYSDIV_7
0x07	/8	25 MHz	Clock source frequency/8	SYSCTL_SYSDIV_8

b. SYSCTL_SYSDIV_1 does not set the USESYSDIV bit. As a result, using this parameter without enabling the PLL results in the system clock having the same frequency as the clock source.

Table 5-6. Examples of Possible System Clock Frequencies Using the SYSDIV2 Field (continued)

SYSDIV2	Divisor	Frequency (BYPASS2=0)	Frequency (BYPASS2=1)	StellarisWare Parameter ^a
0x08	/9	22.22 MHz	Clock source frequency/9	SYSCTL_SYSDIV_9
0x09	/10	20 MHz	Clock source frequency/10	SYSCTL_SYSDIV_10
0x3F	/64	3.125 MHz	Clock source frequency/64	SYSCTL_SYSDIV_64

a. This parameter is used in functions such as SysCtlClockSet() in the Stellaris Peripheral Driver Library.

5.2.4.3 Crystal Configuration for the Main Oscillator (MOSC)

The main oscillator supports the use of a select number of crystals. If the main oscillator is used by the PLL as a reference clock, the supported range of crystals is 3.579545 to 8.192 MHz, otherwise, the range of supported crystals is 1 to 8.192 MHz.

The XTAL bit in the **RCC** register (see page 190) describes the available crystal choices and default programming values.

Software configures the **RCC** register XTAL field with the crystal number. If the PLL is used in the design, the XTAL field value is internally translated to the PLL settings.

5.2.4.4 Main PLL Frequency Configuration

The main PLL is disabled by default during power-on reset and is enabled later by software if required. Software specifies the output divisor to set the system clock frequency, and enables the main PLL to drive the output. The PLL operates at 400 MHz, but is divided by two prior to the application of the output divisor.

If the main oscillator provides the clock reference to the main PLL, the translation provided by hardware and used to program the PLL is available for software in the **XTAL to PLL Translation** (**PLLCFG**) register (see page 194). The internal translation provides a translation within \pm 1% of the targeted PLL VCO frequency. Table 20-9 on page 649 shows the actual PLL frequency and error for a given crystal choice.

The Crystal Value field (XTAL) in the **Run-Mode Clock Configuration (RCC)** register (see page 190) describes the available crystal choices and default programming of the **PLLCFG** register. Any time the XTAL field changes, the new settings are translated and the internal PLL settings are updated.

5.2.4.5 PLL Modes

The PLL has two modes of operation: Normal and Power-Down

- Normal: The PLL multiplies the input clock reference and drives the output.
- Power-Down: Most of the PLL internal circuitry is disabled and the PLL does not drive the output.

The modes are programmed using the RCC/RCC2 register fields (see page 190 and page 195).

5.2.4.6 PLL Operation

If a PLL configuration is changed, the PLL output frequency is unstable until it reconverges (relocks) to the new setting. The time between the configuration change and relock is T_{READY} (see Table 20-8 on page 648). During the relock time, the affected PLL is not usable as a clock reference.

b. SYSCTL_SYSDIV_1 does not set the USESYSDIV bit. As a result, using this parameter without enabling the PLL results in the system clock having the same frequency as the clock source.

PLL is changed by one of the following:

- Change to the XTAL value in the **RCC** register—writes of the same value do not cause a relock.
- Change in the PLL from Power-Down to Normal mode.

A counter is defined to measure the T_{READY} requirement. The counter is clocked by the main oscillator. The range of the main oscillator has been taken into account and the down counter is set to 0x1200 (that is, ~600 μ s at an 8.192 MHz external oscillator clock). Hardware is provided to keep the PLL from being used as a system clock until the T_{READY} condition is met after one of the two changes above. It is the user's responsibility to have a stable clock source (like the main oscillator) before the **RCC/RCC2** register is switched to use the PLL.

If the main PLL is enabled and the system clock is switched to use the PLL in one step, the system control hardware continues to clock the controller from the oscillator selected by the RCC/RCC2 register until the main PLL is stable (T_{READY} time met), after which it changes to the PLL. Software can use many methods to ensure that the system is clocked from the main PLL, including periodically polling the PLLLRIS bit in the Raw Interrupt Status (RIS) register, and enabling the PLL Lock interrupt.

5.2.5 System Control

For power-savings purposes, the **RCGCn**, **SCGCn**, and **DCGCn** registers control the clock gating logic for each peripheral or block in the system while the controller is in Run, Sleep, and Deep-Sleep mode, respectively.

There are four levels of operation for the device defined as:

- Run Mode. In Run mode, the controller actively executes code. Run mode provides normal operation of the processor and all of the peripherals that are currently enabled by the RCGCn registers. The system clock can be any of the available clock sources including the PLL.
- Sleep Mode. In Sleep mode, the clock frequency of the active peripherals is unchanged, but the processor and the memory subsystem are not clocked and therefore no longer execute code. Sleep mode is entered by the Cortex-M3 core executing a WFI(Wait for Interrupt) instruction. Any properly configured interrupt event in the system will bring the processor back into Run mode. See "Power Management" on page 86 for more details.
 - Peripherals are clocked that are enabled in the **SCGCn** register when auto-clock gating is enabled (see the **RCC** register) or the **RCGCn** register when the auto-clock gating is disabled. The system clock has the same source and frequency as that during Run mode.
- Deep-Sleep Mode. In Deep-Sleep mode, the clock frequency of the active peripherals may change (depending on the Run mode clock configuration) in addition to the processor clock being stopped. An interrupt returns the device to Run mode from one of the sleep modes; the sleep modes are entered on request from the code. Deep-Sleep mode is entered by first writing the Deep Sleep Enable bit in the ARM Cortex-M3 NVIC system control register and then executing a WFI instruction. Any properly configured interrupt event in the system will bring the processor back into Run mode. See "Power Management" on page 86 for more details.

The Cortex-M3 processor core and the memory subsystem are not clocked. Peripherals are clocked that are enabled in the **DCGCn** register when auto-clock gating is enabled (see the **RCC** register) or the **RCGCn** register when auto-clock gating is disabled. The system clock source is the main oscillator by default or the internal oscillator specified in the **DSLPCLKCFG** register if one is enabled. When the **DSLPCLKCFG** register is used, the internal oscillator is powered up, if necessary, and the main oscillator is powered down. If the PLL is running at the time of the

WFI instruction, hardware will power the PLL down and override the SYSDIV field of the active RCC/RCC2 register, to be determined by the DSDIVORIDE setting in the DSLPCLKCFG register, up to /16 or /64 respectively. When the Deep-Sleep exit event occurs, hardware brings the system clock back to the source and frequency it had at the onset of Deep-Sleep mode before enabling the clocks that had been stopped during the Deep-Sleep duration.

Caution – If the Cortex-M3 Debug Access Port (DAP) has been enabled, and the device wakes from a low power sleep or deep-sleep mode, the core may start executing code before all clocks to peripherals have been restored to their run mode configuration. The DAP is usually enabled by software tools accessing the JTAG or SWD interface when debugging or flash programming. If this condition occurs, a Hard Fault is triggered when software accesses a peripheral with an invalid clock.

A software delay loop can be used at the beginning of the interrupt routine that is used to wake up a system from a WFI (Wait For Interrupt) instruction. This stalls the execution of any code that accesses a peripheral register that might cause a fault. This loop can be removed for production software as the DAP is most likely not enabled during normal execution.

Because the DAP is disabled by default (power on reset), the user can also power-cycle the device. The DAP is not enabled unless it is enabled through the JTAG or SWD interface.

5.3 Initialization and Configuration

The PLL is configured using direct register writes to the RCC/RCC2 register. If the RCC2 register is being used, the USERCC2 bit must be set and the appropriate RCC2 bit/field is used. The steps required to successfully change the PLL-based system clock are:

- 1. Bypass the PLL and system clock divider by setting the BYPASS bit and clearing the USESYS bit in the RCC register. This configures the system to run off a "raw" clock source and allows for the new PLL configuration to be validated before switching the system clock to the PLL.
- 2. Select the crystal value (XTAL) and oscillator source (OSCSRC), and clear the PWRDN bit in RCC/RCC2. Setting the XTAL field automatically pulls valid PLL configuration data for the appropriate crystal, and clearing the PWRDN bit powers and enables the PLL and its output.
- 3. Select the desired system divider (SYSDIV) in RCC/RCC2 and set the USESYS bit in RCC. The SYSDIV field determines the system frequency for the microcontroller.
- 4. Wait for the PLL to lock by polling the PLLLRIS bit in the Raw Interrupt Status (RIS) register.
- 5. Enable use of the PLL by clearing the BYPASS bit in RCC/RCC2.

5.4 Register Map

Table 5-7 on page 181 lists the System Control registers, grouped by function. The offset listed is a hexadecimal increment to the register's address, relative to the System Control base address of 0x400F.E000.

Note: Spaces in the System Control register space that are not used are reserved for future or internal use. Software should not modify any reserved memory address.

Table 5-7. System Control Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	DID0	RO	-	Device Identification 0	182
0x004	DID1	RO	-	Device Identification 1	198
800x0	DC0	RO	0x007F.002F	Device Capabilities 0	200
0x010	DC1	RO	0x0011.31BF	Device Capabilities 1	201
0x014	DC2	RO	0x0307.1013	Device Capabilities 2	203
0x018	DC3	RO	0x8F07.87C3	Device Capabilities 3	205
0x01C	DC4	RO	0x5000.007F	Device Capabilities 4	207
0x030	PBORCTL	R/W	0x0000.7FFD	Brown-Out Reset Control	184
0x034	LDOPCTL	R/W	0x0000.0000	LDO Power Control	185
0x040	SRCR0	R/W	0x00000000	Software Reset Control 0	227
0x044	SRCR1	R/W	0x00000000	Software Reset Control 1	228
0x048	SRCR2	R/W	0x00000000	Software Reset Control 2	230
0x050	RIS	RO	0x0000.0000	Raw Interrupt Status	186
0x054	IMC	R/W	0x0000.0000	Interrupt Mask Control	187
0x058	MISC	R/W1C	0x0000.0000	Masked Interrupt Status and Clear	188
0x05C	RESC	R/W	-	Reset Cause	189
0x060	RCC	R/W	0x078E.3AD1	Run-Mode Clock Configuration	190
0x064	PLLCFG	RO	-	XTAL to PLL Translation	194
0x070	RCC2	R/W	0x0780.2810	Run-Mode Clock Configuration 2	195
0x100	RCGC0	R/W	0x00000040	Run Mode Clock Gating Control Register 0	209
0x104	RCGC1	R/W	0x00000000	Run Mode Clock Gating Control Register 1	215
0x108	RCGC2	R/W	0x00000000	Run Mode Clock Gating Control Register 2	221
0x110	SCGC0	R/W	0x00000040	Sleep Mode Clock Gating Control Register 0	211
0x114	SCGC1	R/W	0x00000000	Sleep Mode Clock Gating Control Register 1	217
0x118	SCGC2	R/W	0x00000000	Sleep Mode Clock Gating Control Register 2	223
0x120	DCGC0	R/W	0x00000040	Deep Sleep Mode Clock Gating Control Register 0	213
0x124	DCGC1	R/W	0x00000000	Deep Sleep Mode Clock Gating Control Register 1	219
0x128	DCGC2	R/W	0x00000000	Deep Sleep Mode Clock Gating Control Register 2	225
0x144	DSLPCLKCFG	R/W	0x0780.0000	Deep Sleep Clock Configuration	197

5.5 Register Descriptions

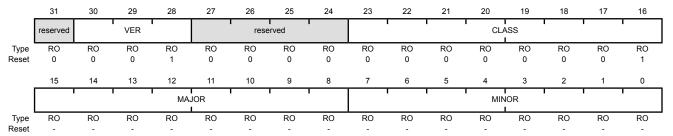
All addresses given are relative to the System Control base address of 0x400F.E000.

Register 1: Device Identification 0 (DID0), offset 0x000

This register identifies the version of the microcontroller. Each microcontroller is uniquely identified by the combined values of the CLASS field in the **DID0** register and the PARTNO field in the **DID1** register.

Device Identification 0 (DID0)

Base 0x400F.E000 Offset 0x000 Type RO, reset -



Bit/Field	Name	Type	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30:28	VER	RO	0x1	DID0 Version
				This field defines the DID0 register format version. The version number is numeric. The value of the VER field is encoded as follows:
				Value Description
				0x1 Second version of the DID0 register format.
27:24	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
23:16	CLASS	RO	0x1	Device Class

The CLASS field value identifies the internal design from which all mask sets are generated for all devices in a particular product line. The CLASS field value is changed for new product lines, for changes in fab process (for example, a remap or shrink), or any case where the MAJOR or MINOR fields require differentiation from prior devices. The value of the CLASS field is encoded as follows (all other encodings are reserved):

Value Description

0x1 Stellaris® Fury-class devices.

Bit/Field	Name	Туре	Reset	Description
15:8	MAJOR	RO	-	Major Revision This field specifies the major revision number of the device. The major revision reflects changes to base layers of the design. The major revision number is indicated in the part number as a letter (A for first revision, B for second, and so on). This field is encoded as follows:
				Value Description
				0x0 Revision A (initial device)
				0x1 Revision B (first base layer revision)
				0x2 Revision C (second base layer revision)
				and so on.
7:0	MINOR	RO	-	Minor Revision
				This field specifies the minor revision number of the device. The minor revision reflects changes to the metal layers of the design. The ${\tt MINOR}$ field value is reset when the ${\tt MAJOR}$ field is changed. This field is numeric and is encoded as follows:
				Value Description
				0x0 Initial device, or a major revision update.
				0x1 First metal layer change.
				0x2 Second metal layer change.
				and so on.

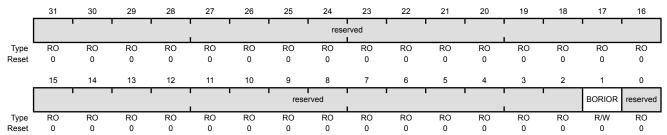
Register 2: Brown-Out Reset Control (PBORCTL), offset 0x030

This register is responsible for controlling reset conditions after initial power-on reset.

Brown-Out Reset Control (PBORCTL)

Base 0x400F.E000

Offset 0x030 Type R/W, reset 0x0000.7FFD



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	BORIOR	R/W	0	BOR Interrupt or Reset
				This bit controls how a BOR event is signaled to the controller. If set, a reset is signaled. Otherwise, an interrupt is signaled.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

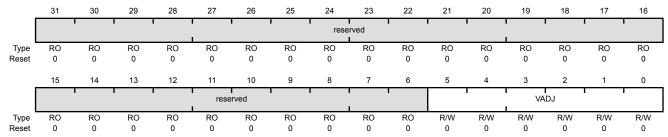
Register 3: LDO Power Control (LDOPCTL), offset 0x034

The VADJ field in this register adjusts the on-chip output voltage (V_{OUT}).

LDO Power Control (LDOPCTL)

Base 0x400F.E000 Offset 0x034

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	VADJ	R/W	0x0	LDO Output Voltage

This field sets the on-chip output voltage. The programming values for the \mathtt{VADJ} field are provided below.

Value	$V_{OUT}(V)$
0x00	2.50
0x01	2.45
0x02	2.40
0x03	2.35
0x04	2.30
0x05	2.25
0x06-0x3F	Reserved
0x1B	2.75
0x1C	2.70
0x1D	2.65
0x1E	2.60
0x1F	2.55

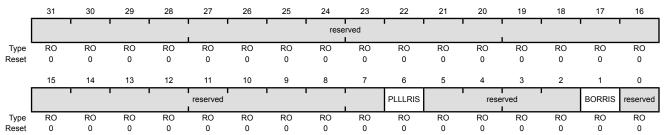
Register 4: Raw Interrupt Status (RIS), offset 0x050

Central location for system control raw interrupts. These are set and cleared by hardware.

Raw Interrupt Status (RIS)

Base 0x400F.E000 Offset 0x050

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PLLLRIS	RO	0	PLL Lock Raw Interrupt Status This bit is set when the PLL T _{READY} Timer asserts.
5:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	BORRIS	RO	0	Brown-Out Reset Raw Interrupt Status This bit is the raw interrupt status for any brown-out conditions. If set, a brown-out condition is currently active. This is an unregistered signal from the brown-out detection circuit. An interrupt is reported if the BORIM bit in the IMC register is set and the BORIOR bit in the PBORCTL register is cleared.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

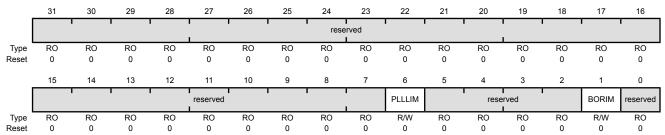
Register 5: Interrupt Mask Control (IMC), offset 0x054

Central location for system control interrupt masks.

Interrupt Mask Control (IMC)

Base 0x400F.E000

Offset 0x054 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PLLLIM	R/W	0	PLL Lock Interrupt Mask
				This bit specifies whether a PLL Lock interrupt is promoted to a controller interrupt. If set, an interrupt is generated if PLLLRIS in RIS is set; otherwise, an interrupt is not generated.
5:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	BORIM	R/W	0	Brown-Out Reset Interrupt Mask
				This bit specifies whether a brown-out condition is promoted to a controller interrupt. If set, an interrupt is generated if BORRIS is set; otherwise, an interrupt is not generated.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

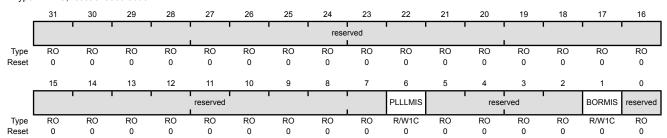
Register 6: Masked Interrupt Status and Clear (MISC), offset 0x058

On a read, this register gives the current masked status value of the corresponding interrupt. All of the bits are R/W1C and this action also clears the corresponding raw interrupt bit in the **RIS** register (see page 186).

Masked Interrupt Status and Clear (MISC)

Base 0x400F.E000

Offset 0x058
Type R/W1C, reset 0x0000.0000



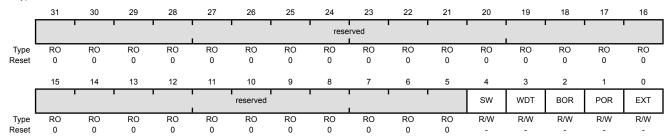
Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PLLLMIS	R/W1C	0	PLL Lock Masked Interrupt Status This bit is set when the PLL T _{READY} timer asserts. The interrupt is cleared by writing a 1 to this bit.
5:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	BORMIS	R/W1C	0	BOR Masked Interrupt Status The BORMIS is simply the BORRIS ANDed with the mask value, BORIM.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 7: Reset Cause (RESC), offset 0x05C

This register is set with the reset cause after reset. The bits in this register are sticky and maintain their state across multiple reset sequences, except when a power- on reset or an external reset is the cause, in which case, all bits other than POR or EXT in the **RESC** register are cleared.

Reset Cause (RESC)

Base 0x400F.E000 Offset 0x05C Type R/W, reset -



Bit/Field	Name	Туре	Reset	Description
31:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SW	R/W	-	Software Reset When set, indicates a software reset is the cause of the reset event.
3	WDT	R/W	-	Watchdog Timer Reset When set, indicates a watchdog reset is the cause of the reset event.
2	BOR	R/W	-	Brown-Out Reset When set, indicates a brown-out reset is the cause of the reset event.
1	POR	R/W	-	Power-On Reset When set, indicates a power-on reset is the cause of the reset event.
0	EXT	R/W	-	External Reset When set, indicates an external reset ($\overline{\tt RST}$ assertion) is the cause of the reset event.

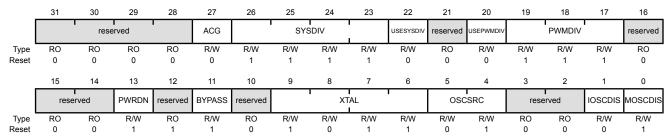
Register 8: Run-Mode Clock Configuration (RCC), offset 0x060

This register is defined to provide source control and frequency speed.

Run-Mode Clock Configuration (RCC)

Base 0x400F.E000 Offset 0x060

Type R/W, reset 0x078E.3AD1



Bit/Field	Name	Tuno	Reset	Description
Bil/Field	Name	Type	Reset	Description
31:28	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
27	ACG	R/W	0	Auto Clock Gating
				This bit specifies whether the system uses the Sleep-Mode Clock Gating Control (SCGCn) registers and Deep-Sleep-Mode Clock Gating Control (DCGCn) registers if the controller enters a Sleep or Deep-Sleep mode (respectively). If set, the SCGCn or DCGCn registers are used to control the clocks distributed to the peripherals when the controller is in a sleep mode. Otherwise, the Run-Mode Clock Gating Control (RCGCn) registers are used when the controller enters a sleep mode.
				The RCGCn registers are always used to control the clocks in Run mode.
				This allows peripherals to consume less power when the controller is in a sleep mode and the peripheral is unused.
26:23	SYSDIV	R/W	0xF	System Clock Divisor
				Specifies which divisor is used to generate the system clock from either the PLL output or the oscillator source (depending on how the BYPASS bit in this register is configured). See Table 5-5 on page 177 for bit encodings.
				If the SYSDIV value is less than MINSYSDIV (see page 201), and the PLL is being used, then the MINSYSDIV value is used as the divisor.
				If the PLL is not being used, the <code>SYSDIV</code> value can be less than <code>MINSYSDIV</code> .
22	USESYSDIV	R/W	0	Enable System Clock Divider
				Use the system clock divider as the source for the system clock. The system clock divider is forced to be used when the PLL is selected as the source.

SYSDIV field in this register.

If the USERCC2 bit in the RCC2 register is set, then the SYSDIV2 field in the RCC2 register is used as the system clock divider rather than the

Bit/Field	Name	Туре	Reset	Description
21	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
20	USEPWMDIV	R/W	0	Enable PWM Clock Divisor
				Use the PWM clock divider as the source for the PWM clock.
19:17	PWMDIV	R/W	0x7	PWM Unit Clock Divisor
				This field specifies the binary divisor used to predivide the system clock down for use as the timing reference for the PWM module. This clock is only power 2 divide and rising edge is synchronous without phase shift from the system clock.
				Value Divisor
				0x0 /2
				0x1 /4
				0x2 /8
				0x3 /16
				0x4 /32
				0x5 /64
				0x6 /64
				0x7 /64 (default)
16:14	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13	PWRDN	R/W	1	PLL Power Down
				This bit connects to the PLL PWRDN input. The reset value of 1 powers down the PLL.
12	reserved	RO	1	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11	BYPASS	R/W	1	PLL Bypass
				Chooses whether the system clock is derived from the PLL output or the OSC source. If set, the clock that drives the system is the OSC source. Otherwise, the clock that drives the system is the PLL output clock divided by the system divider.
				See Table 5-5 on page 177 for programming guidelines.
				Note: The ADC must be clocked from the PLL or directly from a 14-MHz to 18-MHz clock source to operate properly. While the ADC works in a 14-18 MHz range, to maintain a 1 M sample/second rate, the ADC must be provided a 16-MHz clock source.
10	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

July 15, 2014 191

Bit/Field	Name	Туре	Reset	Description							
9:6	XTAL	R/W	0xB	Crystal Value							
				encoding for the the PLL freque	is field is provided b	e attached to the main oscillator. The elow. Depending on the crystal used, actly 400 MHz (see Table ation).					
				Value Crystal Using tl		Not Crystal Frequency (MHz) Using the PLL					
				0x0	1.000	reserved					
				0x1	1.8432	reserved					
				0x2	2.000	reserved					
				0x3	2.4576	reserved					
				0x4	3.5	79545 MHz					
				0x5	3.	6864 MHz					
				0x6		4 MHz					
				0x7	4	.096 MHz					
				0x8	4.	9152 MHz					
				0x9		5 MHz					
				0xA	;	5.12 MHz					
				0xB	0xB 6 MHz (reset value)						
				0xC	6	6.144 MHz					
				0xD	7.	3728 MHz					
				0xE		8 MHz					
				0xF	0xF 8.192 MHz						
5:4	OSCSRC	R/W	0x1	Oscillator Sour	·ce						
				Selects the inp	out source for the OS	SC. The values are:					
				Value Input S	ource						
				0x0 MOSC							
				Main os	scillator						
				0x1 IOSC							
				Internal	l oscillator (default)						
				0x2 IOSC/4							
					l oscillator / 4						
				0x3 30 kHz							
					z internal oscillator						
				For additional of	oscillator sources, s	ee the RCC2 register.					
3:2	reserved	RO	0x0	compatibility w		lue of a reserved bit. To provide the value of a reserved bit should be rite operation.					
1	IOSCDIS	R/W	0	Internal Oscilla	tor Disable						
				0: Internal osci	llator (IOSC) is ena	bled.					
				1: Internal osci	llator is disabled.						

Bit/Field	Name	Туре	Reset	Description
0	MOSCDIS	R/W	1	Main Oscillator Disable 0: Main oscillator is enabled.
				1: Main oscillator is disabled (default).

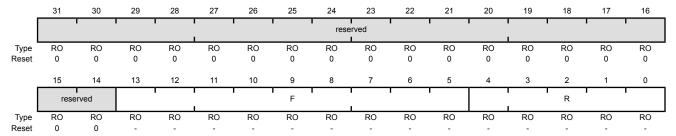
Register 9: XTAL to PLL Translation (PLLCFG), offset 0x064

This register provides a means of translating external crystal frequencies into the appropriate PLL settings. This register is initialized during the reset sequence and updated anytime that the XTAL field changes in the **Run-Mode Clock Configuration (RCC)** register (see page 190).

The PLL frequency is calculated using the PLLCFG field values, as follows:

XTAL to PLL Translation (PLLCFG)

Base 0x400F.E000 Offset 0x064 Type RO, reset -



Bit/Field	Name	Type	Reset	Description
31:14	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13:5	F	RO	-	PLL F Value This field specifies the value supplied to the PLL's F input.
4:0	R	RO	-	PLL R Value

This field specifies the value supplied to the PLL's R input.

Register 10: Run-Mode Clock Configuration 2 (RCC2), offset 0x070

This register overrides the RCC equivalent register fields, as shown in Table 5-8, when the USERCC2 bit is set, allowing the extended capabilities of the RCC2 register to be used while also providing a means to be backward-compatible to previous parts. Each RCC2 field that supersedes an RCC field is located at the same LSB bit position; however, some RCC2 fields are larger than the corresponding RCC field.

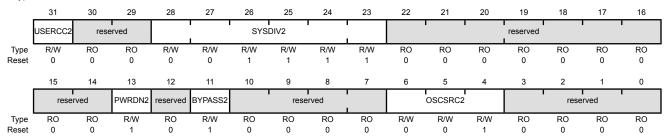
Table 5-8. RCC2 Fields that Override RCC fields

RCC2 Field	Overrides RCC Field
SYSDIV2, bits[28:23]	SYSDIV, bits[26:23]
PWRDN2, bit[13]	PWRDN, bit[13]
BYPASS2, bit[11]	BYPASS, bit[11]
oscsrc2, bits[6:4]	OSCSRC, bits[5:4]

Run-Mode Clock Configuration 2 (RCC2)

Base 0x400F.E000 Offset 0x070

Type R/W, reset 0x0780.2810



Bit/Field	Name	Туре	Reset	Description
31	USERCC2	R/W	0	Use RCC2 When set, overrides the RCC register fields.
30:29	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28:23	SYSDIV2	R/W	0x0F	System Clock Divisor
				Specifies which divisor is used to generate the system clock from either the PLL output or the oscillator source (depending on how the BYPASS2 bit is configured). SYSDIV2 is used for the divisor when both the USESYSDIV bit in the RCC register and the USERCC2 bit in this register are set. See Table 5-6 on page 177 for programming guidelines.
22:14	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13	PWRDN2	R/W	1	Power-Down PLL
				When set, powers down the PLL.
12	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description					
11	BYPASS2	R/W	1	Bypass PLL When set, bypasses the PLL for the clock source. See Table 5-6 on page 177 for programming guidelines.					
10:7	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.					
6:4	OSCSRC2	R/W	0x1	Oscillator Source					
				Selects the input source for the OSC. The values are:					
				Value Description					
				0x0 MOSC					
				Main oscillator					
				0x1 IOSC					
				Internal oscillator					
				0x2 IOSC/4					
				Internal oscillator / 4					
				0x3 30 kHz					
				30-kHz internal oscillator					
				0x4 Reserved					
				0x5 Reserved					
				0x6 Reserved					
				0x7 Reserved					
3:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.					

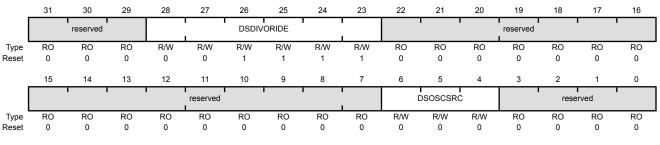
Register 11: Deep Sleep Clock Configuration (DSLPCLKCFG), offset 0x144

This register provides configuration information for the hardware control of Deep Sleep Mode.

Deep Sleep Clock Configuration (DSLPCLKCFG)

Base 0x400F.E000 Offset 0x144

Type R/W, reset 0x0780.0000



Bit/Field	Name	Туре	Reset	Description
31:29	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28:23	DSDIVORIDE	R/W	0x0F	Divider Field Override
				6-bit system divider field to override when Deep-Sleep occurs with PLL running.
22:7	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6:4	DSOSCSRC	R/W	0x0	Clock Source
				Specifies the clock source during Deep-Sleep mode.
				Value Description
				0x0 MOSC
				Use main oscillator as source.
				0x1 IOSC
				Use internal 12-MHz oscillator as source.
				0x2 Reserved
				0x3 30 kHz
				Use 30-kHz internal oscillator as source.
				0x4 Reserved
				0x5 Reserved
				0x6 Reserved
				0x7 Reserved
3:0	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 12: Device Identification 1 (DID1), offset 0x004

This register identifies the device family, part number, temperature range, pin count, and package type. Each microcontroller is uniquely identified by the combined values of the CLASS field in the **DID0** register and the PARTNO field in the **DID1** register.

23

21

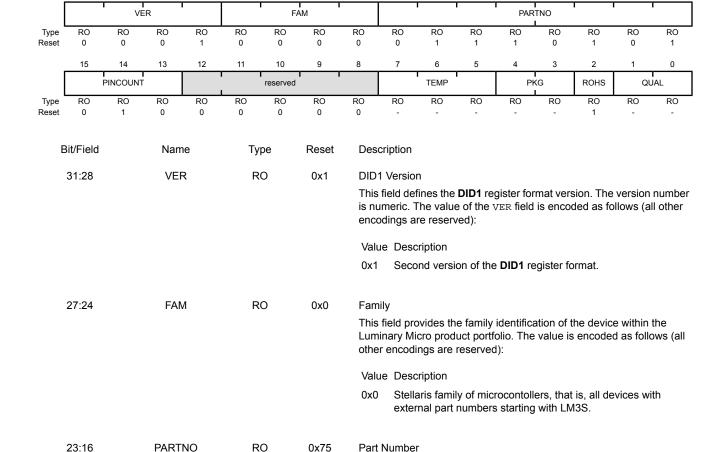
18

16

Device Identification 1 (DID1)

28

Base 0x400F.E000 Offset 0x004 Type RO, reset -



24

value is encoded as follows (all other encodings are reserved):

Value Description 0x75 LM3S6432

15:13 PINCOUNT RO 0x2 Package Pin Count

This field specifies the number of pins on the device package. The value is encoded as follows (all other encodings are reserved):

This field provides the part number of the device within the family. The

Value Description

0x2 100-pin or 108-ball package

Bit/Field	Name	Туре	Reset	Description
12:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:5	TEMP	RO	-	Temperature Range This field specifies the temperature rating of the device. The value is encoded as follows (all other encodings are reserved):
				Value Description
				0x0 Commercial temperature range (0°C to 70°C)
				0x1 Industrial temperature range (-40°C to 85°C)
				0x2 Extended temperature range (-40°C to 105°C)
4:3	PKG	RO	-	Package Type This field specifies the package type. The value is encoded as follows (all other encodings are reserved):
				Value Description 0x0 SOIC package 0x1 LQFP package 0x2 BGA package
2	ROHS	RO	1	RoHS-Compliance This bit specifies whether the device is RoHS-compliant. A 1 indicates the part is RoHS-compliant.
1:0	QUAL	RO	-	Qualification Status This field specifies the qualification status of the device. The value is encoded as follows (all other encodings are reserved):
				Value Description
				0x0 Engineering Sample (unqualified)
				0x1 Pilot Production (unqualified)
				0x2 Fully Qualified

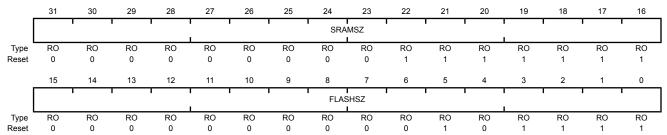
Register 13: Device Capabilities 0 (DC0), offset 0x008

This register is predefined by the part and can be used to verify features.

Device Capabilities 0 (DC0)

Base 0x400F.E000 Offset 0x008

Type RO, reset 0x007F.002F



Bit/Field	Name	Type	Reset	Description
31:16	SRAMSZ	RO	0x007F	SRAM Size Indicates the size of the on-chip SRAM memory. Value Description 0x007F 32 KB of SRAM
15:0	FLASHSZ	RO	0x002F	Flash Size

Indicates the size of the on-chip flash memory.

Value Description

0x002F 96 KB of Flash

PWM

Register 14: Device Capabilities 1 (DC1), offset 0x010

This register provides a list of features available in the system. The Stellaris family uses this register format to indicate the availability of the following family features in the specific device: CANs, PWM, ADC, Watchdog timer, Hibernation module, and debug capabilities. This register also indicates the maximum clock frequency and maximum ADC sample rate. The format of this register is consistent with the **RCGC0**, **SCGC0**, and **DCGC0** clock control registers and the **SRCR0** software reset control register.

Device Capabilities 1 (DC1)

30

Base 0x400F.E000 Offset 0x010

9:8

MAXADCSPD

RO

0x1

Type RO, reset 0x0011.31BF

Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	ſ	MINS	YSDIV		reser	ved	MAXAE	DCSPD	MPU	reserved	TEMPSNS	PLL	WDT	swo	SWD	JTAG	
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset	0	0	1	1	0	0	0	1	1	0	1	1	1	1	1	1	
_	N:4/ = : - 1 -1		N1		т		D4	D									
E	Bit/Field		Nam	ie	Тур	ре	Reset	Des	cription								
	31:21		reserv	/ed	R)	0	Soft	ware sh	ould not i	relv on th	ne value	of a rese	erved bit	. To prov	ride	
										with futu							
								pres	served a	cross a re	ead-mod	ify-write	operation	n.			
	20		PWI	M	R	1	1	DIM	M Modul	e Preser	nt.						
	20		1 441	VI	133	,	'			dicates t		2\A/\1 mc	dula ia r	rocont			
								VVIIE	en set, ii	iuicates t	nat the F	VVIVI IIIC	idule is p	nesent.			
	19:17		reserv	ved	R	С	0	Soft	ware sh	ould not i	rely on th	ne value	of a rese	erved bit	. To prov	ride	
									compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.								
								pres	served a	cross a re	ead-mod	ify-write	operation	n.			
	16		ADO	С	R	C	1	ADO	ADC Module Present								
								Whe	When set, indicates that the ADC module is present.								
	15:12		MINSY	edi/	R	2	0x3	Svo	tom Clar	sk Divido	-						
	13.12		WIINS I	SDIV	K	J	UXS	,	System Clock Divider Minimum 4-bit divider value for system clock. The reset value is								
										ependent							
										k divisor			-	, 11011 10	onango		
								\/al	ue Desc	rintion							
										•							
								0x3	S Spec	cifies a 50	J-MHz C	PU cloc	k with a	PLL divid	der of 4.		
	11:10		reserv	ved	R	С	0			ould not i							
										with futu	•				ed bit sh	ould be	
								pres	servea a	cross a re	eaa-moa	iry-write	operation	DT1.			

Max ADC Speed

Value Description

250K samples/second

Indicates the maximum rate at which the ADC samples data.

Bit/Field	Name	Туре	Reset	Description
7	MPU	RO	1	MPU Present When set, indicates that the Cortex-M3 Memory Protection Unit (MPU) module is present. See the "Cortex-M3 Peripherals" chapter in the Stellaris Data Sheet for details on the MPU.
6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	TEMPSNS	RO	1	Temp Sensor Present When set, indicates that the on-chip temperature sensor is present.
4	PLL	RO	1	PLL Present When set, indicates that the on-chip Phase Locked Loop (PLL) is present.
3	WDT	RO	1	Watchdog Timer Present When set, indicates that a watchdog timer is present.
2	SWO	RO	1	SWO Trace Port Present When set, indicates that the Serial Wire Output (SWO) trace port is present.
1	SWD	RO	1	SWD Present When set, indicates that the Serial Wire Debugger (SWD) is present.
0	JTAG	RO	1	JTAG Present When set, indicates that the JTAG debugger interface is present.

Register 15: Device Capabilities 2 (DC2), offset 0x014

This register provides a list of features available in the system. The Stellaris family uses this register format to indicate the availability of the following family features in the specific device: Analog Comparators, General-Purpose Timers, I2Cs, QEIs, SSIs, and UARTs. The format of this register is consistent with the RCGC1, SCGC1, and DCGC1 clock control registers and the SRCR1 software reset control register.

Device Capabilities 2 (DC2)

Base 0x400F.E000 Offset 0x014

31

Type RO, reset 0x0307.1013

	31	. 30	23		- 21	20	2.5		23	- 22		20	19	10			
		'	rese	rved			COMP1	COMP0			reserved			TIMER2	TIMER1	TIMER0	
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		reserved		12C0			'	reserved		•	'	SSI0	rese	rved	UART1	UART0	
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	
В	it/Field		Nam	ne	Тур	oe	Reset	Description									
31:26 reserved R0					0	0	com	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.									
25 COMP1 RO							1	Ana	loa Com	parator	1 Presen	t					
									-	•	that anal		arator 1	is prese	nt.		
												-9					
	24		COM	P0	R	C	1	Ana	log Com	parator	0 Presen	ıt					
								Whe	en set, in	dicates	that anal	og comp	arator 0	is prese	nt.		
	23:19		reserv	/ed	R	0	0	Software should not rely on the value of a reserved bit. To compatibility with future products, the value of a reserved b preserved across a read-modify-write operation.						•			
	18		TIME	R2	R)	1		er 2 Pres				_				
								Whe	en set, in	dicates	that Gen	eral-Purp	oose Tin	ner modu	ıle 2 is p	resent.	
	17		TIME	R1	R)	1	Time	er 1 Pres	sent							
								Whe	en set, in	dicates	that Gen	eral-Purp	oose Tin	ner modu	ıle 1 is p	resent.	
	16		TIME	RΛ	R	1	1	Time	er 0 Pres	ent							
	10		I IIVIL	110	100	9	•				that Gen	aral Durr	nnea Tin	ner modi	ıle ∩ ie n	recent	
								VVIIC	: 11 SCI, III	luicales	liial Gen	erai-rui į	JUSE 1111	iei iiiout	ile o is b	resent.	
	15:13		reserv	/ed	R	0	0	Software should not rely on the value of a reserved bit. To pricompatibility with future products, the value of a reserved bit preserved across a read-modify-write operation.									
	12		12C	0	R	O	1	I2C	Module	0 Preser	nt						
			•	-	• • •	-	•				that I2C i	module () is pres	ent.			
									001, 111				o p. 00	J. 16.			
	11:5		reserv	/ed	R)	0	com	patibility	with fut	rely on thure produced	ucts, the	value of	a reserv			

Bit/Field	Name	Туре	Reset	Description
4	SSI0	RO	1	SSI0 Present When set, indicates that SSI module 0 is present.
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	UART1	RO	1	UART1 Present When set, indicates that UART module 1 is present.
0	UART0	RO	1	UART0 Present When set, indicates that UART module 0 is present.

Register 16: Device Capabilities 3 (DC3), offset 0x018

This register provides a list of features available in the system. The Stellaris family uses this register format to indicate the availability of the following family features in the specific device: Analog Comparator I/Os, CCP I/Os, ADC I/Os, and PWM I/Os.

Device Capabilities 3 (DC3)

Base 0x400F.E000

17

16

15

ADC1

ADC0

PWMFAULT

RO

RO

RO

1

1

Offset 0x018 Type RO, reset 0x8F07.87C3

	32KHZ		reserved		CCP3	CCP2	CCP1	CCP0			reserved			ADC2	ADC1	ADC0
Type Reset	RO 1	RO 0	RO 0	RO 0	RO 1	RO 1	RO 1	RO 1	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 1	RO 1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWMFAULT		rese	erved		C1PLUS	C1MINUS	C0O	C0PLUS	COMINUS		rese	rved	1	PWM1	PWM0
Type Reset	RO 1	RO 0	RO 0	RO 0	RO 0	RO 1	RO 1	RO 1	RO 1	RO 1	RO 0	RO 0	RO 0	RO 0	RO 1	RO 1
E	Bit/Field		Nan	ne	Ту	ре	Reset	Des	cription							
	31		32KI	ΗZ	R	0	1	Whe	•	t Clock A dicates a t clock.		CCP pin	is prese	ent and c	an be us	sed as a
	30:28		reser	ved	R	0	0	com	patibility	ould not i with futu cross a re	ıre prodi	ucts, the	value of	a reserv	•	
	27		CCF	23	R	0	1		CCP3 Pin Present When set, indicates that Capture/Compare/PWM pin 3 is p					3 is prese	ent.	
	26		CCF	P2	R	0	1		CCP2 Pin Present When set, indicates that Capture/Compare			npare/P\	WM pin 2	2 is prese	ent.	
	25		CCF	P1	R	0	1		P1 Pin P en set, ir	resent idicates t	hat Cap	ture/Con	npare/P\	WM pin 1	I is prese	ent.
	24		CCF	P0	R	0	1		P0 Pin P en set, ir	resent idicates t	hat Cap	ture/Con	npare/P\	WM pin () is prese	ent.
	23:19		reser	ved	R	0	0	com	patibility	ould not i with futu cross a re	ıre prodı	ucts, the	value of	a reserv		
	18		ADO	2	R	.0	1		C2 Pin P en set, in	resent idicates t	hat ADC	pin 2 is	present	<u>:</u> .		

ADC1 Pin Present

ADC0 Pin Present

PWM Fault Pin Present

When set, indicates that ADC pin 1 is present.

When set, indicates that ADC pin 0 is present.

When set, indicates that the PWM Fault pin is present.

Bit/Field	Name	Туре	Reset	Description
14:11	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
10	C1PLUS	RO	1	C1+ Pin Present When set, indicates that the analog comparator 1 (+) input pin is present.
9	C1MINUS	RO	1	C1- Pin Present When set, indicates that the analog comparator 1 (-) input pin is present.
8	C0O	RO	1	C0o Pin Present When set, indicates that the analog comparator 0 output pin is present.
7	COPLUS	RO	1	C0+ Pin Present When set, indicates that the analog comparator 0 (+) input pin is present.
6	COMINUS	RO	1	C0- Pin Present When set, indicates that the analog comparator 0 (-) input pin is present.
5:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PWM1	RO	1	PWM1 Pin Present When set, indicates that the PWM pin 1 is present.
0	PWM0	RO	1	PWM0 Pin Present When set, indicates that the PWM pin 0 is present.

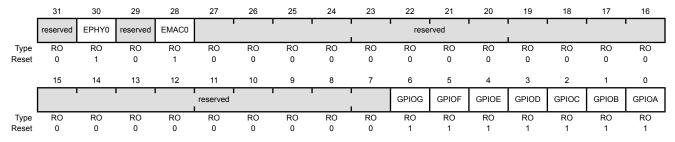
Register 17: Device Capabilities 4 (DC4), offset 0x01C

This register provides a list of features available in the system. The Stellaris family uses this register format to indicate the availability of the following family features in the specific device: Ethernet MAC and PHY, GPIOs, and CCP I/Os. The format of this register is consistent with the RCGC2, SCGC2, and DCGC2 clock control registers and the SRCR2 software reset control register.

Device Capabilities 4 (DC4)

Base 0x400F.E000

Offset 0x01C Type RO, reset 0x5000.007F



Bit/Field	Name	Туре	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30	EPHY0	RO	1	Ethernet PHY0 Present When set, indicates that Ethernet PHY module 0 is present.
29	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	EMAC0	RO	1	Ethernet MAC0 Present When set, indicates that Ethernet MAC module 0 is present.
27:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	GPIOG	RO	1	GPIO Port G Present When set, indicates that GPIO Port G is present.
5	GPIOF	RO	1	GPIO Port F Present When set, indicates that GPIO Port F is present.
4	GPIOE	RO	1	GPIO Port E Present When set, indicates that GPIO Port E is present.
3	GPIOD	RO	1	GPIO Port D Present When set, indicates that GPIO Port D is present.
2	GPIOC	RO	1	GPIO Port C Present When set, indicates that GPIO Port C is present.
1	GPIOB	RO	1	GPIO Port B Present When set, indicates that GPIO Port B is present.

Bit/Field	Name	Туре	Reset	Description
0	GPIOA	RO	1	GPIO Port A Present
				When set, indicates that GPIO Port A is present.

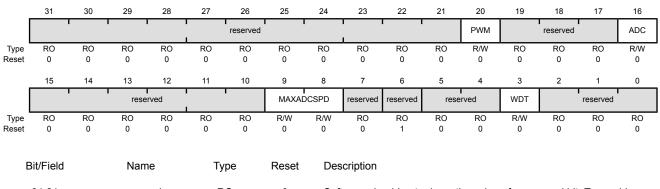
Register 18: Run Mode Clock Gating Control Register 0 (RCGC0), offset 0x100

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC0** is the clock configuration register for running operation, **SCGC0** for Sleep operation, and **DCGC0** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Run Mode Clock Gating Control Register 0 (RCGC0)

Base 0x400F.E000 Offset 0x100

Type R/W, reset 0x00000040



Bit/Field	Name	Туре	Reset	Description
31:21	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
20	PWM	R/W	0	PWM Clock Gating Control
				This bit controls the clock gating for the PWM module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates a bus fault.
19:17	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16	ADC	R/W	0	ADC0 Clock Gating Control
				This bit controls the clock gating for SAR ADC module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates a bus fault.
15:10	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
9:8	MAXADCSPD	R/W	0	ADC Sample Speed This field sets the rate at which the ADC samples data. You cannot set the rate higher than the maximum rate. You can set the sample rate by setting the MAXADCSPD bit as follows:
				Value Description
				0x1 250K samples/second
				0x0 125K samples/second
7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	reserved	RO	1	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	WDT	R/W	0	WDT Clock Gating Control This bit controls the clock gating for the WDT module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates a bus fault.
2:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 19: Sleep Mode Clock Gating Control Register 0 (SCGC0), offset 0x110

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC0** is the clock configuration register for running operation, **SCGC0** for Sleep operation, and **DCGC0** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

23

20

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be

preserved across a read-modify-write operation.

Sleep Mode Clock Gating Control Register 0 (SCGC0)

28

27

RO

0

reserved

26

25

24

Base 0x400F.E000 Offset 0x110

15:10

Type R/W, reset 0x00000040

			·		1	reserved	1 T					PWM		reserved		ADC
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 0	RO 0	RO 0	RO 0	R/W 0
iveset			-							-			ŭ			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	•		rese	rved	I		MAXAD	CSPD	reserved	reserved	rese	erved	WDT		reserved	
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 1	RO 0	RO 0	R/W 0	RO 0	RO 0	RO 0
Reset	O	U	U	U	U	U	U	U	U	'	U	U	U	U	U	O
_							_									
Е	Bit/Field		Nam	ne	Ту	pe	Reset	Des	scription							
	31:21		reserv	ved	R	0	0	con		with futu	ıre prod	ucts, the	value of	a reserv	t. To prov ved bit sh	
	20		PWI	M	R/	W	0	PW	M Clock	Gating C	ontrol					
								rece disa	eives a cl	ock and	function	s. Other	wise, the	unit is u	. If set, th unclocked unit gen	d and
	19:17		reserv	ved	R	Ο	0	con		with futu	ire prod	ucts, the	value of	a reserv	t. To prov ved bit sh	
	16		ADO	С	R/	W	0	ADO	C0 Clock	Gating (Control					
								rece	eives a cl	ock and	function	s. Other	wise, the	unit is u	0. If set, unclocked unit gen	d and

a bus fault.

Bit/Field	Name	Туре	Reset	Description
9:8	MAXADCSPD	R/W	0	ADC Sample Speed This field sets the rate at which the ADC samples data. You cannot set the rate higher than the maximum rate. You can set the sample rate by setting the MAXADCSPD bit as follows:
				Value Description
				0x1 250K samples/second
				0x0 125K samples/second
7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	reserved	RO	1	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	WDT	R/W	0	WDT Clock Gating Control This bit controls the clock gating for the WDT module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates a bus fault.
2:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 20: Deep Sleep Mode Clock Gating Control Register 0 (DCGC0), offset 0x120

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC0** is the clock configuration register for running operation, **SCGC0** for Sleep operation, and **DCGC0** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Deep Sleep Mode Clock Gating Control Register 0 (DCGC0)

Base 0x400F.E000 Offset 0x120

15:7

6

reserved

reserved

RO

RO

0

Type R/W, reset 0x00000040

30

						reserved						PWM		reserved		ADC
Type L	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					reserved	1	1 1) 	reserved	rese	rved	WDT		reserved	
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 0	RO 0	R/W 0	RO 0	RO 0	RO 0
Reset	U	U	U	U	U	O	U	U	Ü	'	U	Ü	O	U	O	U
	Bit/Field		Nam	10	Ту	ne	Reset	Dec	cription							
L	oit/i ieiu		ivaii	ic	ıy	þe	Neset	Des	Cription							
	31:21		reserv	/ed	R	0	0	com	patibility	ould not r with futu cross a re	re produ	ucts, the	value of	a reserv	•	
	20		PWI	М	R/	W	0	PW	M Clock	Gating C	ontrol					
								rece disa	eives a c	trols the o lock and the unit is	function	s. Other	wise, the	unit is เ	ınclocke	d and
	19:17		reserv	/ed	R	0	0	com	patibility	ould not r with futucross a re	re produ	ucts, the	value of	a reserv	•	
	16		ADO	С	R/	W	0	ADO	C0 Clock	Gating C	Control					

This bit controls the clock gating for SAR ADC module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be

preserved across a read-modify-write operation.

preserved across a read-modify-write operation.

a bus fault.

Bit/Field	Name	Туре	Reset	Description
5:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	WDT	R/W	0	WDT Clock Gating Control
				This bit controls the clock gating for the WDT module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, a read or write to the unit generates a bus fault.
2:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

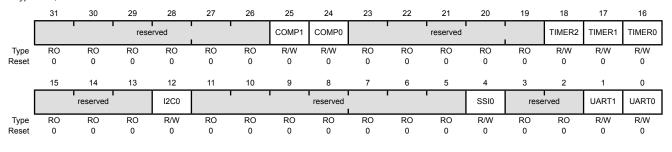
Register 21: Run Mode Clock Gating Control Register 1 (RCGC1), offset 0x104

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC1** is the clock configuration register for running operation, **SCGC1** for Sleep operation, and **DCGC1** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Run Mode Clock Gating Control Register 1 (RCGC1)

Base 0x400F.E000 Offset 0x104

Type R/W, reset 0x00000000



Bit/Field	Name	Type	Reset	Description
31:26	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
25	COMP1	R/W	0	Analog Comparator 1 Clock Gating
				This bit controls the clock gating for analog comparator 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
24	COMP0	R/W	0	Analog Comparator 0 Clock Gating
				This bit controls the clock gating for analog comparator 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
23:19	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
18	TIMER2	R/W	0	Timer 2 Clock Gating Control
				This bit controls the clock gating for General-Purpose Timer module 2.

unit will generate a bus fault.

If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the

Bit/Field	Name	Туре	Reset	Description
17	TIMER1	R/W	0	Timer 1 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
16	TIMER0	R/W	0	Timer 0 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
15:13	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	I2C0	R/W	0	I2C0 Clock Gating Control This bit controls the clock gating for I2C module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
11:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SSIO	R/W	0	SSI0 Clock Gating Control This bit controls the clock gating for SSI module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	UART1	R/W	0	UART1 Clock Gating Control This bit controls the clock gating for UART module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	UART0	R/W	0	UART0 Clock Gating Control This bit controls the clock gating for UART module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

Register 22: Sleep Mode Clock Gating Control Register 1 (SCGC1), offset 0x114

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC1** is the clock configuration register for running operation, **SCGC1** for Sleep operation, and **DCGC1** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

23

21

reserved

20

Software should not rely on the value of a reserved bit. To provide

preserved across a read-modify-write operation.

Timer 2 Clock Gating Control

unit will generate a bus fault.

compatibility with future products, the value of a reserved bit should be

This bit controls the clock gating for General-Purpose Timer module 2. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the

19

18

TIMER2

17

TIMER1

16

TIMER0

Sleep Mode Clock Gating Control Register 1 (SCGC1)

reserved

reserved

TIMER2

27

RO

R/W

0

0

26

25

COMP1

24

COMP0

Base 0x400F.E000 Offset 0x114

23:19

18

Type R/W, reset 0x00000000

30

Туре	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		reserved		I2C0				reserved	'			SSI0	rese	rved	UART1	UART0
Туре	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
В	it/Field		Nam	ne	Тур	ре	Reset	Des	cription							
	31:26		reserv	ved	R	0	0	com	patibility	with futu	ure produ	ne value ucts, the lify-write	value of	a reserv	•	
	25		COM	P1	R/	N	0	Ana	log Com	parator [*]	1 Clock (Gating				
								rece disa	ives a cl	ock and	function	ng for an s. Othen ed, reads	wise, the	· unit is ເ	ınclocke	d and
	24		СОМ	P0	R/	N	0	Ana	log Com	parator (Clock (Gating				
								rece disa	ives a cl	ock and	function	ng for an s. Othen ed, reads	wise, the	· unit is ເ	ınclocke	d and

Bit/Field	Name	Туре	Reset	Description
17	TIMER1	R/W	0	Timer 1 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
16	TIMER0	R/W	0	Timer 0 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
15:13	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	I2C0	R/W	0	I2C0 Clock Gating Control This bit controls the clock gating for I2C module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
11:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SSIO	R/W	0	SSI0 Clock Gating Control This bit controls the clock gating for SSI module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	UART1	R/W	0	UART1 Clock Gating Control This bit controls the clock gating for UART module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	UART0	R/W	0	UART0 Clock Gating Control This bit controls the clock gating for UART module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

16

Register 23: Deep Sleep Mode Clock Gating Control Register 1 (DCGC1), offset 0x124

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC1** is the clock configuration register for running operation, **SCGC1** for Sleep operation, and **DCGC1** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Deep Sleep Mode Clock Gating Control Register 1 (DCGC1)

Base 0x400F.E000 Offset 0x124

Type R/W, reset 0x00000000

_	01	- 00													17	
			rese	rved	i	1	COMP1	COMP0			reserved	1		TIMER2	TIMER1	TIMER0
Туре	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		reserved		I2C0		'	•	reserved	'		•	SSI0	rese	erved	UART1	UART0
Туре	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	N:4/ = : - 1 -1		Nissa		т.		Deset	D								
	Bit/Field		Nan	ie	Ту	pe	Reset	Des	cription							
	31:26		reser	ved	R	0	0	Soft	ware sho	ould not	rely on th	he value	of a res	erved bit	To prov	ride
	00						·				ure prodi				•	
											read-mod	-				
	25 COMP1		R/	W	0	Ana	log Com	parator	1 Clock (Gating						
											clock gati	•	•	•		
											I function					
									bled. If tr is fault.	ne unit is	unclock	ed, reads	or write	es to the i	ınıt will g	enerate
								a bu	is iauit.							
	24		COM	P0	R/	W	0	Ana	log Com	parator	0 Clock (Gating				
								This	bit contr	ols the	clock gati	ng for an	alog cor	mparator	0. If set,	the unit
											I function		,			
										ne unit is	unclock	ed, reads	or write	es to the u	unit will g	enerate
								a bu	s fault.							
	23:19		reser	/ed	R	0	0	Soft	ware sho	ould not	rely on th	he value	of a res	erved bit	. To prov	ride
											ure prodi					
								pres	erved ac	cross a i	read-mod	dify-write	operation	on.		
	18		TIME	R2	R/	W	0	Time	er 2 Cloc	k Gatin	g Control					
								This	bit contr	ols the	clock gat	ing for G	eneral-f	ourpose	Timer m	odule 2.
								If se	t, the un	it receiv	es a cloc	k and fur	nctions.	Otherwis	se, the u	nit is
								uncl	ocked ar	nd disab	led. If the	e unit is ι	unclocke	ed, reads	or write	s to the

unit will generate a bus fault.

Bit/Field	Name	Туре	Reset	Description
17	TIMER1	R/W	0	Timer 1 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
16	TIMER0	R/W	0	Timer 0 Clock Gating Control This bit controls the clock gating for General-Purpose Timer module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
15:13	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	I2C0	R/W	0	I2C0 Clock Gating Control This bit controls the clock gating for I2C module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
11:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SSIO	R/W	0	SSI0 Clock Gating Control This bit controls the clock gating for SSI module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	UART1	R/W	0	UART1 Clock Gating Control This bit controls the clock gating for UART module 1. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	UART0	R/W	0	UART0 Clock Gating Control This bit controls the clock gating for UART module 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

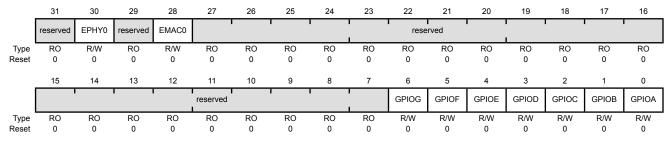
Register 24: Run Mode Clock Gating Control Register 2 (RCGC2), offset 0x108

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. RCGC2 is the clock configuration register for running operation, SCGC2 for Sleep operation, and DCGC2 for Deep-Sleep operation. Setting the ACG bit in the Run-Mode Clock Configuration (RCC) register specifies that the system uses sleep modes.

Run Mode Clock Gating Control Register 2 (RCGC2)

Base 0x400F.E000 Offset 0x108

Type R/W, reset 0x00000000



Bit/Field	Name	Туре	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30	EPHY0	R/W	0	PHY0 Clock Gating Control
				This bit controls the clock gating for Ethernet PHY unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
29	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	EMAC0	R/W	0	MAC0 Clock Gating Control
				This bit controls the clock gating for Ethernet MAC unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
27:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	GPIOG	R/W	0	Port G Clock Gating Control
				This bit controls the clock gating for Port G. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If

the unit is unclocked, reads or writes to the unit will generate a bus fault.

Bit/Field	Name	Туре	Reset	Description
5	GPIOF	R/W	0	Port F Clock Gating Control
				This bit controls the clock gating for Port F. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
4	GPIOE	R/W	0	Port E Clock Gating Control
				This bit controls the clock gating for Port E. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3	GPIOD	R/W	0	Port D Clock Gating Control
				This bit controls the clock gating for Port D. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
2	GPIOC	R/W	0	Port C Clock Gating Control
				This bit controls the clock gating for Port C. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
1	GPIOB	R/W	0	Port B Clock Gating Control
				This bit controls the clock gating for Port B. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	GPIOA	R/W	0	Port A Clock Gating Control
				This bit controls the clock gating for Port A. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

Register 25: Sleep Mode Clock Gating Control Register 2 (SCGC2), offset 0x118

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC2** is the clock configuration register for running operation, **SCGC2** for Sleep operation, and **DCGC2** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Sleep Mode Clock Gating Control Register 2 (SCGC2)

Base 0x400F.E000 Offset 0x118 Type R/W, reset 0x00000000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved	EPHY0	reserved	EMAC0	ľ					rese	rved					
Type	RO	R/W	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1		1		reserved	1		i		GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30	EPHY0	R/W	0	PHY0 Clock Gating Control This bit controls the clock gating for Ethernet PHY unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
29	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	EMAC0	R/W	0	MAC0 Clock Gating Control This bit controls the clock gating for Ethernet MAC unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
27:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	GPIOG	R/W	0	Port G Clock Gating Control This bit controls the clock gating for Port G. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

Bit/Field	Name	Туре	Reset	Description
5	GPIOF	R/W	0	Port F Clock Gating Control
				This bit controls the clock gating for Port F. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
4	GPIOE	R/W	0	Port E Clock Gating Control
				This bit controls the clock gating for Port E. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3	GPIOD	R/W	0	Port D Clock Gating Control
				This bit controls the clock gating for Port D. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
2	GPIOC	R/W	0	Port C Clock Gating Control
				This bit controls the clock gating for Port C. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
1	GPIOB	R/W	0	Port B Clock Gating Control
				This bit controls the clock gating for Port B. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	GPIOA	R/W	0	Port A Clock Gating Control
				This bit controls the clock gating for Port A. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

Register 26: Deep Sleep Mode Clock Gating Control Register 2 (DCGC2), offset 0x128

This register controls the clock gating logic. Each bit controls a clock enable for a given interface, function, or unit. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled (saving power). If the unit is unclocked, reads or writes to the unit will generate a bus fault. The reset state of these bits is 0 (unclocked) unless otherwise noted, so that all functional units are disabled. It is the responsibility of software to enable the ports necessary for the application. Note that these registers may contain more bits than there are interfaces, functions, or units to control. This is to assure reasonable code compatibility with other family and future parts. **RCGC2** is the clock configuration register for running operation, **SCGC2** for Sleep operation, and **DCGC2** for Deep-Sleep operation. Setting the ACG bit in the **Run-Mode Clock Configuration (RCC)** register specifies that the system uses sleep modes.

Deep Sleep Mode Clock Gating Control Register 2 (DCGC2)

Base 0x400F.E000 Offset 0x128 Type R/W, reset 0x00000000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved	EPHY0	reserved	EMAC0			1	1	1	rese	rved	1		1		
Type	RO	R/W	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			1		reserved		1	'	1	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30	EPHY0	R/W	0	PHY0 Clock Gating Control This bit controls the clock gating for Ethernet PHY unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
29	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	EMAC0	R/W	0	MAC0 Clock Gating Control This bit controls the clock gating for Ethernet MAC unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
27:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	GPIOG	R/W	0	Port G Clock Gating Control This bit controls the clock gating for Port G. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

Bit/Field	Name	Туре	Reset	Description
5	GPIOF	R/W	0	Port F Clock Gating Control
				This bit controls the clock gating for Port F. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
4	GPIOE	R/W	0	Port E Clock Gating Control
				This bit controls the clock gating for Port E. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
3	GPIOD	R/W	0	Port D Clock Gating Control
				This bit controls the clock gating for Port D. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
2	GPIOC	R/W	0	Port C Clock Gating Control
				This bit controls the clock gating for Port C. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
1	GPIOB	R/W	0	Port B Clock Gating Control
				This bit controls the clock gating for Port B. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.
0	GPIOA	R/W	0	Port A Clock Gating Control
				This bit controls the clock gating for Port A. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled. If the unit is unclocked, reads or writes to the unit will generate a bus fault.

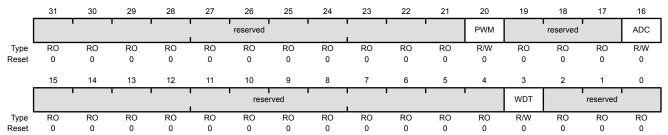
Register 27: Software Reset Control 0 (SRCR0), offset 0x040

Writes to this register are masked by the bits in the Device Capabilities 1 (DC1) register.

Software Reset Control 0 (SRCR0)

Base 0x400F.E000 Offset 0x040

Offset 0x040 Type R/W, reset 0x00000000



Bit/Field	Name	Туре	Reset	Description
31:21	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
20	PWM	R/W	0	PWM Reset Control
				Reset control for PWM module.
19:17	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16	ADC	R/W	0	ADC0 Reset Control
				Reset control for SAR ADC module 0.
15:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	WDT	R/W	0	WDT Reset Control
				Reset control for Watchdog unit.
2:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 28: Software Reset Control 1 (SRCR1), offset 0x044

Writes to this register are masked by the bits in the **Device Capabilities 2 (DC2)** register.

Software Reset Control 1 (SRCR1)

Base 0x400F.E000

Offset 0x044
Type R/W, reset 0x00000000

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ſ			rese	rved	1		COMP1	COMP0			reserved	1	1	TIMER2	TIMER1	TIMER0
Type	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		reserved		12C0			1	reserved				SSI0	rese	erved	UART1	UART0
Type •	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	,	0	Ü	
Bit/Field	Name	Туре	Reset	Description
31:26	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
25	COMP1	R/W	0	Analog Comp 1 Reset Control Reset control for analog comparator 1.
24	COMP0	R/W	0	Analog Comp 0 Reset Control Reset control for analog comparator 0.
23:19	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
18	TIMER2	R/W	0	Timer 2 Reset Control Reset control for General-Purpose Timer module 2.
17	TIMER1	R/W	0	Timer 1 Reset Control Reset control for General-Purpose Timer module 1.
16	TIMER0	R/W	0	Timer 0 Reset Control Reset control for General-Purpose Timer module 0.
15:13	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	I2C0	R/W	0	I2C0 Reset Control Reset control for I2C unit 0.
11:5	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	SSI0	R/W	0	SSI0 Reset Control Reset control for SSI unit 0.

Bit/Field	Name	Туре	Reset	Description
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	UART1	R/W	0	UART1 Reset Control Reset control for UART unit 1.
0	UART0	R/W	0	UART0 Reset Control Reset control for UART unit 0.

Register 29: Software Reset Control 2 (SRCR2), offset 0x048

Writes to this register are masked by the bits in the **Device Capabilities 4 (DC4)** register.

Software Reset Control 2 (SRCR2)

Base 0x400F.E000

Offset 0x048
Type R/W, reset 0x00000000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved	EPHY0	reserved	EMAC0			1	1		rese	rved					
Type	RO	R/W	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					reserved		1	1		GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W						
Reset	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ	Λ

		ŭ ŭ	· ·	
Bit/Field	Name	Туре	Reset	Description
31	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
30	EPHY0	R/W	0	PHY0 Reset Control Reset control for Ethernet PHY unit 0.
29	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
28	EMAC0	R/W	0	MAC0 Reset Control
				Reset control for Ethernet MAC unit 0.
27:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	GPIOG	R/W	0	Port G Reset Control Reset control for GPIO Port G.
5	GPIOF	R/W	0	Port F Reset Control Reset control for GPIO Port F.
4	GPIOE	R/W	0	Port E Reset Control Reset control for GPIO Port E.
3	GPIOD	R/W	0	Port D Reset Control Reset control for GPIO Port D.
2	GPIOC	R/W	0	Port C Reset Control Reset control for GPIO Port C.
1	GPIOB	R/W	0	Port B Reset Control Reset control for GPIO Port B.
0	GPIOA	R/W	0	Port A Reset Control Reset control for GPIO Port A.

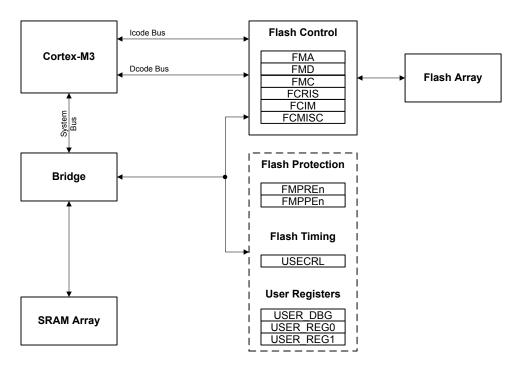
6 Internal Memory

The LM3S6432 microcontroller comes with 32 KB of bit-banded SRAM and 96 KB of flash memory. The flash controller provides a user-friendly interface, making flash programming a simple task. Flash protection can be applied to the flash memory on a 2-KB block basis.

6.1 Block Diagram

Figure 6-1 on page 231 illustrates the Flash functions. The dashed boxes in the figure indicate registers residing in the System Control module rather than the Flash Control module.

Figure 6-1. Flash Block Diagram



6.2 Functional Description

This section describes the functionality of the SRAM and Flash memories.

6.2.1 SRAM Memory

The internal SRAM of the Stellaris[®] devices is located at address 0x2000.0000 of the device memory map. To reduce the number of time consuming read-modify-write (RMW) operations, ARM has introduced *bit-banding* technology in the Cortex-M3 processor. With a bit-band-enabled processor, certain regions in the memory map (SRAM and peripheral space) can use address aliases to access individual bits in a single, atomic operation.

The bit-band alias is calculated by using the formula:

```
bit-band alias = bit-band base + (byte offset * 32) + (bit number * 4)
```

For example, if bit 3 at address 0x2000.1000 is to be modified, the bit-band alias is calculated as:

```
0x2200.0000 + (0x1000 * 32) + (3 * 4) = 0x2202.000C
```

With the alias address calculated, an instruction performing a read/write to address 0x2202.000C allows direct access to only bit 3 of the byte at address 0x2000.1000.

For details about bit-banding, see "Bit-Banding" on page 72.

6.2.2 Flash Memory

The flash is organized as a set of 1-KB blocks that can be individually erased. Erasing a block causes the entire contents of the block to be reset to all 1s. An individual 32-bit word can be programmed to change bits that are currently 1 to a 0. These blocks are paired into a set of 2-KB blocks that can be individually protected. The protection allows blocks to be marked as read-only or execute-only, providing different levels of code protection. Read-only blocks cannot be erased or programmed, protecting the contents of those blocks from being modified. Execute-only blocks cannot be erased or programmed, and can only be read by the controller instruction fetch mechanism, protecting the contents of those blocks from being read by either the controller or by a debugger.

See also "Serial Flash Loader" on page 661 for a preprogrammed flash-resident utility used to download code to the flash memory of a device without the use of a debug interface.

6.2.2.1 Flash Memory Timing

The timing for the flash is automatically handled by the flash controller. However, in order to do so, it must know the clock rate of the system in order to time its internal signals properly. The number of clock cycles per microsecond must be provided to the flash controller for it to accomplish this timing. It is software's responsibility to keep the flash controller updated with this information via the **USec Reload (USECRL)** register.

On reset, the **USECRL** register is loaded with a value that configures the flash timing so that it works with the maximum clock rate of the part. If software changes the system operating frequency, the new operating frequency minus 1 (in MHz) must be loaded into **USECRL** before any flash modifications are attempted. For example, if the device is operating at a speed of 20 MHz, a value of 0x13 (20-1) must be written to the **USECRL** register.

6.2.2.2 Flash Memory Protection

The user is provided two forms of flash protection per 2-KB flash blocks in two pairs of 32-bit wide registers. The protection policy for each form is controlled by individual bits (per policy per block) in the **FMPPEn** and **FMPREn** registers.

- Flash Memory Protection Program Enable (FMPPEn): If set, the block may be programmed (written) or erased. If cleared, the block may not be changed.
- Flash Memory Protection Read Enable (FMPREn): If a bit is set, the corresponding block may be executed or read by software or debuggers. If a bit is cleared, the corresponding block may only be executed, and contents of the memory block are prohibited from being read as data.

The policies may be combined as shown in Table 6-1 on page 232.

Table 6-1. Flash Protection Policy Combinations

FMPPEn	FMPREn	Protection
0	0	Execute-only protection. The block may only be executed and may not be written or erased.
		This mode is used to protect code.

Table 6-1. Flash Protection Policy Combinations (continued)

FMPPEn	FMPREn	Protection
1	0	The block may be written, erased or executed, but not read. This combination is unlikely to be used.
0	1	Read-only protection. The block may be read or executed but may not be written or erased. This mode is used to lock the block from further modification while allowing any read or execute access.
1	1	No protection. The block may be written, erased, executed or read.

A Flash memory access that attempts to read a read-protected block (**FMPREn** bit is set) is prohibited and generates a bus fault. A Flash memory access that attempts to program or erase a program-protected block (**FMPPEn** bit is set) is prohibited and can optionally generate an interrupt (by setting the AMASK bit in the **Flash Controller Interrupt Mask (FCIM)** register) to alert software developers of poorly behaving software during the development and debug phases.

The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. These settings create a policy of open access and programmability. The register bits may be changed by clearing the specific register bit. The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. The changes are committed using the **Flash Memory Control (FMC)** register. Details on programming these bits are discussed in "Nonvolatile Register Programming" on page 235.

6.2.2.3 Execute-Only Protection

Execute-only protection prevents both modification and visibility to a protected flash block. This mode is intended to be used in situations where a device requires debug capability, yet portions of the application space must be protected from external access. An example of this is a company who wishes to sell Stellaris devices with their proprietary software pre-programmed, yet allow the end user to add custom code to an unprotected region of the flash (such as a motor control module with a customizable motor configuration section in flash).

Literal data introduces a complication to the protection mechanism. When C code is compiled and linked, literal data (constants, and so on) is typically placed in the text section, between functions, by the compiler. The literal data is accessed at run time through the use of the LDR instruction, which loads the data from memory using a PC-relative memory address. The execution of the LDR instruction generates a read transaction across the Cortex-M3's DCode bus, which is subject to the execute-only protection mechanism. If the accessed block is marked as execute only, the transaction is blocked, and the processor is prevented from loading the constant data and, therefore, inhibiting correct execution. Therefore, using execute-only protection requires that literal data be handled differently. There are three ways to address this:

- 1. Use a compiler that allows literal data to be collected into a separate section that is put into one or more read-enabled flash blocks. Note that the LDR instruction may use a PC-relative address—in which case the literal pool cannot be located outside the span of the offset—or the software may reserve a register to point to the base address of the literal pool and the LDR offset is relative to the beginning of the pool.
- 2. Use a compiler that generates literal data from arithmetic instruction immediate data and subsequent computation.
- 3. Use method 1 or 2, but in assembly language, if the compiler does not support either method.

6.2.2.4 Read-Only Protection

Read-only protection prevents the contents of the flash block from being re-programmed, while still allowing the content to be read by processor or the debug interface. Note that if a **FMPREn** bit is cleared, all read accesses to the Flash memory block are disallowed, including any data accesses. Care must be taken not to store required data in a Flash memory block that has the associated **FMPREn** bit cleared.

The read-only mode does not prevent read access to the stored program, but it does provide protection against accidental (or malicious) erasure or programming. Read-only is especially useful for utilities like the boot loader when the debug interface is permanently disabled. In such combinations, the boot loader, which provides access control to the Flash memory, is protected from being erased or modified.

6.2.2.5 Permanently Disabling Debug

For extremely sensitive applications, the debug interface to the processor and peripherals can be permanently disabled, blocking all accesses to the device through the JTAG or SWD interfaces. With the debug interface disabled, it is still possible to perform standard IEEE instructions (such as boundary scan operations), but access to the processor and peripherals is blocked.

The DBG0 and DBG1 bits of the **User Debug (USER_DBG)** register control whether the debug interface is turned on or off.

The debug interface should not be permanently disabled without providing some mechanism—such as the boot loader—to provide customer-installable updates or bug fixes. Disabling the debug interface is permanent and cannot be reversed.

6.2.2.6 Interrupts

The Flash memory controller can generate interrupts when the following conditions are observed:

- Programming Interrupt signals when a program or erase action is complete.
- Access Interrupt signals when a program or erase action has been attempted on a 2-kB block of memory that is protected by its corresponding FMPPEn bit.

The interrupt events that can trigger a controller-level interrupt are defined in the **Flash Controller Masked Interrupt Status (FCMIS)** register (see page 243) by setting the corresponding MASK bits. If interrupts are not used, the raw interrupt status is always visible via the **Flash Controller Raw Interrupt Status (FCRIS)** register (see page 242).

Interrupts are always cleared (for both the **FCMIS** and **FCRIS** registers) by writing a 1 to the corresponding bit in the **Flash Controller Masked Interrupt Status and Clear (FCMISC)** register (see page 244).

6.3 Flash Memory Initialization and Configuration

6.3.1 Flash Programming

The Stellaris devices provide a user-friendly interface for flash programming. All erase/program operations are handled via three registers: **FMA**, **FMD**, and **FMC**.

During a Flash memory operation (write, page erase, or mass erase) access to the Flash memory is inhibited. As a result, instruction and literal fetches are held off until the Flash memory operation is complete. If instruction execution is required during a Flash memory operation, the code that is executing must be placed in SRAM and executed from there while the flash operation is in progress.

6.3.1.1 To program a 32-bit word

- 1. Write source data to the **FMD** register.
- 2. Write the target address to the **FMA** register.
- 3. Write the flash write key and the WRITE bit (a value of 0xA442.0001) to the FMC register.
- 4. Poll the FMC register until the WRITE bit is cleared.

6.3.1.2 To perform an erase of a 1-KB page

- 1. Write the page address to the **FMA** register.
- 2. Write the flash write key and the ERASE bit (a value of 0xA442.0002) to the FMC register.
- 3. Poll the FMC register until the ERASE bit is cleared.

6.3.1.3 To perform a mass erase of the flash

- 1. Write the flash write key and the MERASE bit (a value of 0xA442.0004) to the FMC register.
- 2. Poll the FMC register until the MERASE bit is cleared.

6.3.2 Nonvolatile Register Programming

Note: The USER DBG register requires a POR before the committed changes take effect.

This section discusses how to update registers that are resident within the Flash memory itself. These registers exist in a separate space from the main Flash memory array and are not affected by an ERASE or MASS ERASE operation. The bits in these registers can be changed from 1 to 0 with a write operation. Prior to being committed, the register contents are unaffected by any reset condition except power-on reset, which returns the register contents to the original value. By committing the register values using the COMT bit in the **FMC** register, the register contents become nonvolatile and are therefore retained following power cycling. Once the register contents are committed, the contents are permanent, and they cannot be restored to their factory default values.

With the exception of the **USER_DBG** register, the settings in these registers can be tested before committing them to Flash memory. For the **USER_DBG** register, the data to be written is loaded into the **FMD** register before it is committed. The **FMD** register is read only and does not allow the **USER_DBG** operation to be tried before committing it to nonvolatile memory.

Important: The Flash memory registers can only have bits changed from 1 to 0 by user programming and can only be committed once. After being committed, these registers cannot be restored to their factory default values.

In addition, the USER_REG0, USER_REG1, USER_REG2, USER_REG3, and USER_DBG registers each use bit 31 (NW) to indicate that they have not been committed and bits in the register may be changed from 1 to 0. These five registers can only be committed once whereas the Flash memory protection registers may be committed multiple times. Table 6-2 on page 236 provides the FMA address required for commitment of each of the registers and the source of the data to be written when the FMC register is written with a value of 0xA442.0008. After writing the COMT bit, the user may poll the FMC register to wait for the commit operation to complete.

Table 6-2. User-Programmable Flash Memory Resident Registers

Register to be Committed	FMA Value	Data Source
FMPRE0	0x0000.0000	FMPRE0
FMPRE1	0x0000.0002	FMPRE1
FMPPE0	0x0000.0001	FMPPE0
FMPPE1	0x0000.0003	FMPPE1
USER_REG0	0x8000.0000	USER_REG0
USER_REG1	0x8000.0001	USER_REG1
USER_REG2	0x8000.0002	USER_REG2
USER_REG3	0x8000.0003	USER_REG3
USER_DBG	0x7510.0000	FMD

6.4 Register Map

Table 6-3 on page 236 lists the Flash memory and control registers. The offset listed is a hexadecimal increment to the register's address. The **FMA**, **FMD**, **FMC**, **FCRIS**, **FCIM**, and **FCMISC** register offsets are relative to the Flash memory control base address of 0x400F.D000. The Flash memory protection register offsets are relative to the System Control base address of 0x400F.E000.

Table 6-3. Flash Register Map

Offset	Name	Туре	Reset	Description	See page			
Flash Memory Control Registers (Flash Control Offset)								
0x000	FMA	R/W	0x0000.0000	Flash Memory Address	238			
0x004	FMD	R/W	0x0000.0000	Flash Memory Data	239			
0x008	FMC	R/W	0x0000.0000	Flash Memory Control	240			
0x00C	FCRIS	RO	0x0000.0000	Flash Controller Raw Interrupt Status	242			
0x010	FCIM	R/W	0x0000.0000	Flash Controller Interrupt Mask	243			
0x014	FCMISC	R/W1C	0x0000.0000	Flash Controller Masked Interrupt Status and Clear	244			
Flash Memory Protection Registers (System Control Offset)								
0x130	FMPRE0	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 0	247			
0x200	FMPRE0	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 0	247			
0x134	FMPPE0	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 0	248			
0x400	FMPPE0	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 0	248			
0x140	USECRL	R/W	0x31	USec Reload	246			
0x1D0	USER_DBG	R/W	0xFFFF.FFFE	User Debug	249			
0x1E0	USER_REG0	R/W	0xFFFF.FFFF	User Register 0	250			
0x1E4	USER_REG1	R/W	0xFFFF.FFFF	User Register 1	251			
0x204	FMPRE1	R/W	0x0000.FFFF	Flash Memory Protection Read Enable 1	252			
0x208	FMPRE2	R/W	0x0000.0000	Flash Memory Protection Read Enable 2	253			

Table 6-3. Flash Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0x20C	FMPRE3	R/W	0x0000.0000	Flash Memory Protection Read Enable 3	254
0x404	FMPPE1	R/W	0x0000.FFFF	Flash Memory Protection Program Enable 1	255
0x408	FMPPE2	R/W	0x0000.0000	Flash Memory Protection Program Enable 2	256
0x40C	FMPPE3	R/W	0x0000.0000	Flash Memory Protection Program Enable 3	257

6.5 Flash Register Descriptions (Flash Control Offset)

This section lists and describes the Flash Memory registers, in numerical order by address offset. Registers in this section are relative to the Flash control base address of 0x400F.D000.

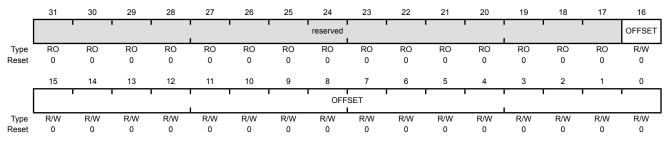
Register 1: Flash Memory Address (FMA), offset 0x000

During a write operation, this register contains a 4-byte-aligned address and specifies where the data is written. During erase operations, this register contains a 1 KB-aligned address and specifies which page is erased. Note that the alignment requirements must be met by software or the results of the operation are unpredictable.

Flash Memory Address (FMA)

Base 0x400F.D000

Offset 0x000 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:17	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16:0	OFFSET	R/W	0x0	Address Offset

Address offset in flash where operation is performed, except for nonvolatile registers (see "Nonvolatile Register Programming" on page 235 for details on values for this field).

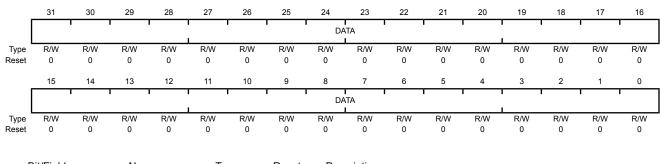
Register 2: Flash Memory Data (FMD), offset 0x004

This register contains the data to be written during the programming cycle or read during the read cycle. Note that the contents of this register are undefined for a read access of an execute-only block. This register is not used during the erase cycles.

Flash Memory Data (FMD)

Base 0x400F.D000

Offset 0x004 Type R/W, reset 0x0000.0000



Bit/Field Name Type Reset Description
31:0 DATA R/W 0x0 Data Value

Data value for write operation.

Register 3: Flash Memory Control (FMC), offset 0x008

When this register is written, the flash controller initiates the appropriate access cycle for the location specified by the **Flash Memory Address (FMA)** register (see page 238). If the access is a write access, the data contained in the **Flash Memory Data (FMD)** register (see page 239) is written.

This is the final register written and initiates the memory operation. There are four control bits in the lower byte of this register that, when set, initiate the memory operation. The most used of these register bits are the ERASE and WRITE bits.

It is a programming error to write multiple control bits and the results of such an operation are unpredictable.

Flash Memory Control (FMC)

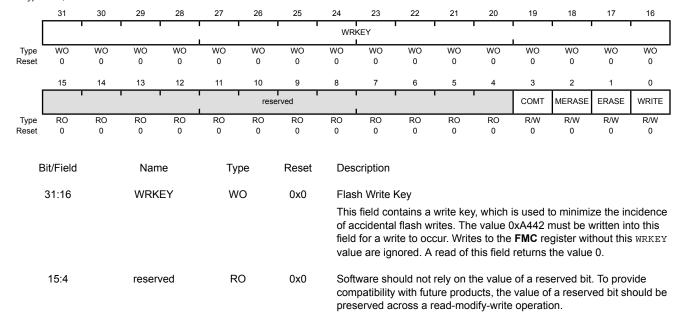
Base 0x400F.D000 Offset 0x008

3

2

COMT

Type R/W, reset 0x0000.0000



Commit Register Value

Commit (write) of register value to nonvolatile storage. A write of 0 has no effect on the state of this bit.

If read, the state of the previous commit access is provided. If the previous commit access is complete, a 0 is returned; otherwise, if the commit access is not complete, a 1 is returned.

This can take up to 50 µs.

MERASE R/W 0 Mass Erase Flash Memory

0

R/W

If this bit is set, the flash main memory of the device is all erased. A write of 0 has no effect on the state of this bit.

If read, the state of the previous mass erase access is provided. If the previous mass erase access is complete, a 0 is returned; otherwise, if the previous mass erase access is not complete, a 1 is returned.

This can take up to 250 ms.

Bit/Field	Name	Туре	Reset	Description
1	ERASE	R/W	0	Erase a Page of Flash Memory
				If this bit is set, the page of flash main memory as specified by the contents of FMA is erased. A write of 0 has no effect on the state of this bit.
				If read, the state of the previous erase access is provided. If the previous erase access is complete, a 0 is returned; otherwise, if the previous erase access is not complete, a 1 is returned.
				This can take up to 25 ms.
0	WRITE	R/W	0	Write a Word into Flash Memory
				If this bit is set, the data stored in FMD is written into the location as specified by the contents of FMA . A write of 0 has no effect on the state of this bit.
				If read, the state of the previous write update is provided. If the previous write access is complete, a 0 is returned; otherwise, if the write access is not complete, a 1 is returned.
				This can take up to 50 μs.

Register 4: Flash Controller Raw Interrupt Status (FCRIS), offset 0x00C

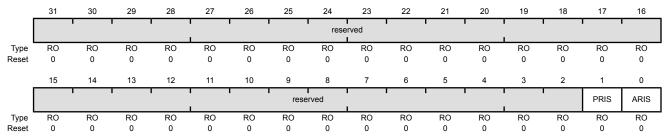
This register indicates that the flash controller has an interrupt condition. An interrupt is only signaled if the corresponding FCIM register bit is set.

Flash Controller Raw Interrupt Status (FCRIS)

Base 0x400F.D000

Dit/Fiold

Offset 0x00C Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PRIS	RO	0	Programming Raw Interrupt Status

This bit provides status on programming cycles which are write or erase actions generated through the FMC register bits (see page 240).

Value Description

- 1 The programming cycle has completed.
- 0 The programming cycle has not completed.

This status is sent to the interrupt controller when the PMASK bit in the FCIM register is set.

This bit is cleared by writing a 1 to the PMISC bit in the FCMISC register.

0 **ARIS** RO 0 Access Raw Interrupt Status

Value Description

- A program or erase action was attempted on a block of Flash memory that contradicts the protection policy for that block as set in the FMPPEn registers.
- No access has tried to improperly program or erase the Flash 0 memory.

This status is sent to the interrupt controller when the AMASK bit in the FCIM register is set.

This bit is cleared by writing a 1 to the AMISC bit in the FCMISC register.

Register 5: Flash Controller Interrupt Mask (FCIM), offset 0x010

This register controls whether the flash controller generates interrupts to the controller.

Flash Controller Interrupt Mask (FCIM)

Base 0x400F.D000 Offset 0x010

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PMASK	R/W	0	Programming Interrupt Mask
				This bit controls the reporting of the programming raw interrupt status to the interrupt controller.
				Value Description
				1 An interrupt is sent to the interrupt controller when the PRIS bit is set.
				O The PRIS interrupt is suppressed and not sent to the interrupt controller.
0	AMASK	R/W	0	Access Interrupt Mask

Access Interrupt Mask

This bit controls the reporting of the access raw interrupt status to the interrupt controller.

Value Description

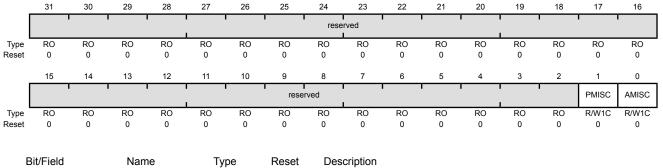
- An interrupt is sent to the interrupt controller when the ARIS bit
- 0 The ARIS interrupt is suppressed and not sent to the interrupt controller.

Register 6: Flash Controller Masked Interrupt Status and Clear (FCMISC), offset 0x014

This register provides two functions. First, it reports the cause of an interrupt by indicating which interrupt source or sources are signalling the interrupt. Second, it serves as the method to clear the interrupt reporting.

Flash Controller Masked Interrupt Status and Clear (FCMISC)

Base 0x400F.D000 Offset 0x014
Type R/W1C, reset 0x0000.0000



		.) -		
31:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PMISC	R/W1C	0	Programming Masked Interrupt Status and Clear

Value Description

- 1 When read, a 1 indicates that an unmasked interrupt was signaled because a programming cycle completed.
 - Writing a 1 to this bit clears PMISC and also the PRIS bit in the FCRIS register (see page 242).
- When read, a 0 indicates that a programming cycle complete 0 interrupt has not occurred.

A write of 0 has no effect on the state of this bit.

0	AMISC	R/W1C	0	Access Mask	ed Interrupt	Status and C	Clear
---	-------	-------	---	-------------	--------------	--------------	-------

Value Description

- When read, a 1 indicates that an unmasked interrupt was signaled because a program or erase action was attempted on a block of Flash memory that contradicts the protection policy for that block as set in the FMPPEn registers.
 - Writing a 1 to this bit clears AMISC and also the ARIS bit in the FCRIS register (see page 242).
- 0 When read, a 0 indicates that no improper accesses have occurred.

A write of 0 has no effect on the state of this bit.

6.6 Flash Register Descriptions (System Control Offset)

The remainder of this section lists and describes the Flash Memory registers, in numerical order by address offset. Registers in this section are relative to the System Control base address of 0x400F.E000.

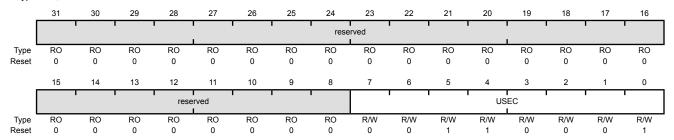
Register 7: USec Reload (USECRL), offset 0x140

Note: Offset is relative to System Control base address of 0x400F.E000

This register is provided as a means of creating a 1-µs tick divider reload value for the flash controller. The internal flash has specific minimum and maximum requirements on the length of time the high voltage write pulse can be applied. It is required that this register contain the operating frequency (in MHz -1) whenever the flash is being erased or programmed. The user is required to change this value if the clocking conditions are changed for a flash erase/program operation.

USec Reload (USECRL)

Base 0x400F.E000 Offset 0x140 Type R/W, reset 0x31



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	USEC	R/W	0x31	Microsecond Reload Value

MHz -1 of the controller clock when the flash is being erased or programmed.

If the maximum system frequency is being used, USEC should be set to 0x31 (50 MHz) whenever the flash is being erased or programmed.

Register 8: Flash Memory Protection Read Enable 0 (FMPRE0), offset 0x130 and 0x200

Note: This register is aliased for backwards compatability.

Note: Offset is relative to System Control base address of 0x400FE000.

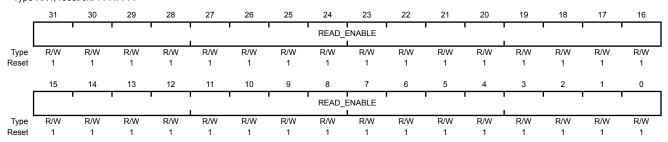
This register stores the read-only protection bits for each 2-KB flash block (**FMPPEn** stores the execute-only bits). Flash memory up to a total of 64 KB is controlled by this register. Other **FMPREn** registers (if any) provide protection for other 64K blocks. This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Read Enable 0 (FMPRE0)

READ_ENABLE

Base 0x400F.E000 Offset 0x130 and 0x200 Type R/W, reset 0xFFFF.FFFF

31:0



0xFFFFFFF

Bit/Field Name Type Reset Description

R/W

Flash Read Enable. Enables 2-KB Flash memory blocks to be executed or read. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0xFFFFFFF Bits [31:0] each enable protection on a 2-KB block of Flash memory up to the total of 64 KB.

Register 9: Flash Memory Protection Program Enable 0 (FMPPE0), offset 0x134 and 0x400

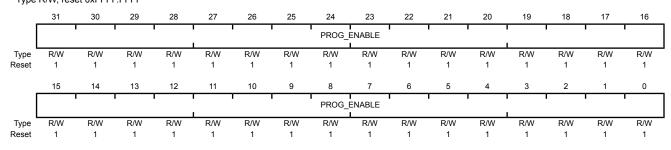
Note: This register is aliased for backwards compatability.

Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the execute-only protection bits for each 2-KB flash block (**FMPREn** stores the execute-only bits). Flash memory up to a total of 64 KB is controlled by this register. Other **FMPPEn** registers (if any) provide protection for other 64K blocks. This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Program Enable 0 (FMPPE0)

Base 0x400F.E000 Offset 0x134 and 0x400 Type R/W, reset 0xFFFF.FFFF



Bit/Field Name Type Reset Description

31:0 PROG_ENABLE R/W 0xFFFFFFF Flash Programming Enable

Configures 2-KB flash blocks to be execute only. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0xFFFFFFF Bits [31:0] each enable protection on a 2-KB block of Flash memory up to the total of 64 KB.

Register 10: User Debug (USER_DBG), offset 0x1D0

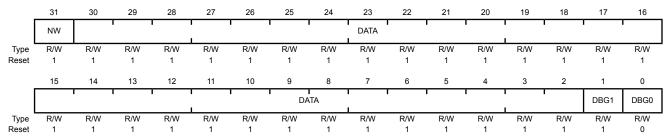
Note: Offset is relative to System Control base address of 0x400FE000.

This register provides a write-once mechanism to disable external debugger access to the device in addition to 27 additional bits of user-defined data. The DBG0 bit (bit 0) is set to 0 from the factory and the DBG1 bit (bit 1) is set to 1, which enables external debuggers. Changing the DBG1 bit to 0 disables any external debugger access to the device permanently, starting with the next power-up cycle of the device. The NW bit (bit 31) indicates that the register has not yet been committed and is controlled through hardware to ensure that the register is only committed once. Prior to being committed, bits can only be changed from 1 to 0. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. Once committed, this register cannot be restored to the factory default value.

User Debug (USER_DBG)

Base 0x400F.E000 Offset 0x1D0

Type R/W, reset 0xFFFF.FFFE



Bit/Field	Name	Туре	Reset	Description
31	NW	R/W	1	User Debug Not Written When set, this bit indicates that this 32-bit register has not been committed. When clear, this bit specifies that this register has been committed and may not be committed again.
30:2	DATA	R/W	0x1FFFFFFF	User Data Contains the user data value. This field is initialized to all 1s and can only be committed once.
1	DBG1	R/W	1	Debug Control 1 The DBG1 bit must be 1 and DBG0 must be 0 for debug to be available.
0	DBG0	R/W	0	Debug Control 0 The DBG1 bit must be 1 and DBG0 must be 0 for debug to be available.

Register 11: User Register 0 (USER_REG0), offset 0x1E0

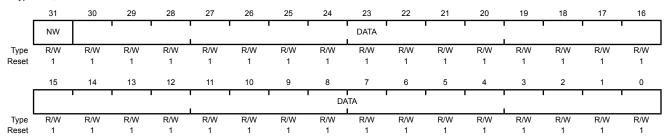
Note: Offset is relative to System Control base address of 0x400FE000.

This register provides 31 bits of user-defined data that is non-volatile and can only be committed once. Bit 31 indicates that the register is available to be committed and is controlled through hardware to ensure that the register is only committed once. Prior to being committed, bits can only be changed from 1 to 0. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. The write-once characteristics of this register are useful for keeping static information like communication addresses that need to be unique per part and would otherwise require an external EEPROM or other non-volatile device. Once committed, this register cannot be restored to the factory default value.

User Register 0 (USER REG0)

Base 0x400F.E000 Offset 0x1E0

Type R/W, reset 0xFFFF.FFFF



Bit/Field	Name	Type	Reset	Description
31	NW	R/W	1	Not Written When set, this bit indicates that this 32-bit register has not been committed. When clear, this bit specifies that this register has been committed and may not be committed again.
30:0	DATA	R/W 0x	7FFFFFF	User Data

Contains the user data value. This field is initialized to all 1s and can only be committed once.

Register 12: User Register 1 (USER_REG1), offset 0x1E4

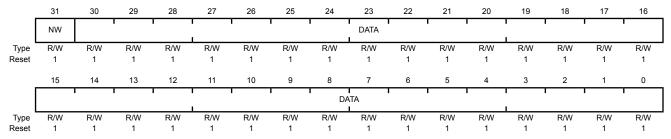
Note: Offset is relative to System Control base address of 0x400FE000.

This register provides 31 bits of user-defined data that is non-volatile and can only be committed once. Bit 31 indicates that the register is available to be committed and is controlled through hardware to ensure that the register is only committed once. Prior to being committed, bits can only be changed from 1 to 0. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. The write-once characteristics of this register are useful for keeping static information like communication addresses that need to be unique per part and would otherwise require an external EEPROM or other non-volatile device. Once committed, this register cannot be restored to the factory default value.

User Register 1 (USER REG1)

Base 0x400F.E000 Offset 0x1E4

Type R/W, reset 0xFFFF.FFFF



Bit/Field	Name	Туре	Reset	Description
31	NW	R/W	1	Not Written
				When set, this bit indicates that this 32-bit register has not been committed. When clear, this bit specifies that this register has been committed and may not be committed again.
30:0	DATA	R/M/ Ox	7FFFFFF	User Data

Contains the user data value. This field is initialized to all 1s and can only be committed once.

Register 13: Flash Memory Protection Read Enable 1 (FMPRE1), offset 0x204

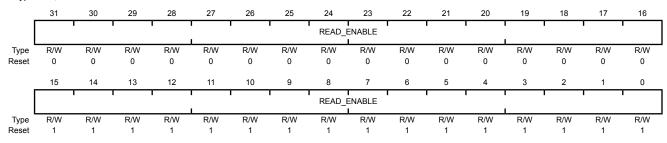
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the read-only protection bits for each 2-KB flash block (**FMPPEn** stores the execute-only bits). Flash memory up to a total of 64 KB is controlled by this register. Other **FMPREn** registers (if any) provide protection for other 64K blocks. This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. If the Flash memory size on the device is less than 64 KB, this register usually reads as zeroes, but software should not rely on these bits to be zero. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Read Enable 1 (FMPRE1)

Base 0x400F.E000 Offset 0x204

Type R/W, reset 0x0000.FFFF



Description

Bit/Field Name Type Reset

31:0 READ ENABLE R/W 0x0000FFFF

Flash Read Enable. Enables 2-KB Flash memory blocks to be executed or read. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0x0000FFFF Bits [31:0] each enable protection on a 2-KB block of Flash memory in memory range from 65 to 128 KB.

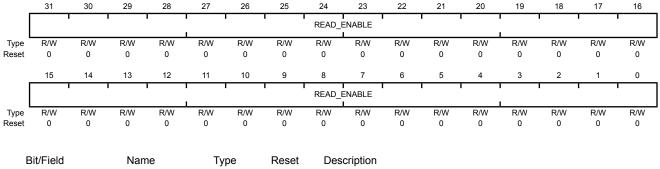
Register 14: Flash Memory Protection Read Enable 2 (FMPRE2), offset 0x208

Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the read-only protection bits for each 2-KB flash block (**FMPPEn** stores the execute-only bits). This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Read Enable 2 (FMPRE2)

Base 0x400F.E000 Offset 0x208 Type R/W, reset 0x0000.0000



31:0 READ_ENABLE R/W 0x00000000 Flash Read Enable

Enables 2-KB flash blocks to be executed or read. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0x00000000 Enables 96 KB of flash.

Register 15: Flash Memory Protection Read Enable 3 (FMPRE3), offset 0x20C

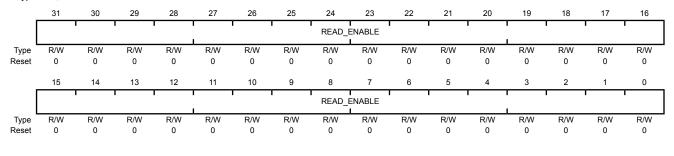
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the read-only protection bits for each 2-KB flash block (**FMPPEn** stores the execute-only bits). This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Read Enable 3 (FMPRE3)

Base 0x400F.E000 Offset 0x20C

Type R/W, reset 0x0000.0000



Bit/Field Name Type Reset Description

31:0 READ_ENABLE R/W 0x00000000 Flash Read Enable

Enables 2-KB flash blocks to be executed or read. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0x00000000 Enables 96 KB of flash.

Register 16: Flash Memory Protection Program Enable 1 (FMPPE1), offset 0x404

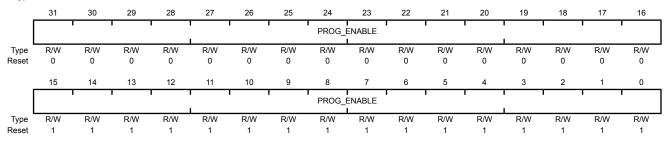
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the execute-only protection bits for each 2-KB flash block (**FMPREn** stores the execute-only bits). Flash memory up to a total of 64 KB is controlled by this register. Other **FMPPEn** registers (if any) provide protection for other 64K blocks. This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. The reset value shown only applies to power-on reset; any other type of reset does not affect this register. If the Flash memory size on the device is less than 64 KB, this register usually reads as zeroes, but software should not rely on these bits to be zero. For additional information, see the "Flash Memory Protection" section.



Base 0x400F.E000 Offset 0x404

Type R/W, reset 0x0000.FFFF



Bit/Field Name Type Reset Description

31:0 PROG_ENABLE R/W 0x0000FFFF Flash Programming Enable

Value Description

0x0000FFFF Bits [31:0] each enable protection on a 2-KB block of Flash memory in memory range from 65 to 128 KB.

Register 17: Flash Memory Protection Program Enable 2 (FMPPE2), offset 0x408

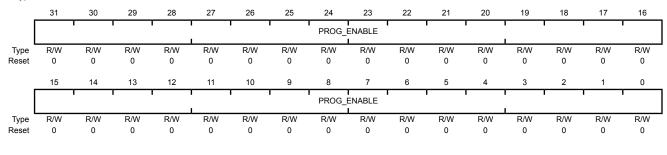
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the execute-only protection bits for each 2-KB flash block (**FMPREn** stores the execute-only bits). This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Program Enable 2 (FMPPE2)

Base 0x400F.E000 Offset 0x408

Type R/W, reset 0x0000.0000



Bit/Field Name Type Reset Description

31:0 PROG_ENABLE R/W 0x00000000 Flash Programming Enable

Configures 2-KB flash blocks to be execute only. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description

0x000000000 Enables 96 KB of flash.

Register 18: Flash Memory Protection Program Enable 3 (FMPPE3), offset 0x40C

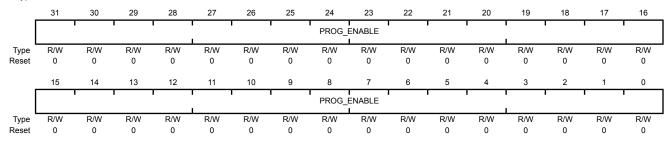
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the execute-only protection bits for each 2-KB flash block (**FMPREn** stores the execute-only bits). This register is loaded during the power-on reset sequence. The factory settings for the **FMPREn** and **FMPPEn** registers are a value of 1 for all implemented banks. This achieves a policy of open access and programmability. The register bits may be changed by writing the specific register bit. However, this register is R/W0; the user can only change the protection bit from a 1 to a 0 (and may NOT change a 0 to a 1). The changes are not permanent until the register is committed (saved), at which point the bit change is permanent. If a bit is changed from a 1 to a 0 and not committed, it may be restored by executing a power-on reset sequence. For additional information, see the "Flash Memory Protection" section.

Flash Memory Protection Program Enable 3 (FMPPE3)

Base 0x400F.E000 Offset 0x40C

Type R/W, reset 0x0000.0000



Bit/Field Name Type Reset Description

31:0 PROG_ENABLE R/W 0x00000000 Flash Programming Enable

Configures 2-KB flash blocks to be execute only. The policies may be combined as shown in the table "Flash Protection Policy Combinations".

Value Description
0x00000000 Enables 96 KB of flash.

7 General-Purpose Input/Outputs (GPIOs)

The GPIO module is composed of seven physical GPIO blocks, each corresponding to an individual GPIO port (Port A, Port B, Port C, Port D, Port E, Port F, Port G). The GPIO module supports 14-43 programmable input/output pins, depending on the peripherals being used.

The GPIO module has the following features:

- 14-43 GPIOs, depending on configuration
- 5-V-tolerant in input configuration
- Fast toggle capable of a change every two clock cycles
- Programmable control for GPIO interrupts
 - Interrupt generation masking
 - Edge-triggered on rising, falling, or both
 - Level-sensitive on High or Low values
- Bit masking in both read and write operations through address lines
- Can initiate an ADC sample sequence
- Pins configured as digital inputs are Schmitt-triggered.
- Programmable control for GPIO pad configuration
 - Weak pull-up or pull-down resistors
 - 2-mA, 4-mA, and 8-mA pad drive for digital communication; up to four pads can be configured with an 18-mA pad drive for high-current applications
 - Slew rate control for the 8-mA drive
 - Open drain enables
 - Digital input enables

7.1 Signal Description

GPIO signals have alternate hardware functions. Table 7-4 on page 261 and Table 7-5 on page 262 list the GPIO pins and the analog and digital alternate functions. The AINx analog signals are not 5-V tolerant and go through an isolation circuit before reaching their circuitry. These signals are configured by clearing the corresponding DEN bit in the GPIO Digital Enable (GPIODEN) register. Other analog signals are 5-V tolerant and are connected directly to their circuitry (CO-, CO+, C1-, C1+). These signals are configured by clearing the DEN bit in the GPIO Digital Enable (GPIODEN) register. The digital alternate hardware functions are enabled by setting the appropriate bit in the GPIO Alternate Function Select (GPIOAFSEL) and GPIODEN registers and configuring the PMCx bit field in the GPIO Port Control (GPIOPCTL) register to the numeric enoding shown in the table below. Note that each pin must be programmed individually; no type of grouping is implied by the columns in the table.

Important: All GPIO pins are configured as GPIOs and tri-stated by default (GPIOAFSEL=0, GPIODEN=0, GPIOPDR=0, GPIOPUR=0, and GPIOPCTL=0, with the exception of the four JTAG/SWD pins (shown in the table below). A Power-On-Reset (POR) or asserting RST puts the pins back to their default state.

Table 7-1. GPIO Pins With Non-Zero Reset Values

GPIO Pins	Default State	GPIOAFSEL	GPIODEN	GPIOPDR	GPIOPUR	GPIOPCTL
PA[1:0]	UART0	1	1	0	0	0x1
PA[5:2]	SSI0	1	1	0	0	0x1
PB[3:2]	I ² C0	1	1	0	0	0x1
PC[3:0]	JTAG/SWD	1	1	0	1	0x3

Table 7-2. GPIO Pins and Alternate Functions (100LQFP)

10	Pin Number	Multiplexed Function	Multiplexed Function
PA0	26	U0Rx	
PA1	27	UOTx	
PA2	28	SSIOClk	
PA3	29	SSI0Fss	
PA4	30	SSIORx	
PA5	31	SSIOTX	
PA6	34	CCP1	
PA7	35		
PB0	66	CCP0	
PB1	67	CCP2	
PB2	70	I2C0SCL	
PB3	71	I2C0SDA	
PB4	92	C0-	
PB5	91	C1-	
PB6	90	C0+	
PB7	89	TRST	
PC0	80	TCK	SWCLK
PC1	79	TMS	SWDIO
PC2	78	TDI	
PC3	77	TDO	SWO
PC4	25		
PC5	24	C1+	
PC6	23	CCP3	
PC7	22		
PD0	10	PWM0	
PD1	11	PWM1	
PD2	12	U1Rx	
PD3	13	U1Tx	
PD4	95		

Table 7-2. GPIO Pins and Alternate Functions (100LQFP) (continued)

10	Pin Number	Multiplexed Function	Multiplexed Function
PD5	96		
PD6	99	Fault	
PD7	100	C0o	
PE0	72		
PE1	73		
PE2	74		
PE3	75		
PE4	6		
PF0	47		
PF1	61		
PF2	60	LED1	
PF3	59	LED0	
PG0	19		
PG1	18		

Table 7-3. GPIO Pins and Alternate Functions (108BGA)

10	Pin Number	Multiplexed Function	Multiplexed Function
PA0	L3	U0Rx	
PA1	M3	UOTx	
PA2	M4	SSI0Clk	
PA3	L4	SSI0Fss	
PA4	L5	SSI0Rx	
PA5	M5	SSIOTx	
PA6	L6	CCP1	
PA7	M6		
PB0	E12	CCP0	
PB1	D12	CCP2	
PB2	C11	I2C0SCL	
PB3	C12	I2C0SDA	
PB4	A6	C0-	
PB5	B7	C1-	
PB6	A7	C0+	
PB7	A8	TRST	
PC0	A9	TCK	SWCLK
PC1	В9	TMS	SWDIO
PC2	B8	TDI	
PC3	A10	TDO	SWO
PC4	L1		
PC5	M1	C1+	
PC6	M2	CCP3	
PC7	L2		

Table 7-3. GPIO Pins and Alternate Functions (108BGA) (continued)

10	Pin Number	Multiplexed Function	Multiplexed Function
PD0	G1	PWM0	
PD1	G2	PWM1	
PD2	H2	U1Rx	
PD3	H1	UlTx	
PD4	E1		
PD5	E2		
PD6	F2	Fault	
PD7	F1	C0o	
PE0	A11		
PE1	B12		
PE2	B11		
PE3	A12		
PE4	D1		
PF0	M9		
PF1	H12		
PF2	J11	LED1	
PF3	J12	LED0	
PG0	K1		
PG1	K2		

Table 7-4. GPIO Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
PA0	26	I/O	TTL	GPIO port A bit 0.
PA1	27	I/O	TTL	GPIO port A bit 1.
PA2	28	I/O	TTL	GPIO port A bit 2.
PA3	29	I/O	TTL	GPIO port A bit 3.
PA4	30	I/O	TTL	GPIO port A bit 4.
PA5	31	I/O	TTL	GPIO port A bit 5.
PA6	34	I/O	TTL	GPIO port A bit 6.
PA7	35	I/O	TTL	GPIO port A bit 7.
PB0	66	I/O	TTL	GPIO port B bit 0.
PB1	67	I/O	TTL	GPIO port B bit 1.
PB2	70	I/O	TTL	GPIO port B bit 2.
PB3	71	I/O	TTL	GPIO port B bit 3.
PB4	92	I/O	TTL	GPIO port B bit 4.
PB5	91	I/O	TTL	GPIO port B bit 5.
PB6	90	I/O	TTL	GPIO port B bit 6.
PB7	89	I/O	TTL	GPIO port B bit 7.
PC0	80	I/O	TTL	GPIO port C bit 0.
PC1	79	I/O	TTL	GPIO port C bit 1.
PC2	78	I/O	TTL	GPIO port C bit 2.

Table 7-4. GPIO Signals (100LQFP) (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
PC3	77	I/O	TTL	GPIO port C bit 3.
PC4	25	I/O	TTL	GPIO port C bit 4.
PC5	24	I/O	TTL	GPIO port C bit 5.
PC6	23	I/O	TTL	GPIO port C bit 6.
PC7	22	I/O	TTL	GPIO port C bit 7.
PD0	10	I/O	TTL	GPIO port D bit 0.
PD1	11	I/O	TTL	GPIO port D bit 1.
PD2	12	I/O	TTL	GPIO port D bit 2.
PD3	13	I/O	TTL	GPIO port D bit 3.
PD4	95	I/O	TTL	GPIO port D bit 4.
PD5	96	I/O	TTL	GPIO port D bit 5.
PD6	99	I/O	TTL	GPIO port D bit 6.
PD7	100	I/O	TTL	GPIO port D bit 7.
PE0	72	I/O	TTL	GPIO port E bit 0.
PE1	73	I/O	TTL	GPIO port E bit 1.
PE2	74	I/O	TTL	GPIO port E bit 2.
PE3	75	I/O	TTL	GPIO port E bit 3.
PE4	6	I/O	TTL	GPIO port E bit 4.
PF0	47	I/O	TTL	GPIO port F bit 0.
PF1	61	I/O	TTL	GPIO port F bit 1.
PF2	60	I/O	TTL	GPIO port F bit 2.
PF3	59	I/O	TTL	GPIO port F bit 3.
PG0	19	I/O	TTL	GPIO port G bit 0.
PG1	18	I/O	TTL	GPIO port G bit 1.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 7-5. GPIO Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
PA0	L3	I/O	TTL	GPIO port A bit 0.
PA1	M3	I/O	TTL	GPIO port A bit 1.
PA2	M4	I/O	TTL	GPIO port A bit 2.
PA3	L4	I/O	TTL	GPIO port A bit 3.
PA4	L5	I/O	TTL	GPIO port A bit 4.
PA5	M5	I/O	TTL	GPIO port A bit 5.
PA6	L6	I/O	TTL	GPIO port A bit 6.
PA7	M6	I/O	TTL	GPIO port A bit 7.
PB0	E12	I/O	TTL	GPIO port B bit 0.
PB1	D12	I/O	TTL	GPIO port B bit 1.
PB2	C11	I/O	TTL	GPIO port B bit 2.
PB3	C12	I/O	TTL	GPIO port B bit 3.
PB4	A6	I/O	TTL	GPIO port B bit 4.

Table 7-5. GPIO Signals (108BGA) (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
PB5	B7	I/O	TTL	GPIO port B bit 5.
PB6	A7	I/O	TTL	GPIO port B bit 6.
PB7	A8	I/O	TTL	GPIO port B bit 7.
PC0	A9	I/O	TTL	GPIO port C bit 0.
PC1	B9	I/O	TTL	GPIO port C bit 1.
PC2	В8	I/O	TTL	GPIO port C bit 2.
PC3	A10	I/O	TTL	GPIO port C bit 3.
PC4	L1	I/O	TTL	GPIO port C bit 4.
PC5	M1	I/O	TTL	GPIO port C bit 5.
PC6	M2	I/O	TTL	GPIO port C bit 6.
PC7	L2	I/O	TTL	GPIO port C bit 7.
PD0	G1	I/O	TTL	GPIO port D bit 0.
PD1	G2	I/O	TTL	GPIO port D bit 1.
PD2	H2	I/O	TTL	GPIO port D bit 2.
PD3	H1	I/O	TTL	GPIO port D bit 3.
PD4	E1	I/O	TTL	GPIO port D bit 4.
PD5	E2	I/O	TTL	GPIO port D bit 5.
PD6	F2	I/O	TTL	GPIO port D bit 6.
PD7	F1	I/O	TTL	GPIO port D bit 7.
PE0	A11	I/O	TTL	GPIO port E bit 0.
PE1	B12	I/O	TTL	GPIO port E bit 1.
PE2	B11	I/O	TTL	GPIO port E bit 2.
PE3	A12	I/O	TTL	GPIO port E bit 3.
PE4	D1	I/O	TTL	GPIO port E bit 4.
PF0	M9	I/O	TTL	GPIO port F bit 0.
PF1	H12	I/O	TTL	GPIO port F bit 1.
PF2	J11	I/O	TTL	GPIO port F bit 2.
PF3	J12	I/O	TTL	GPIO port F bit 3.
PG0	K1	I/O	TTL	GPIO port G bit 0.
PG1	K2	I/O	TTL	GPIO port G bit 1.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

7.2 Functional Description

Important: All GPIO pins are tri-stated by default (GPIOAFSEL=0, GPIODEN=0, GPIOPDR=0, and GPIOPUR=0), with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). The JTAG/SWD pins default to their JTAG/SWD functionality (GPIOAFSEL=1, GPIODEN=1 and GPIOPUR=1). A Power-On-Reset (POR) or asserting RST puts both groups of pins back to their default state.

While debugging systems where PB7 is being used as a GPIO, care must be taken to ensure that a low value is not applied to the pin when the part is reset. Because PB7 reverts to the $\overline{\texttt{TRST}}$ function after reset, a Low value on the pin causes the JTAG controller to be reset, resulting in a loss of JTAG communication.

Each GPIO port is a separate hardware instantiation of the same physical block (see Figure 7-1 on page 264). The LM3S6432 microcontroller contains seven ports and thus seven of these physical GPIO blocks.

Commit Mode Control Control GPIOLOCK GPIOAFSEL **GPIOCR** Alternate Input DEMUX Alternate Output Alternate Output Enable Digital Pad Output S Package I/O Pin **GPIO** Input Data I/O Pad Control GPIO Output **GPIODATA** Pad Output Enable GPIO Output Enable **GPIODIR** Interrupt Pad Control Control GPIODR2R **GPIOIS** Interrupt **GPIOIBE** GPIODR4R **GPIOIEV** GPIODR8R **GPIOSLR GPIOIM GPIOPUR GPIORIS GPIOPDR GPIOMIS GPIOODR GPIOICR GPIODEN Identification Registers** GPIOPeriphID4 GPIOPeriphID0 GPIOPCellID0 GPIOPeriphID1 GPIOPeriphID5 GPIOPCellID1 GPIOPeriphID2 GPIOPeriphID6 GPIOPCellID2 GPIOPeriphID3 GPIOPeriphID7 GPIOPCellID3

Figure 7-1. GPIO Port Block Diagram

7.2.1 Data Control

The data control registers allow software to configure the operational modes of the GPIOs. The data direction register configures the GPIO as an input or an output while the data register either captures incoming data or drives it out to the pads.

7.2.1.1 Data Direction Operation

The **GPIO Direction (GPIODIR)** register (see page 272) is used to configure each individual pin as an input or output. When the data direction bit is set to 0, the GPIO is configured as an input and the corresponding data register bit will capture and store the value on the GPIO port. When the data direction bit is set to 1, the GPIO is configured as an output and the corresponding data register bit will be driven out on the GPIO port.

7.2.1.2 Data Register Operation

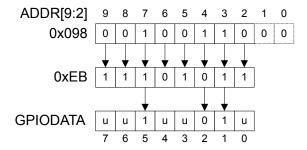
To aid in the efficiency of software, the GPIO ports allow for the modification of individual bits in the GPIO Data (GPIODATA) register (see page 271) by using bits [9:2] of the address bus as a mask.

This allows software drivers to modify individual GPIO pins in a single instruction, without affecting the state of the other pins. This is in contrast to the "typical" method of doing a read-modify-write operation to set or clear an individual GPIO pin. To accommodate this feature, the **GPIODATA** register covers 256 locations in the memory map.

During a write, if the address bit associated with that data bit is set to 1, the value of the **GPIODATA** register is altered. If it is cleared to 0, it is left unchanged.

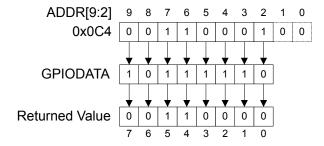
For example, writing a value of 0xEB to the address GPIODATA + 0x098 would yield as shown in Figure 7-2 on page 265, where u is data unchanged by the write.

Figure 7-2. GPIODATA Write Example



During a read, if the address bit associated with the data bit is set to 1, the value is read. If the address bit associated with the data bit is set to 0, it is read as a zero, regardless of its actual value. For example, reading address GPIODATA + 0x0C4 yields as shown in Figure 7-3 on page 265.

Figure 7-3. GPIODATA Read Example



7.2.2 Interrupt Control

The interrupt capabilities of each GPIO port are controlled by a set of seven registers. With these registers, it is possible to select the source of the interrupt, its polarity, and the edge properties. When one or more GPIO inputs cause an interrupt, a single interrupt output is sent to the interrupt controller for the entire GPIO port. For edge-triggered interrupts, software must clear the interrupt to enable any further interrupts. For a level-sensitive interrupt, it is assumed that the external source holds the level constant for the interrupt to be recognized by the controller.

Three registers are required to define the edge or sense that causes interrupts:

- GPIO Interrupt Sense (GPIOIS) register (see page 273)
- GPIO Interrupt Both Edges (GPIOIBE) register (see page 274)
- GPIO Interrupt Event (GPIOIEV) register (see page 275)

Interrupts are enabled/disabled via the GPIO Interrupt Mask (GPIOIM) register (see page 276).

When an interrupt condition occurs, the state of the interrupt signal can be viewed in two locations: the **GPIO Raw Interrupt Status (GPIORIS)** and **GPIO Masked Interrupt Status (GPIOMIS)** registers (see page 277 and page 278). As the name implies, the **GPIOMIS** register only shows interrupt conditions that are allowed to be passed to the controller. The **GPIORIS** register indicates that a GPIO pin meets the conditions for an interrupt, but has not necessarily been sent to the controller.

In addition to providing GPIO functionality, PB4 can also be used as an external trigger for the ADC. If PB4 is configured as a non-masked interrupt pin (the appropriate bit of GPIOIM is set to 1), not only is an interrupt for PortB generated, but an external trigger signal is sent to the ADC. If the **ADC Event Multiplexer Select (ADCEMUX)** register is configured to use the external trigger, an ADC conversion is initiated.

If no other PortB pins are being used to generate interrupts, the **Interrupt 0-31 Set Enable (EN0)** register can disable the PortB interrupts, and the ADC interrupt can be used to read back the converted data. Otherwise, the PortB interrupt handler needs to ignore and clear interrupts on PB4, and wait for the ADC interrupt or the ADC interrupt must be disabled in the **EN0** register and the PortB interrupt handler must poll the ADC registers until the conversion is completed. See page 106 for more information.

Interrupts are cleared by writing a 1 to the appropriate bit of the **GPIO Interrupt Clear (GPIOICR)** register (see page 279).

When programming the following interrupt control registers, the interrupts should be masked (**GPIOIM** set to 0). Writing any value to an interrupt control register (**GPIOIS**, **GPIOIBE**, or **GPIOIEV**) can generate a spurious interrupt if the corresponding bits are enabled.

7.2.3 Mode Control

The GPIO pins can be controlled by either hardware or software. When hardware control is enabled via the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 280), the pin state is controlled by its alternate function (that is, the peripheral). Software control corresponds to GPIO mode, where the **GPIODATA** register is used to read/write the corresponding pins.

7.2.4 Commit Control

The GPIO commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Protection is currently provided for the five JTAG/SWD pins (PB7 and PC[3:0]). Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 280) are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register (see page 290) has been unlocked and the appropriate bits of the **GPIO Commit (GPIOCR)** register (see page 291) have been set to 1.

7.2.5 Pad Control

The pad control registers allow for GPIO pad configuration by software based on the application requirements. The pad control registers include the GPIODR2R, GPIODR4R, GPIODR8R, GPIODDR, GPIOPDR, GPIOPDR, GPIOPDR, and GPIODEN registers. These registers control drive strength, open-drain configuration, pull-up and pull-down resistors, slew-rate control and digital enable.

For special high-current applications, the GPIO output buffers may be used with the following restrictions. With the GPIO pins configured as 8-mA output drivers, a total of four GPIO outputs may be used to sink current loads up to 18 mA each. At 18-mA sink current loading, the V_{OL} value is specified as 1.2 V. The high-current GPIO package pins must be selected such that there are only a maximum of two per side of the physical package or BGA pin group with the total number of high-current GPIO outputs not exceeding four for the entire package.

7.2.6 Identification

The identification registers configured at reset allow software to detect and identify the module as a GPIO block. The identification registers include the **GPIOPeriphID0-GPIOPeriphID7** registers as well as the **GPIOPCeIIID0-GPIOPCeIIID3** registers.

7.3 Initialization and Configuration

To use the GPIO, the peripheral clock must be enabled by setting the appropriate GPIO Port bit field (GPIOn) in the **RCGC2** register.

On reset, all GPIO pins (except for the five JTAG pins) are configured out of reset to be undriven (tristate): **GPIOAFSEL**=0, **GPIODEN**=0, **GPIOPDR**=0, and **GPIOPUR**=0. Table 7-6 on page 267 shows all possible configurations of the GPIO pads and the control register settings required to achieve them. Table 7-7 on page 267 shows how a rising edge interrupt would be configured for pin 2 of a GPIO port.

Table 7-6. GPIO Pad Configuration Examples

Configuration	GPIO Reg	GPIO Register Bit Value ^a											
Configuration	AFSEL	DIR	ODR	DEN	PUR	PDR	DR2R	DR4R	DR8R	SLR			
Digital Input (GPIO)	0	0	0	1	?	?	Х	Х	Х	Х			
Digital Output (GPIO)	0	1	0	1	?	?	?	?	?	?			
Open Drain Output (GPIO)	0	1	1	1	Х	Х	?	?	?	?			
Open Drain Input/Output (I ² C)	1	Х	1	1	Х	Х	?	?	?	?			
Digital Input (Timer CCP)	1	Х	0	1	?	?	Х	Х	Х	Х			
Digital Output (PWM)	1	Х	0	1	?	?	?	?	?	?			
Digital Output (Timer PWM)	1	Х	0	1	?	?	?	?	?	?			
Digital Input/Output (SSI)	1	Х	0	1	?	?	?	?	?	?			
Digital Input/Output (UART)	1	Х	0	1	?	?	?	?	?	?			
Analog Input (Comparator)	0	0	0	0	0	0	Х	Х	Х	Х			
Digital Output (Comparator)	1	Х	0	1	?	?	?	?	?	?			

a. X=Ignored (don't care bit)

Table 7-7. GPIO Interrupt Configuration Example

		Pin 2 Bit Val	ue ^a						
Register	Interrupt Event Trigger	7	6	5	4	3	2	1	0
GPIOIS	0=edge 1=level	Х	Х	Х	Х	Х	0	Х	Х

^{?=}Can be either 0 or 1, depending on the configuration

Table 7-7. GPIO Interrupt Configuration Example (continued)

	Desired	Pin 2 Bit Va	lue ^a						
Register	Register Interrupt Event Trigger	7	6	5	4	3	2	1	0
GPIOIBE	0=single edge 1=both edges	Х	Х	Х	Х	Х	0	Х	Х
GPIOIEV	0=Low level, or negative edge 1=High level, or positive edge		X	X	X	X	1	X	Х
GPIOIM	0=masked 1=not masked	0	0	0	0	0	1	0	0

a. X=Ignored (don't care bit)

7.4 Register Map

Table 7-8 on page 269 lists the GPIO registers. The offset listed is a hexadecimal increment to the register's address, relative to that GPIO port's base address:

GPIO Port A: 0x4000.4000GPIO Port B: 0x4000.5000GPIO Port C: 0x4000.6000

■ GPIO Port D: 0x4000.7000

GPIO Port E: 0x4002.4000GPIO Port F: 0x4002.5000GPIO Port G: 0x4002.6000

Note that the GPIO module clock must be enabled before the registers can be programmed (see page 221). There must be a delay of 3 system clocks after the GPIO module clock is enabled before any GPIO module registers are accessed.

Important: The GPIO registers in this chapter are duplicated in each GPIO block; however, depending on the block, all eight bits may not be connected to a GPIO pad. In those cases, writing to those unconnected bits has no effect, and reading those unconnected bits returns no meaningful data.

Note: The default reset value for the GPIOAFSEL, GPIOPUR, and GPIODEN registers are 0x0000.0000 for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins default to JTAG/SWD functionality. Because of this, the default reset value of these registers for GPIO Port B is 0x0000.0080 while the default reset value for Port C is 0x0000.000F.

The default register type for the **GPIOCR** register is RO for all GPIO pins with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins are currently the only GPIOs that are protected by the **GPIOCR** register. Because of this, the register type for GPIO Port B7 and GPIO Port C[3:0] is R/W.

The default reset value for the **GPIOCR** register is 0x0000.00FF for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). To ensure that the JTAG port is not accidentally programmed as a GPIO, these five pins default to non-committable. Because of this, the default reset value of **GPIOCR** for GPIO Port B is 0x0000.007F while the default reset value of GPIOCR for Port C is 0x0000.00F0.

Table 7-8. GPIO Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	GPIODATA	R/W	0x0000.0000	GPIO Data	271
0x400	GPIODIR	R/W	0x0000.0000	GPIO Direction	272
0x404	GPIOIS	R/W	0x0000.0000	GPIO Interrupt Sense	273
0x408	GPIOIBE	R/W	0x0000.0000	GPIO Interrupt Both Edges	274
0x40C	GPIOIEV	R/W	0x0000.0000	GPIO Interrupt Event	275
0x410	GPIOIM	R/W	0x0000.0000	GPIO Interrupt Mask	276
0x414	GPIORIS	RO	0x0000.0000	GPIO Raw Interrupt Status	277
0x418	GPIOMIS	RO	0x0000.0000	GPIO Masked Interrupt Status	278
0x41C	GPIOICR	W1C	0x0000.0000	GPIO Interrupt Clear	279
0x420	GPIOAFSEL	R/W	-	GPIO Alternate Function Select	280
0x500	GPIODR2R	R/W	0x0000.00FF	GPIO 2-mA Drive Select	282
0x504	GPIODR4R	R/W	0x0000.0000	GPIO 4-mA Drive Select	283
0x508	GPIODR8R	R/W	0x0000.0000	GPIO 8-mA Drive Select	284
0x50C	GPIOODR	R/W	0x0000.0000	GPIO Open Drain Select	285
0x510	GPIOPUR	R/W	-	GPIO Pull-Up Select	286
0x514	GPIOPDR	R/W	0x0000.0000	GPIO Pull-Down Select	287
0x518	GPIOSLR	R/W	0x0000.0000	GPIO Slew Rate Control Select	288
0x51C	GPIODEN	R/W	-	GPIO Digital Enable	289
0x520	GPIOLOCK	R/W	0x0000.0001	GPIO Lock	290
0x524	GPIOCR	-	-	GPIO Commit	291
0xFD0	GPIOPeriphID4	RO	0x0000.0000	GPIO Peripheral Identification 4	293
0xFD4	GPIOPeriphID5	RO	0x0000.0000	GPIO Peripheral Identification 5	294
0xFD8	GPIOPeriphID6	RO	0x0000.0000	GPIO Peripheral Identification 6	295
0xFDC	GPIOPeriphID7	RO	0x0000.0000	GPIO Peripheral Identification 7	296
0xFE0	GPIOPeriphID0	RO	0x0000.0061	GPIO Peripheral Identification 0	297
0xFE4	GPIOPeriphID1	RO	0x0000.0000	GPIO Peripheral Identification 1	298
0xFE8	GPIOPeriphID2	RO	0x0000.0018	GPIO Peripheral Identification 2	299
0xFEC	GPIOPeriphID3	RO	0x0000.0001	GPIO Peripheral Identification 3	300
				·	

Table 7-8. GPIO Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0xFF0	GPIOPCellID0	RO	0x0000.000D	GPIO PrimeCell Identification 0	301
0xFF4	GPIOPCellID1	RO	0x0000.00F0	GPIO PrimeCell Identification 1	302
0xFF8	GPIOPCellID2	RO	0x0000.0005	GPIO PrimeCell Identification 2	303
0xFFC	GPIOPCellID3	RO	0x0000.00B1	GPIO PrimeCell Identification 3	304

7.5 Register Descriptions

The remainder of this section lists and describes the GPIO registers, in numerical order by address offset.

Register 1: GPIO Data (GPIODATA), offset 0x000

The **GPIODATA** register is the data register. In software control mode, values written in the **GPIODATA** register are transferred onto the GPIO port pins if the respective pins have been configured as outputs through the **GPIO Direction (GPIODIR)** register (see page 272).

In order to write to **GPIODATA**, the corresponding bits in the mask, resulting from the address bus bits [9:2], must be High. Otherwise, the bit values remain unchanged by the write.

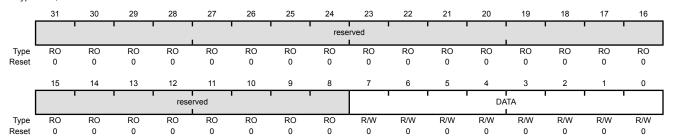
Similarly, the values read from this register are determined for each bit by the mask bit derived from the address used to access the data register, bits [9:2]. Bits that are 1 in the address mask cause the corresponding bits in **GPIODATA** to be read, and bits that are 0 in the address mask cause the corresponding bits in **GPIODATA** to be read as 0, regardless of their value.

A read from **GPIODATA** returns the last bit value written if the respective pins are configured as outputs, or it returns the value on the corresponding input pin when these are configured as inputs. All bits are cleared by a reset.

GPIO Data (GPIODATA)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x000

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DATA	R/W	0x00	GPIO Data

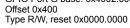
This register is virtually mapped to 256 locations in the address space. To facilitate the reading and writing of data to these registers by independent drivers, the data read from and the data written to the registers are masked by the eight address lines $\mathtt{ipaddr}[9:2]$. Reads from this register return its current state. Writes to this register only affect bits that are not masked by $\mathtt{ipaddr}[9:2]$ and are configured as outputs. See "Data Register Operation" on page 264 for examples of reads and writes.

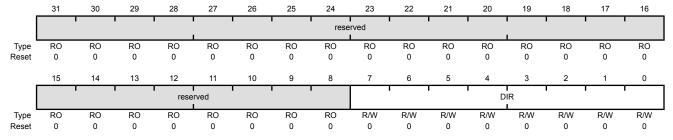
Register 2: GPIO Direction (GPIODIR), offset 0x400

The **GPIODIR** register is the data direction register. Bits set to 1 in the **GPIODIR** register configure the corresponding pin to be an output, while bits set to 0 configure the pins to be inputs. All bits are cleared by a reset, meaning all GPIO pins are inputs by default.

GPIO Direction (GPIODIR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x400





Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DIR	R/W	0x00	GPIO Data Direction

The ${\tt DIR}$ values are defined as follows:

- 0 Pins are inputs.
- 1 Pins are outputs.

Register 3: GPIO Interrupt Sense (GPIOIS), offset 0x404

The **GPIOIS** register is the interrupt sense register. Bits set to 1 in **GPIOIS** configure the corresponding pins to detect levels, while bits set to 0 configure the pins to detect edges. All bits are cleared by a reset.

GPIO Interrupt Sense (GPIOIS)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x404 Type R/W, reset 0x0000.0000

Reset

0

0

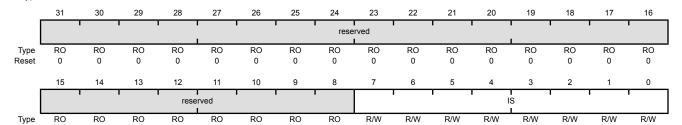
0

0

0

0

0



0

0

Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	IS	R/W	0x00	GPIO Interrupt Sense

The ${\tt IS}$ values are defined as follows:

0

0

0

0

0

0

0

- 0 Edge on corresponding pin is detected (edge-sensitive).
- 1 Level on corresponding pin is detected (level-sensitive).

Register 4: GPIO Interrupt Both Edges (GPIOIBE), offset 0x408

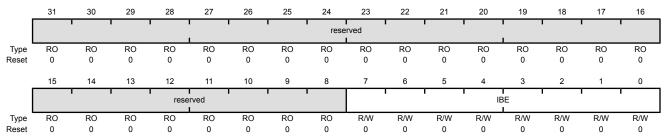
The **GPIOIBE** register is the interrupt both-edges register. When the corresponding bit in the **GPIO Interrupt Sense (GPIOIS)** register (see page 273) is set to detect edges, bits set to High in **GPIOIBE** configure the corresponding pin to detect both rising and falling edges, regardless of the corresponding bit in the **GPIO Interrupt Event (GPIOIEV)** register (see page 275). Clearing a bit configures the pin to be controlled by **GPIOIEV**. All bits are cleared by a reset.

GPIO Interrupt Both Edges (GPIOIBE)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x408

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	IBE	R/W	0x00	GPIO Interrupt Both Edges

The IBE values are defined as follows:

Value Description

- 0 Interrupt generation is controlled by the GPIO Interrupt Event (GPIOIEV) register (see page 275).
- 1 Both edges on the corresponding pin trigger an interrupt.

Note: Single edge is determined by the corresponding bit in **GPIOIEV**.

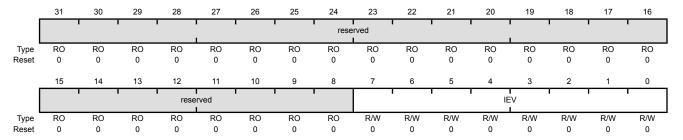
Register 5: GPIO Interrupt Event (GPIOIEV), offset 0x40C

The **GPIOIEV** register is the interrupt event register. Bits set to High in **GPIOIEV** configure the corresponding pin to detect rising edges or high levels, depending on the corresponding bit value in the **GPIO Interrupt Sense (GPIOIS)** register (see page 273). Clearing a bit configures the pin to detect falling edges or low levels, depending on the corresponding bit value in **GPIOIS**. All bits are cleared by a reset.

GPIO Interrupt Event (GPIOIEV)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x40C Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	IEV	R/W	0x00	GPIO Interrupt Event

The IEV values are defined as follows:

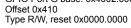
- 9 Falling edge or Low levels on corresponding pins trigger interrupts.
- Rising edge or High levels on corresponding pins trigger interrupts.

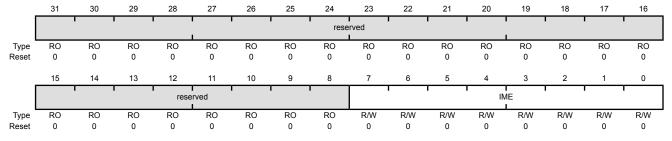
Register 6: GPIO Interrupt Mask (GPIOIM), offset 0x410

The **GPIOIM** register is the interrupt mask register. Bits set to High in **GPIOIM** allow the corresponding pins to trigger their individual interrupts and the combined **GPIOINTR** line. Clearing a bit disables interrupt triggering on that pin. All bits are cleared by a reset.

GPIO Interrupt Mask (GPIOIM)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x410





Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	IME	R/W	0x00	GPIO Interrupt Mask Enable

The ${\tt IME}$ values are defined as follows:

- 0 Corresponding pin interrupt is masked.
- 1 Corresponding pin interrupt is not masked.

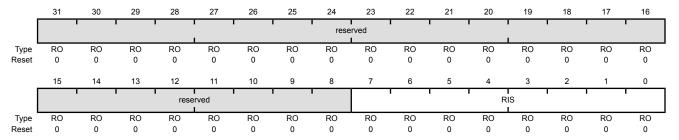
Register 7: GPIO Raw Interrupt Status (GPIORIS), offset 0x414

The **GPIORIS** register is the raw interrupt status register. Bits read High in **GPIORIS** reflect the status of interrupt trigger conditions detected (raw, prior to masking), indicating that all the requirements have been met, before they are finally allowed to trigger by the **GPIO Interrupt Mask (GPIOIM)** register (see page 276). Bits read as zero indicate that corresponding input pins have not initiated an interrupt. All bits are cleared by a reset.

GPIO Raw Interrupt Status (GPIORIS)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x414

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	RIS	RO	0x00	GPIO Interrupt Raw Status

Reflects the status of interrupt trigger condition detection on pins (raw, prior to masking).

The RIS values are defined as follows:

- 0 Corresponding pin interrupt requirements not met.
- 1 Corresponding pin interrupt has met requirements.

Register 8: GPIO Masked Interrupt Status (GPIOMIS), offset 0x418

The **GPIOMIS** register is the masked interrupt status register. Bits read High in **GPIOMIS** reflect the status of input lines triggering an interrupt. Bits read as Low indicate that either no interrupt has been generated, or the interrupt is masked.

In addition to providing GPIO functionality, PB4 can also be used as an external trigger for the ADC. If PB4 is configured as a non-masked interrupt pin (the appropriate bit of GPIOIM is set to 1), not only is an interrupt for PortB generated, but an external trigger signal is sent to the ADC. If the **ADC Event Multiplexer Select (ADCEMUX)** register is configured to use the external trigger, an ADC conversion is initiated.

If no other PortB pins are being used to generate interrupts, the **Interrupt 0-31 Set Enable (EN0)** register can disable the PortB interrupts, and the ADC interrupt can be used to read back the converted data. Otherwise, the PortB interrupt handler needs to ignore and clear interrupts on PB4, and wait for the ADC interrupt or the ADC interrupt must be disabled in the **EN0** register and the PortB interrupt handler must poll the ADC registers until the conversion is completed. See page 106 for more information.

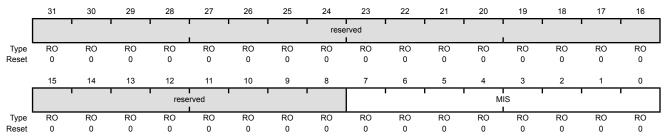
GPIOMIS is the state of the interrupt after masking.

GPIO Masked Interrupt Status (GPIOMIS)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x418

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	MIS	RO	0x00	GPIO Masked Interrupt Status

Masked value of interrupt due to corresponding pin.

The MIS values are defined as follows:

- 0 Corresponding GPIO line interrupt not active.
- 1 Corresponding GPIO line asserting interrupt.

Register 9: GPIO Interrupt Clear (GPIOICR), offset 0x41C

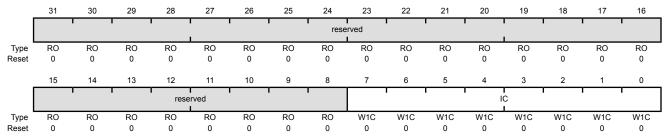
The **GPIOICR** register is the interrupt clear register. Writing a 1 to a bit in this register clears the corresponding interrupt edge detection logic register. Writing a 0 has no effect.

GPIO Interrupt Clear (GPIOICR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x41C

Type W1C, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	IC	W1C	0x00	GPIO Interrupt Clear

The ${\tt IC}$ values are defined as follows:

- 0 Corresponding interrupt is unaffected.
- 1 Corresponding interrupt is cleared.

Register 10: GPIO Alternate Function Select (GPIOAFSEL), offset 0x420

The **GPIOAFSEL** register is the mode control select register. Writing a 1 to any bit in this register selects the hardware control for the corresponding GPIO line. All bits are cleared by a reset, therefore no GPIO line is set to hardware control by default.

The GPIO commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Protection is currently provided for the five JTAG/SWD pins (PB7 and PC[3:0]). Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 280) are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register (see page 290) has been unlocked and the appropriate bits of the **GPIO Commit (GPIOCR)** register (see page 291) have been set to 1.

Important: All GPIO pins are tri-stated by default (GPIOAFSEL=0, GPIODEN=0, GPIOPDR=0, and GPIOPUR=0), with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). The JTAG/SWD pins default to their JTAG/SWD functionality (GPIOAFSEL=1, GPIODEN=1 and GPIOPUR=1). A Power-On-Reset (POR) or asserting RST puts both groups of pins back to their default state.

While debugging systems where PB7 is being used as a GPIO, care must be taken to ensure that a low value is not applied to the pin when the part is reset. Because PB7 reverts to the $\overline{\texttt{TRST}}$ function after reset, a Low value on the pin causes the JTAG controller to be reset, resulting in a loss of JTAG communication.

Caution – It is possible to create a software sequence that prevents the debugger from connecting to the Stellaris® microcontroller. If the program code loaded into flash immediately changes the JTAG pins to their GPIO functionality, the debugger may not have enough time to connect and halt the controller before the JTAG pin functionality switches. This may lock the debugger out of the part. This can be avoided with a software routine that restores JTAG functionality based on an external or software trigger.

GPIO Alternate Function Select (GPIOAFSEL)

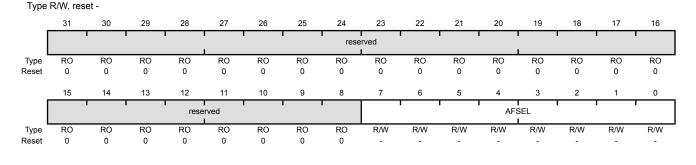
Name

Type

Reset

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x420

Bit/Field



31:8 reserved RO 0x00 Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Description

Bit/Field	Name	Type	Reset	Description
7:0	AFSEL	R/W	-	GPIO Alternate Function Select
				The AFSEL values are defined as follows:

Value Description

- 0 Software control of corresponding GPIO line (GPIO mode).
- 1 Hardware control of corresponding GPIO line (alternate hardware function).

Note: The default reset value for the GPIOAFSEL,

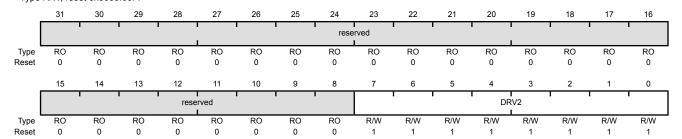
GPIOPUR, and GPIODEN registers are 0x0000.0000 for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins default to JTAG/SWD functionality. Because of this, the default reset value of these registers for GPIO Port B is 0x0000.0080 while the default reset value for Port C is 0x0000.000F.

Register 11: GPIO 2-mA Drive Select (GPIODR2R), offset 0x500

The **GPIODR2R** register is the 2-mA drive control register. It allows for each GPIO signal in the port to be individually configured without affecting the other pads. When writing a DRV2 bit for a GPIO signal, the corresponding DRV4 bit in the **GPIODR4R** register and the DRV8 bit in the **GPIODR8R** register are automatically cleared by hardware.

GPIO 2-mA Drive Select (GPIODR2R)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x500 Type R/W, reset 0x0000.00FF



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DRV2	R/W	0xFF	Output Pad 2-mA Drive Enable

A write of 1 to either **GPIODR4[n]** or **GPIODR8[n]** clears the corresponding 2-mA enable bit. The change is effective on the second clock cycle after the write.

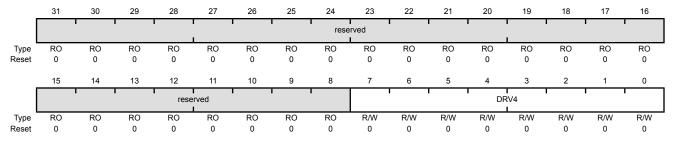
Register 12: GPIO 4-mA Drive Select (GPIODR4R), offset 0x504

The **GPIODR4R** register is the 4-mA drive control register. It allows for each GPIO signal in the port to be individually configured without affecting the other pads. When writing the DRV4 bit for a GPIO signal, the corresponding DRV2 bit in the **GPIODR2R** register and the DRV8 bit in the **GPIODR8R** register are automatically cleared by hardware.

GPIO 4-mA Drive Select (GPIODR4R)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 GPIO Port G base: 0x4002.6000

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DRV4	R/W	0x00	Output Pad 4-mA Drive Enable

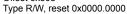
A write of 1 to either **GPIODR2[n]** or **GPIODR8[n]** clears the corresponding 4-mA enable bit. The change is effective on the second clock cycle after the write.

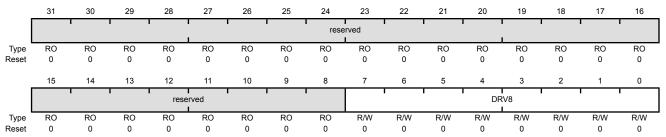
Register 13: GPIO 8-mA Drive Select (GPIODR8R), offset 0x508

The **GPIODR8R** register is the 8-mA drive control register. It allows for each GPIO signal in the port to be individually configured without affecting the other pads. When writing the DRV8 bit for a GPIO signal, the corresponding DRV2 bit in the **GPIODR2R** register and the DRV4 bit in the **GPIODR4R** register are automatically cleared by hardware.

GPIO 8-mA Drive Select (GPIODR8R)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x508





Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DRV8	R/W	0x00	Output Pad 8-mA Drive Enable

A write of 1 to either **GPIODR2[n]** or **GPIODR4[n]** clears the corresponding 8-mA enable bit. The change is effective on the second clock cycle after the write.

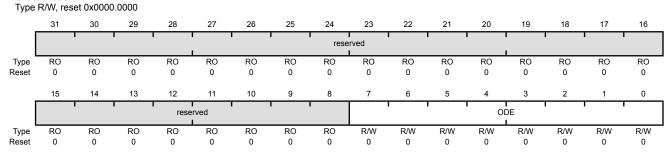
Register 14: GPIO Open Drain Select (GPIOODR), offset 0x50C

The **GPIOODR** register is the open drain control register. Setting a bit in this register enables the open drain configuration of the corresponding GPIO pad. When open drain mode is enabled, the corresponding bit should also be set in the **GPIO Digital Enable (GPIODEN)** register (see page 289). Corresponding bits in the drive strength registers (**GPIODR2R**, **GPIODR4R**, **GPIODR8R**, and **GPIOSLR**) can be set to achieve the desired rise and fall times. The GPIO acts as an open-drain input if the corresponding bit in the **GPIODIR** register is cleared. If open drain is selected while the GPIO is configured as an input, the GPIO will remain an input and the open-drain selection has no effect until the GPIO is changed to an output.

When using the I^2C module, in addition to configuring the pin to open drain, the **GPIO Alternate** Function Select (GPIOAFSEL) register bits for the I^2C clock and data pins should be set to 1 (see examples in "Initialization and Configuration" on page 267).

GPIO Open Drain Select (GPIOODR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x50C



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	ODE	R/W	0x00	Output Pad Open Drain Enable
				The ODE values are defined as follows:

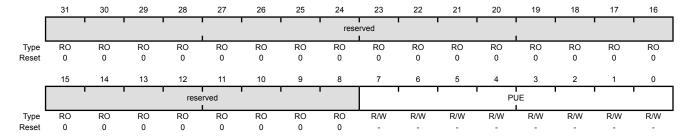
- 0 Open drain configuration is disabled.
- 1 Open drain configuration is enabled.

Register 15: GPIO Pull-Up Select (GPIOPUR), offset 0x510

The **GPIOPUR** register is the pull-up control register. When a bit is set to 1, it enables a weak pull-up resistor on the corresponding GPIO signal. Setting a bit in **GPIOPUR** automatically clears the corresponding bit in the **GPIO Pull-Down Select (GPIOPDR)** register (see page 287).

GPIO Pull-Up Select (GPIOPUR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x510 Type R/W, reset -



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PUE	R/W	_	Pad Weak Pull-Up Enable

Value Description

- 0 The corresponding pin's weak pull-up resistor is disabled.
- 1 The corresponding pin's weak pull-up resistor is enabled.

A write of 1 to **GPIOPDR[n]** clears the corresponding **GPIOPUR[n]** enables. The change is effective on the second clock cycle after the write.

Note:

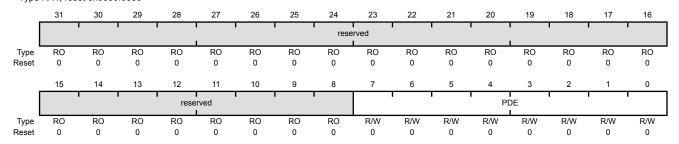
The default reset value for the **GPIOAFSEL**, **GPIOPUR**, and **GPIODEN** registers are 0x0000.0000 for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins default to JTAG/SWD functionality. Because of this, the default reset value of these registers for GPIO Port B is 0x0000.0080 while the default reset value for Port C is 0x0000.000F.

Register 16: GPIO Pull-Down Select (GPIOPDR), offset 0x514

The **GPIOPDR** register is the pull-down control register. When a bit is set to 1, it enables a weak pull-down resistor on the corresponding GPIO signal. Setting a bit in **GPIOPDR** automatically clears the corresponding bit in the **GPIO Pull-Up Select (GPIOPUR)** register (see page 286).

GPIO Pull-Down Select (GPIOPDR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x514 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PDE	R/W	0x00	Pad Weak Pull-Down Enable

Value Description

- 0 The corresponding pin's weak pull-down resistor is disabled.
- 1 The corresponding pin's weak pull-down resistor is enabled.

A write of 1 to **GPIOPUR[n]** clears the corresponding **GPIOPDR[n]** enables. The change is effective on the second clock cycle after the write.

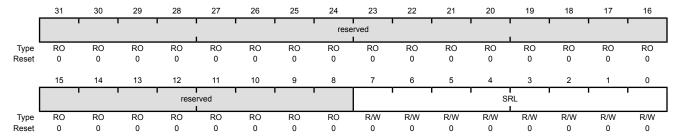
Register 17: GPIO Slew Rate Control Select (GPIOSLR), offset 0x518

The **GPIOSLR** register is the slew rate control register. Slew rate control is only available when using the 8-mA drive strength option via the **GPIO 8-mA Drive Select (GPIODR8R)** register (see page 284).

GPIO Slew Rate Control Select (GPIOSLR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x518 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	SRL	R/W	0x00	Slew Rate Limit Enable (8-mA drive only)
				The SRL values are defined as follows:

- 0 Slew rate control disabled.
- 1 Slew rate control enabled.

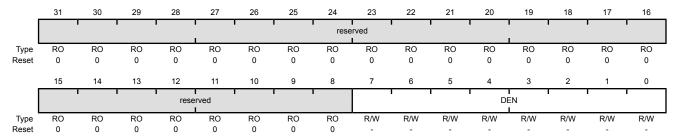
Register 18: GPIO Digital Enable (GPIODEN), offset 0x51C

Note: Pins configured as digital inputs are Schmitt-triggered.

The **GPIODEN** register is the digital enable register. By default, with the exception of the GPIO signals used for JTAG/SWD function, all other GPIO signals are configured out of reset to be undriven (tristate). Their digital function is disabled; they do not drive a logic value on the pin and they do not allow the pin voltage into the GPIO receiver. To use the pin in a digital function (either GPIO or alternate function), the corresponding GPIODEN bit must be set.

GPIO Digital Enable (GPIODEN)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x51C Type R/W, reset -



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DEN	R/W	-	Digital Enable

The DEN values are defined as follows:

Value Description

- 0 Digital functions disabled.
- Digital functions enabled.

Note: The default reset value for the GPIOAFSEL,
GPIOPUR, and GPIODEN registers are 0x0000.0000
for all GPIO pins, with the exception of the five
JTAG/SWD pins (PB7 and PC[3:0]). These five pins
default to JTAG/SWD functionality. Because of this,
the default reset value of these registers for GPIO

Port B is 0x0000.0080 while the default reset value

for Port C is 0x0000.000F.

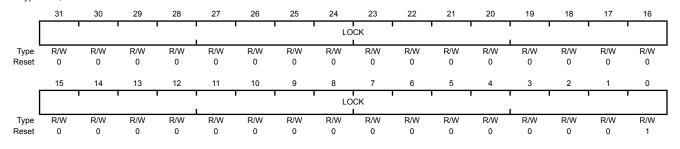
Register 19: GPIO Lock (GPIOLOCK), offset 0x520

The **GPIOLOCK** register enables write access to the **GPIOCR** register (see page 291). Writing 0x1ACC.E551 to the **GPIOLOCK** register will unlock the **GPIOCR** register. Writing any other value to the **GPIOLOCK** register re-enables the locked state. Reading the **GPIOLOCK** register returns the lock status rather than the 32-bit value that was previously written. Therefore, when write accesses are disabled, or locked, reading the **GPIOLOCK** register returns 0x00000001. When write accesses are enabled, or unlocked, reading the **GPIOLOCK** register returns 0x000000000.

GPIO Lock (GPIOLOCK)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x520

Type R/W, reset 0x0000.0001



Bit/Field	Name	Type	Reset	Description
31:0	LOCK	R/W	0x0000.0001	GPIO Lock

A write of the value 0x1ACC.E551 unlocks the **GPIO Commit (GPIOCR)** register for write access.

A write of any other value or a write to the **GPIOCR** register reapplies the lock, preventing any register updates. A read of this register returns the following values:

Value Description
0x0000.0001 Locked
0x0000.0000 Unlocked

Register 20: GPIO Commit (GPIOCR), offset 0x524

The GPIOCR register is the commit register. The value of the GPIOCR register determines which bits of the GPIOAFSEL register are committed when a write to the GPIOAFSEL register is performed. If a bit in the GPIOCR register is a zero, the data being written to the corresponding bit in the GPIOAFSEL register will not be committed and will retain its previous value. If a bit in the GPIOCR register is a one, the data being written to the corresponding bit of the GPIOAFSEL register will be committed to the register and will reflect the new value.

The contents of the GPIOCR register can only be modified if the GPIOLOCK register is unlocked. Writes to the GPIOCR register are ignored if the GPIOLOCK register is locked.

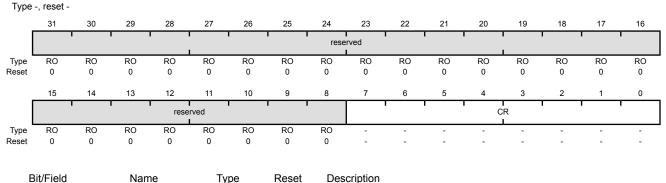
Important: This register is designed to prevent accidental programming of the registers that control connectivity to the JTAG/SWD debug hardware. By initializing the bits of the GPIOCR register to 0 for PB7 and PC[3:0], the JTAG/SWD debug port can only be converted to GPIOs through a deliberate set of writes to the GPIOLOCK, GPIOCR, and the corresponding registers.

> Because this protection is currently only implemented on the JTAG/SWD pins on PB7 and PC[3:0], all of the other bits in the **GPIOCR** registers cannot be written with 0x0. These bits are hardwired to 0x1, ensuring that it is always possible to commit new values to the **GPIOAFSEL**register bits of these other pins.

GPIO Commit (GPIOCR)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000 5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0x524



Bit/Field Name Type Reset 31:8 reserved RO 0x00

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Bit/Field	Name	Type	Reset	Description
7:0	CR	_	_	GPIO Commit

On a bit-wise basis, any bit set allows the corresponding **GPIOAFSEL** bit to be set to its alternate function.

Note:

The default register type for the **GPIOCR** register is RO for all GPIO pins with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins are currently the only GPIOs that are protected by the **GPIOCR** register. Because of this, the register type for GPIO Port B7 and GPIO Port C[3:0] is R/W.

The default reset value for the **GPIOCR** register is 0x0000.00FF for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). To ensure that the JTAG port is not accidentally programmed as a GPIO, these five pins default to non-committable. Because of this, the default reset value of **GPIOCR** for GPIO Port B is 0x0000.007F while the default reset value of GPIOCR for Port C is 0x0000.00F0.

Register 21: GPIO Peripheral Identification 4 (GPIOPeriphID4), offset 0xFD0

The GPIOPeriphID4, GPIOPeriphID5, GPIOPeriphID6, and GPIOPeriphID7 registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 4 (GPIOPeriphID4)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFD0 Type RO, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	·					rese	rved					1		
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		•	'	rese	rved	l		1	'		l	PI	I D4 I	•		'
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

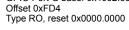
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x00	GPIO Peripheral ID Register[7:0]

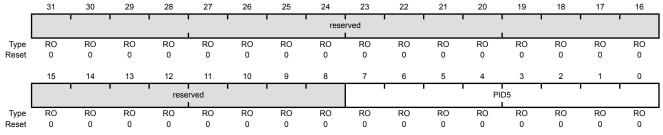
Register 22: GPIO Peripheral Identification 5 (GPIOPeriphID5), offset 0xFD4

The **GPIOPeriphID4**, **GPIOPeriphID5**, **GPIOPeriphID6**, and **GPIOPeriphID7** registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 5 (GPIOPeriphID5)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port G base: 0x4002.5000 GPIO Port G base: 0x4002.6000





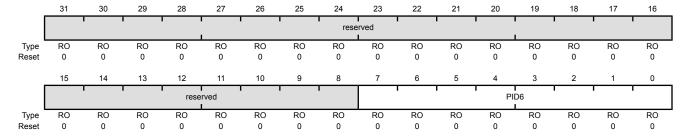
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID5	RO	0x00	GPIO Peripheral ID Register[15:8]

Register 23: GPIO Peripheral Identification 6 (GPIOPeriphID6), offset 0xFD8

The GPIOPeriphID4, GPIOPeriphID5, GPIOPeriphID6, and GPIOPeriphID7 registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 6 (GPIOPeriphID6)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFD8
Type RO, reset 0x0000.0000



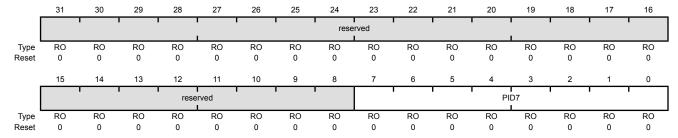
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID6	RO	0x00	GPIO Peripheral ID Register[23:16]

Register 24: GPIO Peripheral Identification 7 (GPIOPeriphID7), offset 0xFDC

The GPIOPeriphID4, GPIOPeriphID5, GPIOPeriphID6, and GPIOPeriphID7 registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 7 (GPIOPeriphID7)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFDC Type RO, reset 0x0000.0000



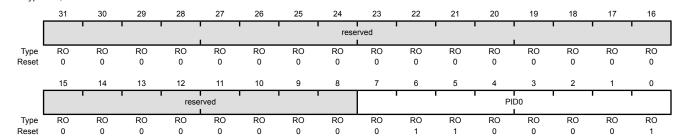
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID7	RO	0x00	GPIO Peripheral ID Register[31:24]

Register 25: GPIO Peripheral Identification 0 (GPIOPeriphID0), offset 0xFE0

The **GPIOPeriphID0**, **GPIOPeriphID1**, **GPIOPeriphID2**, and **GPIOPeriphID3** registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 0 (GPIOPeriphID0)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFE0 Type RO, reset 0x0000.0061



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID0	RO	0x61	GPIO Peripheral ID Register[7:0]

Can be used by software to identify the presence of this peripheral.

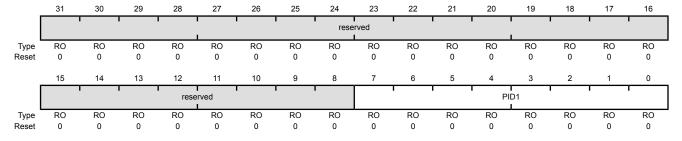
Register 26: GPIO Peripheral Identification 1 (GPIOPeriphID1), offset 0xFE4

The **GPIOPeriphID0**, **GPIOPeriphID1**, **GPIOPeriphID2**, and **GPIOPeriphID3** registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 1 (GPIOPeriphID1)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0xFE4
Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID1	RO	0x00	GPIO Peripheral ID Register[15:8]

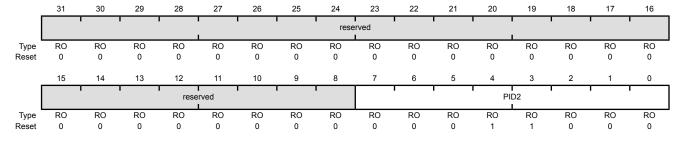
Can be used by software to identify the presence of this peripheral.

Register 27: GPIO Peripheral Identification 2 (GPIOPeriphID2), offset 0xFE8

The GPIOPeriphID0, GPIOPeriphID1, GPIOPeriphID2, and GPIOPeriphID3 registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 2 (GPIOPeriphID2)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFE8
Type RO, reset 0x0000.0018



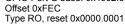
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID2	RO	0x18	GPIO Peripheral ID Register[23:16] Can be used by software to identify the presence of this peripheral.

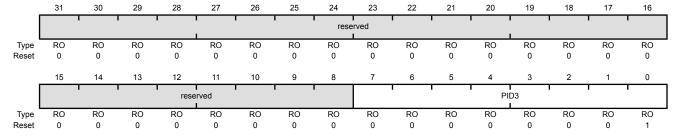
Register 28: GPIO Peripheral Identification 3 (GPIOPeriphID3), offset 0xFEC

The **GPIOPeriphID0**, **GPIOPeriphID1**, **GPIOPeriphID2**, and **GPIOPeriphID3** registers can conceptually be treated as one 32-bit register; each register contains eight bits of the 32-bit register, used by software to identify the peripheral.

GPIO Peripheral Identification 3 (GPIOPeriphID3)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000





Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	GPIO Peripheral ID Register[31:24]

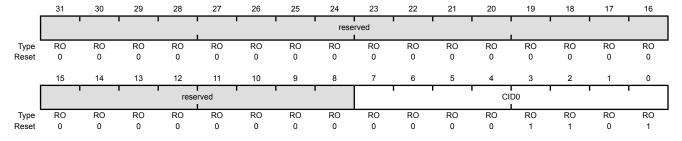
Can be used by software to identify the presence of this peripheral.

Register 29: GPIO PrimeCell Identification 0 (GPIOPCellID0), offset 0xFF0

The GPIOPCeIIID1, GPIOPCeIIID2, and GPIOPCeIIID3 registers are four 8-bit wide registers, that can conceptually be treated as one 32-bit register. The register is used as a standard cross-peripheral identification system.

GPIO PrimeCell Identification 0 (GPIOPCellID0)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFF0
Type RO, reset 0x0000.000D



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	GPIO PrimeCell ID Register[7:0]

Provides software a standard cross-peripheral identification system.

Register 30: GPIO PrimeCell Identification 1 (GPIOPCellID1), offset 0xFF4

The **GPIOPCeIIID1**, **GPIOPCeIIID1**, and **GPIOPCeIIID3** registers are four 8-bit wide registers, that can conceptually be treated as one 32-bit register. The register is used as a standard cross-peripheral identification system.

GPIO PrimeCell Identification 1 (GPIOPCellID1)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 GPIO Port G base: 0x4002.6000 Offset 0xFF4 Type RO, reset 0x0000.00F0

Reset

0

0

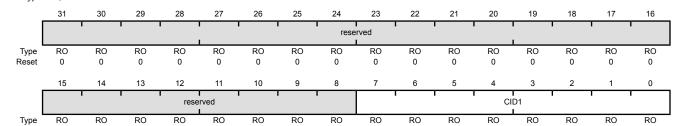
0

0

0

0

0



0

1

Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	GPIO PrimeCell ID Register[15:8]

Provides software a standard cross-peripheral identification system.

0

0

0

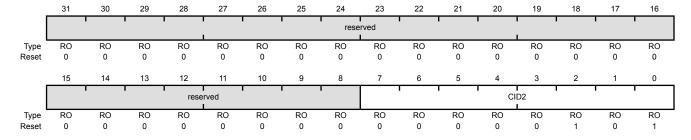
0

Register 31: GPIO PrimeCell Identification 2 (GPIOPCelIID2), offset 0xFF8

The GPIOPCeIIID1, GPIOPCeIIID2, and GPIOPCeIIID3 registers are four 8-bit wide registers, that can conceptually be treated as one 32-bit register. The register is used as a standard cross-peripheral identification system.

GPIO PrimeCell Identification 2 (GPIOPCellID2)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFF8
Type RO, reset 0x0000.0005



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	GPIO PrimeCell ID Register[23:16]

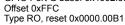
Provides software a standard cross-peripheral identification system.

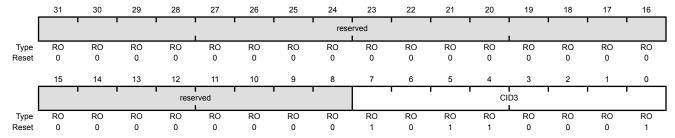
Register 32: GPIO PrimeCell Identification 3 (GPIOPCellID3), offset 0xFFC

The **GPIOPCeIIID1**, **GPIOPCeIIID1**, and **GPIOPCeIIID3** registers are four 8-bit wide registers, that can conceptually be treated as one 32-bit register. The register is used as a standard cross-peripheral identification system.

GPIO PrimeCell Identification 3 (GPIOPCellID3)

GPIO Port A base: 0x4000.4000 GPIO Port B base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port D base: 0x4000.7000 GPIO Port E base: 0x4002.4000 GPIO Port F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0x





Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID3	RO	0xB1	GPIO PrimeCell ID Register[31:24]

Provides software a standard cross-peripheral identification system.

8 General-Purpose Timers

Programmable timers can be used to count or time external events that drive the Timer input pins. The Stellaris[®] General-Purpose Timer Module (GPTM) contains three GPTM blocks (Timer0, Timer1, and Timer 2). Each GPTM block provides two 16-bit timers/counters (referred to as TimerA and TimerB) that can be configured to operate independently as timers or event counters, or configured to operate as one 32-bit timer or one 32-bit Real-Time Clock (RTC).

In addition, timers can be used to trigger analog-to-digital conversions (ADC). The ADC trigger signals from all of the general-purpose timers are ORed together before reaching the ADC module, so only one timer should be used to trigger ADC events.

The GPT Module is one timing resource available on the Stellaris microcontrollers. Other timer resources include the System Timer (SysTick) (see 91) and the PWM timer in the PWM module (see "PWM Timer" on page 581).

The General-Purpose Timers provide the following features:

- Three General-Purpose Timer Modules (GPTM), each of which provides two 16-bit timers/counters. Each GPTM can be configured to operate independently:
 - As a single 32-bit timer
 - As one 32-bit Real-Time Clock (RTC) to event capture
 - For Pulse Width Modulation (PWM)
 - To trigger analog-to-digital conversions
- 32-bit Timer modes
 - Programmable one-shot timer
 - Programmable periodic timer
 - Real-Time Clock when using an external 32.768-KHz clock as the input
 - User-enabled stalling when the controller asserts CPU Halt flag during debug
 - ADC event trigger
- 16-bit Timer modes
 - General-purpose timer function with an 8-bit prescaler (for one-shot and periodic modes only)
 - Programmable one-shot timer
 - Programmable periodic timer
 - User-enabled stalling when the controller asserts CPU Halt flag during debug
 - ADC event trigger
- 16-bit Input Capture modes
 - Input edge count capture

- Input edge time capture
- 16-bit PWM mode
 - Simple PWM mode with software-programmable output inversion of the PWM signal

8.1 Block Diagram

Note: In Figure 8-1 on page 306, the specific CCP pins available depend on the Stellaris device. See Table 8-1 on page 306 for the available CCPs.

Figure 8-1. GPTM Module Block Diagram

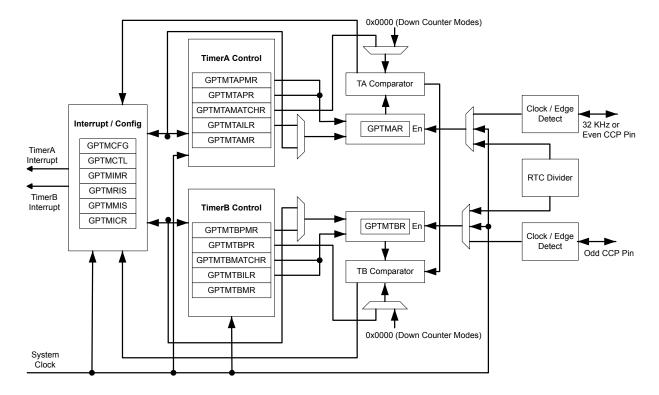


Table 8-1. Available CCP Pins

Timer	16-Bit Up/Down Counter	Even CCP Pin	Odd CCP Pin
Timer 0	TimerA	CCP0	-
	TimerB	-	CCP1
Timer 1	TimerA	CCP2	-
	TimerB	-	CCP3
Timer 2	TimerA	-	-
	TimerB	-	-

8.2 Signal Description

Table 8-2 on page 307 and Table 8-3 on page 307 list the external signals of the GP Timer module and describe the function of each. The GP Timer signals are alternate functions for some GPIO signals and default to be GPIO signals at reset. The column in the table below titled "Pin Assignment"

lists the possible GPIO pin placements for these GP Timer signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the GP Timer function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 8-2. General-Purpose Timers Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
CCP0	66	I/O	TTL	Capture/Compare/PWM 0.
CCP1	34	I/O	TTL	Capture/Compare/PWM 1.
CCP2	67	I/O	TTL	Capture/Compare/PWM 2.
CCP3	23	I/O	TTL	Capture/Compare/PWM 3.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 8-3. General-Purpose Timers Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
CCP0	E12	I/O	TTL	Capture/Compare/PWM 0.
CCP1	L6	I/O	TTL	Capture/Compare/PWM 1.
CCP2	D12	I/O	TTL	Capture/Compare/PWM 2.
CCP3	M2	I/O	TTL	Capture/Compare/PWM 3.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

8.3 Functional Description

The main components of each GPTM block are two free-running 16-bit up/down counters (referred to as TimerA and TimerB), two 16-bit match registers, two prescaler match registers, and two 16-bit load/initialization registers and their associated control functions. The exact functionality of each GPTM is controlled by software and configured through the register interface.

Software configures the GPTM using the **GPTM Configuration (GPTMCFG)** register (see page 317), the **GPTM TimerA Mode (GPTMTAMR)** register (see page 318), and the **GPTM TimerB Mode (GPTMTBMR)** register (see page 320). When in one of the 32-bit modes, the timer can only act as a 32-bit timer. However, when configured in 16-bit mode, the GPTM can have its two 16-bit timers configured in any combination of the 16-bit modes.

8.3.1 GPTM Reset Conditions

After reset has been applied to the GPTM module, the module is in an inactive state, and all control registers are cleared and in their default states. Counters TimerA and TimerB are initialized to 0xFFFF, along with their corresponding load registers: the GPTM TimerA Interval Load (GPTMTBILR) register (see page 331) and the GPTM TimerB Interval Load (GPTMTBILR) register (see page 332). The prescale counters are initialized to 0x00: the GPTM TimerA Prescale (GPTMTAPR) register (see page 335) and the GPTM TimerB Prescale (GPTMTBPR) register (see page 336).

8.3.2 32-Bit Timer Operating Modes

This section describes the three GPTM 32-bit timer modes (One-Shot, Periodic, and RTC) and their configuration.

The GPTM is placed into 32-bit mode by writing a 0 (One-Shot/Periodic 32-bit timer mode) or a 1 (RTC mode) to the **GPTM Configuration (GPTMCFG)** register. In both configurations, certain GPTM registers are concatenated to form pseudo 32-bit registers. These registers include:

- GPTM TimerA Interval Load (GPTMTAILR) register [15:0], see page 331
- GPTM TimerB Interval Load (GPTMTBILR) register [15:0], see page 332
- **GPTM TimerA (GPTMTAR)** register [15:0], see page 339
- **GPTM TimerB (GPTMTBR)** register [15:0], see page 340

In the 32-bit modes, the GPTM translates a 32-bit write access to **GPTMTAILR** into a write access to both **GPTMTAILR** and **GPTMTBILR**. The resulting word ordering for such a write operation is:

```
GPTMTBILR[15:0]:GPTMTAILR[15:0]
```

Likewise, a read access to **GPTMTAR** returns the value:

```
GPTMTBR[15:0]:GPTMTAR[15:0]
```

8.3.2.1 32-Bit One-Shot/Periodic Timer Mode

In 32-bit one-shot and periodic timer modes, the concatenated versions of the TimerA and TimerB registers are configured as a 32-bit down-counter. The selection of one-shot or periodic mode is determined by the value written to the TAMR field of the **GPTM TimerA Mode (GPTMTAMR)** register (see page 318), and there is no need to write to the **GPTM TimerB Mode (GPTMTBMR)** register.

When software writes the TAEN bit in the **GPTM Control (GPTMCTL)** register (see page 322), the timer begins counting down from its preloaded value. Once the 0x0000.0000 state is reached, the timer reloads its start value from the concatenated **GPTMTAILR** on the next cycle. If configured to be a one-shot timer, the timer stops counting and clears the TAEN bit in the **GPTMCTL** register. If configured as a periodic timer, it continues counting.

In addition to reloading the count value, the GPTM generates interrupts and triggers when it reaches the 0x000.0000 state. The GPTM sets the TATORIS bit in the GPTM Raw Interrupt Status (GPTMRIS) register (see page 327), and holds it until it is cleared by writing the GPTM Interrupt Clear (GPTMICR) register (see page 329). If the time-out interrupt is enabled in the GPTM Interrupt Mask (GPTMIMR) register (see page 325), the GPTM also sets the TATOMIS bit in the GPTM Masked Interrupt Status (GPTMMIS) register (see page 328). The ADC trigger is enabled by setting the TAOTE bit in GPTMCTL.

If software reloads the **GPTMTAILR** register while the counter is running, the counter loads the new value on the next clock cycle and continues counting from the new value.

If the TASTALL bit in the **GPTMCTL** register is set, the timer freezes counting while the processor is halted by the debugger. The timer resumes counting when the processor resumes execution.

8.3.2.2 32-Bit Real-Time Clock Timer Mode

In Real-Time Clock (RTC) mode, the concatenated versions of the TimerA and TimerB registers are configured as a 32-bit up-counter. When RTC mode is selected for the first time, the counter is loaded with a value of 0x0000.0001. All subsequent load values must be written to the **GPTM TimerA Match (GPTMTAMATCHR)** register (see page 333) by the controller.

The input clock on an even CCP input is required to be 32.768 KHz in RTC mode. The clock signal is then divided down to a 1 Hz rate and is passed along to the input of the 32-bit counter.

When software writes the TAEN bit in the **GPTMCTL** register, the counter starts counting up from its preloaded value of 0x0000.0001. When the current count value matches the preloaded value in the **GPTMTAMATCHR** register, it rolls over to a value of 0x0000.0000 and continues counting until

either a hardware reset, or it is disabled by software (clearing the TAEN bit). When a match occurs, the GPTM asserts the RTCRIS bit in **GPTMRIS**. If the RTC interrupt is enabled in **GPTMIMR**, the GPTM also sets the RTCMIS bit in **GPTMMIS** and generates a controller interrupt. The status flags are cleared by writing the RTCCINT bit in **GPTMICR**.

If the TASTALL and/or TBSTALL bits in the **GPTMCTL** register are set, the timer does not freeze if the RTCEN bit is set in **GPTMCTL**.

8.3.3 16-Bit Timer Operating Modes

The GPTM is placed into global 16-bit mode by writing a value of 0x4 to the **GPTM Configuration** (**GPTMCFG**) register (see page 317). This section describes each of the GPTM 16-bit modes of operation. TimerA and TimerB have identical modes, so a single description is given using an **n** to reference both.

8.3.3.1 16-Bit One-Shot/Periodic Timer Mode

In 16-bit one-shot and periodic timer modes, the timer is configured as a 16-bit down-counter with an optional 8-bit prescaler that effectively extends the counting range of the timer to 24 bits. The selection of one-shot or periodic mode is determined by the value written to the TnMR field of the **GPTMTnMR** register. The optional prescaler is loaded into the **GPTM Timern Prescale (GPTMTnPR)** register.

When software writes the \mathtt{TnEN} bit in the **GPTMCTL** register, the timer begins counting down from its preloaded value. Once the 0x0000 state is reached, the timer reloads its start value from **GPTMTnILR** and **GPTMTnPR** on the next cycle. If configured to be a one-shot timer, the timer stops counting and clears the \mathtt{TnEN} bit in the **GPTMCTL** register. If configured as a periodic timer, it continues counting.

In addition to reloading the count value, the timer generates interrupts and triggers when it reaches the 0x0000 state. The GPTM sets the TnTORIS bit in the **GPTMRIS** register, and holds it until it is cleared by writing the **GPTMICR** register. If the time-out interrupt is enabled in **GPTMIMR**, the GPTM also sets the TnTOMIS bit in **GPTMISR** and generates a controller interrupt. The ADC trigger is enabled by setting the TnOTE bit in the **GPTMCTL** register.

If software reloads the **GPTMTAILR** register while the counter is running, the counter loads the new value on the next clock cycle and continues counting from the new value.

If the TnSTALL bit in the **GPTMCTL** register is set, the timer freezes counting while the processor is halted by the debugger. The timer resumes counting when the processor resumes execution.

The following example shows a variety of configurations for a 16-bit free running timer while using the prescaler. All values assume a 50-MHz clock with Tc=20 ns (clock period).

Prescale	#Clock (T c) ^a	Max Time	Units
00000000	1	1.3107	mS
0000001	2	2.6214	mS
0000010	3	3.9322	mS
11111101	254	332.9229	mS
11111110	255	334.2336	mS
1111111	256	335.5443	mS

Table 8-4. 16-Bit Timer With Prescaler Configurations

a. Tc is the clock period.

8.3.3.2 16-Bit Input Edge Count Mode

Note: For rising-edge detection, the input signal must be High for at least two system clock periods following the rising edge. Similarly, for falling-edge detection, the input signal must be Low for at least two system clock periods following the falling edge. Based on this criteria, the maximum input frequency for edge detection is 1/4 of the system frequency.

Note: The prescaler is not available in 16-Bit Input Edge Count mode.

In Edge Count mode, the timer is configured as a down-counter capable of capturing three types of events: rising edge, falling edge, or both. To place the timer in Edge Count mode, the TnCMR bit of the GPTMTnMR register must be set to 0. The type of edge that the timer counts is determined by the TnEVENT fields of the GPTMCTL register. During initialization, the GPTM Timern Match (GPTMTnMATCHR) register is configured so that the difference between the value in the GPTMTnILR register and the GPTMTnMATCHR register equals the number of edge events that must be counted.

When software writes the TnEN bit in the **GPTM Control (GPTMCTL)** register, the timer is enabled for event capture. Each input event on the CCP pin decrements the counter by 1 until the event count matches **GPTMTnMATCHR**. When the counts match, the GPTM asserts the CnMRIS bit in the **GPTMRIS** register (and the CnMMIS bit, if the interrupt is not masked).

The counter is then reloaded using the value in **GPTMTnILR**, and stopped since the GPTM automatically clears the TnEN bit in the **GPTMCTL** register. Once the event count has been reached, all further events are ignored until TnEN is re-enabled by software.

Figure 8-2 on page 310 shows how input edge count mode works. In this case, the timer start value is set to **GPTMTnILR** =0x000A and the match value is set to **GPTMTnMATCHR** =0x0006 so that four edge events are counted. The counter is configured to detect both edges of the input signal.

Note that the last two edges are not counted since the timer automatically clears the TnEN bit after the current count matches the value in the **GPTMTnMATCHR** register.

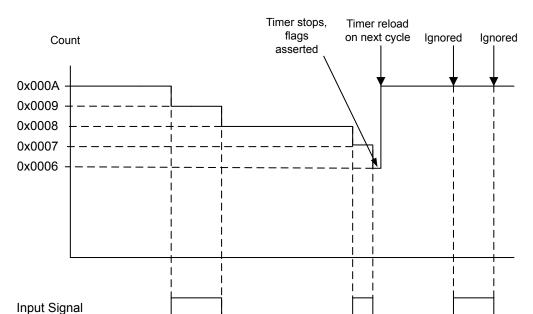


Figure 8-2. 16-Bit Input Edge Count Mode Example

July 15, 2014

8.3.3.3 16-Bit Input Edge Time Mode

Note: For rising-edge detection, the input signal must be High for at least two system clock periods following the rising edge. Similarly, for falling edge detection, the input signal must be Low for at least two system clock periods following the falling edge. Based on this criteria, the maximum input frequency for edge detection is 1/4 of the system frequency.

Note: The prescaler is not available in 16-Bit Input Edge Time mode.

In Edge Time mode, the timer is configured as a free-running down-counter initialized to the value loaded in the **GPTMTnILR** register (or 0xFFFF at reset). The timer is capable of capturing three types of events: rising edge, falling edge, or both. The timer is placed into Edge Time mode by setting the TnCMR bit in the **GPTMTnMR** register, and the type of event that the timer captures is determined by the TnEVENT fields of the **GPTMCTL** register.

When software writes the TnEN bit in the **GPTMCTL** register, the timer is enabled for event capture. When the selected input event is detected, the current Tn counter value is captured in the **GPTMTnR** register and is available to be read by the controller. The GPTM then asserts the CnERIS bit (and the CnEMIS bit, if the interrupt is not masked).

After an event has been captured, the timer does not stop counting. It continues to count until the \mathtt{TnEN} bit is cleared. When the timer reaches the 0x0000 state, it is reloaded with the value from the **GPTMTnILR** register.

Figure 8-3 on page 311 shows how input edge timing mode works. In the diagram, it is assumed that the start value of the timer is the default value of 0xFFFF, and the timer is configured to capture rising edge events.

Each time a rising edge event is detected, the current count value is loaded into the **GPTMTnR** register, and is held there until another rising edge is detected (at which point the new count value is loaded into **GPTMTnR**).

Count GPTMTnR=X GPTMTnR=Z GPTMTnR=Z X Y Y

Figure 8-3. 16-Bit Input Edge Time Mode Example

8.3.3.4 16-Bit PWM Mode

Input Signal

Note: The prescaler is not available in 16-Bit PWM mode.

The GPTM supports a simple PWM generation mode. In PWM mode, the timer is configured as a down-counter with a start value (and thus period) defined by **GPTMTnILR**. In this mode, the PWM frequency and period are synchronous events and therefore guaranteed to be glitch free. PWM mode is enabled with the **GPTMTnMR** register by setting the TnAMS bit to 0x1, the TnCMR bit to 0x0, and the TnMR field to 0x2.

When software writes the \mathtt{TnEN} bit in the **GPTMCTL** register, the counter begins counting down until it reaches the 0x0000 state. On the next counter cycle, the counter reloads its start value from **GPTMTnILR** and continues counting until disabled by software clearing the \mathtt{TnEN} bit in the **GPTMCTL** register. No interrupts or status bits are asserted in PWM mode.

The output PWM signal asserts when the counter is at the value of the **GPTMTnILR** register (its start state), and is deasserted when the counter value equals the value in the **GPTM Timern Match Register (GPTMTnMATCHR)**. Software has the capability of inverting the output PWM signal by setting the TnPWML bit in the **GPTMCTL** register.

Figure 8-4 on page 312 shows how to generate an output PWM with a 1-ms period and a 66% duty cycle assuming a 50-MHz input clock and **TnPWML** =0 (duty cycle would be 33% for the **TnPWML** =1 configuration). For this example, the start value is **GPTMTnIRL**=0xC350 and the match value is **GPTMTnMATCHR**=0x411A.

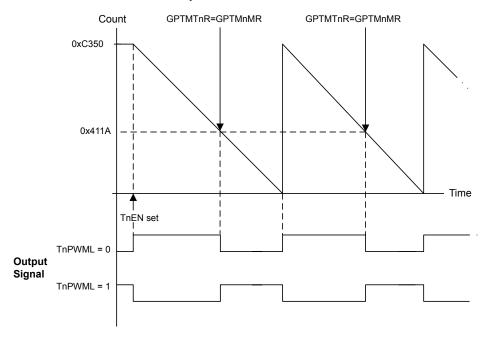


Figure 8-4. 16-Bit PWM Mode Example

8.4 Initialization and Configuration

To use the general-purpose timers, the peripheral clock must be enabled by setting the TIMERO, TIMER1, and TIMER2 bits in the **RCGC1** register.

This section shows module initialization and configuration examples for each of the supported timer modes.

8.4.1 32-Bit One-Shot/Periodic Timer Mode

The GPTM is configured for 32-bit One-Shot and Periodic modes by the following sequence:

- 1. Ensure the timer is disabled (the TAEN bit in the **GPTMCTL** register is cleared) before making any changes.
- 2. Write the **GPTM Configuration Register (GPTMCFG)** with a value of 0x0.
- 3. Set the TAMR field in the GPTM TimerA Mode Register (GPTMTAMR):
 - **a.** Write a value of 0x1 for One-Shot mode.
 - **b.** Write a value of 0x2 for Periodic mode.
- 4. Load the start value into the GPTM TimerA Interval Load Register (GPTMTAILR).
- 5. If interrupts are required, set the TATOIM bit in the GPTM Interrupt Mask Register (GPTMIMR).
- 6. Set the TAEN bit in the GPTMCTL register to enable the timer and start counting.
- 7. Poll the TATORIS bit in the **GPTMRIS** register or wait for the interrupt to be generated (if enabled). In both cases, the status flags are cleared by writing a 1 to the TATOCINT bit of the **GPTM** Interrupt Clear Register (GPTMICR).

In One-Shot mode, the timer stops counting after step 7 on page 313. To re-enable the timer, repeat the sequence. A timer configured in Periodic mode does not stop counting after it times out.

8.4.2 32-Bit Real-Time Clock (RTC) Mode

To use the RTC mode, the timer must have a 32.768-KHz input signal on an even CCP input. To enable the RTC feature, follow these steps:

- 1. Ensure the timer is disabled (the TAEN bit is cleared) before making any changes.
- 2. Write the GPTM Configuration Register (GPTMCFG) with a value of 0x1.
- 3. Write the desired match value to the GPTM TimerA Match Register (GPTMTAMATCHR).
- 4. Set/clear the RTCEN bit in the GPTM Control Register (GPTMCTL) as desired.
- 5. If interrupts are required, set the RTCIM bit in the GPTM Interrupt Mask Register (GPTMIMR).
- 6. Set the TAEN bit in the **GPTMCTL** register to enable the timer and start counting.

When the timer count equals the value in the **GPTMTAMATCHR** register, the GPTM asserts the RTCRIS bit in the **GPTMRIS** register and continues counting until Timer A is disabled or a hardware reset. The interrupt is cleared by writing the RTCCINT bit in the **GPTMICR** register.

8.4.3 16-Bit One-Shot/Periodic Timer Mode

A timer is configured for 16-bit One-Shot and Periodic modes by the following sequence:

- 1. Ensure the timer is disabled (the TnEN bit is cleared) before making any changes.
- 2. Write the GPTM Configuration Register (GPTMCFG) with a value of 0x4.

- 3. Set the TnMR field in the GPTM Timer Mode (GPTMTnMR) register:
 - **a.** Write a value of 0x1 for One-Shot mode.
 - **b.** Write a value of 0x2 for Periodic mode.
- 4. If a prescaler is to be used, write the prescale value to the **GPTM Timern Prescale Register** (**GPTMTnPR**).
- 5. Load the start value into the GPTM Timer Interval Load Register (GPTMTnILR).
- 6. If interrupts are required, set the Thtoim bit in the GPTM Interrupt Mask Register (GPTMIMR).
- 7. Set the TnEN bit in the **GPTM Control Register (GPTMCTL)** to enable the timer and start counting.
- 8. Poll the TnTORIS bit in the GPTMRIS register or wait for the interrupt to be generated (if enabled). In both cases, the status flags are cleared by writing a 1 to the TnTOCINT bit of the GPTM Interrupt Clear Register (GPTMICR).

In One-Shot mode, the timer stops counting after step 8 on page 314. To re-enable the timer, repeat the sequence. A timer configured in Periodic mode does not stop counting after it times out.

8.4.4 16-Bit Input Edge Count Mode

A timer is configured to Input Edge Count mode by the following sequence:

- 1. Ensure the timer is disabled (the TnEN bit is cleared) before making any changes.
- 2. Write the **GPTM Configuration (GPTMCFG)** register with a value of 0x4.
- 3. In the GPTM Timer Mode (GPTMTnMR) register, write the TnCMR field to 0x0 and the TnMR field to 0x3.
- **4.** Configure the type of event(s) that the timer captures by writing the Tnevent field of the **GPTM Control (GPTMCTL)** register.
- 5. Load the timer start value into the **GPTM Timern Interval Load (GPTMTnILR)** register.
- 6. Load the desired event count into the GPTM Timern Match (GPTMTnMATCHR) register.
- 7. If interrupts are required, set the CnMIM bit in the GPTM Interrupt Mask (GPTMIMR) register.
- 8. Set the TnEN bit in the **GPTMCTL** register to enable the timer and begin waiting for edge events.
- 9. Poll the CnMRIS bit in the **GPTMRIS** register or wait for the interrupt to be generated (if enabled). In both cases, the status flags are cleared by writing a 1 to the CnMCINT bit of the **GPTM** Interrupt Clear (GPTMICR) register.

In Input Edge Count Mode, the timer stops after the desired number of edge events has been detected. To re-enable the timer, ensure that the TnEN bit is cleared and repeat step 4 on page 314 through step 9 on page 314.

8.4.5 16-Bit Input Edge Timing Mode

A timer is configured to Input Edge Timing mode by the following sequence:

- 1. Ensure the timer is disabled (the TnEN bit is cleared) before making any changes.
- 2. Write the **GPTM Configuration (GPTMCFG)** register with a value of 0x4.
- 3. In the GPTM Timer Mode (GPTMTnMR) register, write the TnCMR field to 0x1 and the TnMR field to 0x3.
- **4.** Configure the type of event that the timer captures by writing the Tnevent field of the **GPTM Control (GPTMCTL)** register.
- 5. Load the timer start value into the GPTM Timern Interval Load (GPTMTnILR) register.
- 6. If interrupts are required, set the CnEIM bit in the GPTM Interrupt Mask (GPTMIMR) register.
- 7. Set the Then bit in the GPTM Control (GPTMCTL) register to enable the timer and start counting.
- 8. Poll the Cners bit in the GPTMRIS register or wait for the interrupt to be generated (if enabled). In both cases, the status flags are cleared by writing a 1 to the Cnecint bit of the GPTM Interrupt Clear (GPTMICR) register. The time at which the event happened can be obtained by reading the GPTM Timern (GPTMTnR) register.

In Input Edge Timing mode, the timer continues running after an edge event has been detected, but the timer interval can be changed at any time by writing the **GPTMTnILR** register. The change takes effect at the next cycle after the write.

8.4.6 16-Bit PWM Mode

A timer is configured to PWM mode using the following sequence:

- 1. Ensure the timer is disabled (the TnEN bit is cleared) before making any changes.
- 2. Write the **GPTM Configuration (GPTMCFG)** register with a value of 0x4.
- 3. In the **GPTM Timer Mode (GPTMTnMR)** register, set the TnAMS bit to 0x1, the TnCMR bit to 0x0. and the TnMR field to 0x2.
- **4.** Configure the output state of the PWM signal (whether or not it is inverted) in the TnPWML field of the **GPTM Control (GPTMCTL)** register.
- 5. Load the timer start value into the GPTM Timern Interval Load (GPTMTnILR) register.
- 6. Load the GPTM Timern Match (GPTMTnMATCHR) register with the desired value.
- 7. Set the TnEN bit in the **GPTM Control (GPTMCTL)** register to enable the timer and begin generation of the output PWM signal.

In PWM Timing mode, the timer continues running after the PWM signal has been generated. The PWM period can be adjusted at any time by writing the **GPTMTnILR** register, and the change takes effect at the next cycle after the write.

8.5 Register Map

Table 8-5 on page 316 lists the GPTM registers. The offset listed is a hexadecimal increment to the register's address, relative to that timer's base address:

■ Timer0: 0x4003.0000

Timer1: 0x4003.1000Timer2: 0x4003.2000

Note that the Timer module clock must be enabled before the registers can be programmed (see page 215). There must be a delay of 3 system clocks after the Timer module clock is enabled before any Timer module registers are accessed.

Table 8-5. Timers Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	GPTMCFG	R/W	0x0000.0000	GPTM Configuration	317
0x004	GPTMTAMR	R/W	0x0000.0000	GPTM TimerA Mode	318
0x008	GPTMTBMR	R/W	0x0000.0000	GPTM TimerB Mode	320
0x00C	GPTMCTL	R/W	0x0000.0000	GPTM Control	322
0x018	GPTMIMR	R/W	0x0000.0000	GPTM Interrupt Mask	325
0x01C	GPTMRIS	RO	0x0000.0000	GPTM Raw Interrupt Status	327
0x020	GPTMMIS	RO	0x0000.0000	GPTM Masked Interrupt Status	328
0x024	GPTMICR	W1C	0x0000.0000	GPTM Interrupt Clear	329
0x028	GPTMTAILR	R/W	0xFFFF.FFFF	GPTM TimerA Interval Load	331
0x02C	GPTMTBILR	R/W	0x0000.FFFF	GPTM TimerB Interval Load	332
0x030	GPTMTAMATCHR	R/W	0xFFFF.FFFF	GPTM TimerA Match	333
0x034	GPTMTBMATCHR	R/W	0x0000.FFFF	GPTM TimerB Match	334
0x038	GPTMTAPR	R/W	0x0000.0000	GPTM TimerA Prescale	335
0x03C	GPTMTBPR	R/W	0x0000.0000	GPTM TimerB Prescale	336
0x040	GPTMTAPMR	R/W	0x0000.0000	GPTM TimerA Prescale Match	337
0x044	GPTMTBPMR	R/W	0x0000.0000	GPTM TimerB Prescale Match	338
0x048	GPTMTAR	RO	0xFFFF.FFFF	GPTM TimerA	339
0x04C	GPTMTBR	RO	0x0000.FFFF	GPTM TimerB	340

8.6 Register Descriptions

The remainder of this section lists and describes the GPTM registers, in numerical order by address offset.

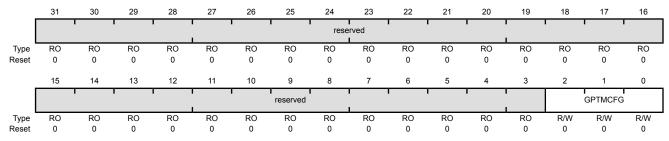
Register 1: GPTM Configuration (GPTMCFG), offset 0x000

This register configures the global operation of the GPTM module. The value written to this register determines whether the GPTM is in 32- or 16-bit mode.

GPTM Configuration (GPTMCFG)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x000

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:3	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2:0	GPTMCFG	R/W	0x0	GPTM Configuration

The GPTMCFG values are defined as follows:

Value Description

0x0 32-bit timer configuration.

0x1 32-bit real-time clock (RTC) counter configuration.

0x2 Reserved 0x3 Reserved

0x4-0x7 16-bit timer configuration, function is controlled by bits 1:0 of **GPTMTAMR** and **GPTMTBMR**.

Register 2: GPTM TimerA Mode (GPTMTAMR), offset 0x004

This register configures the GPTM based on the configuration selected in the **GPTMCFG** register. When in 16-bit PWM mode, set the TAAMS bit to 0x1, the TACMR bit to 0x0, and the TAMR field to 0x2.

GPTM TimerA Mode (GPTMTAMR)

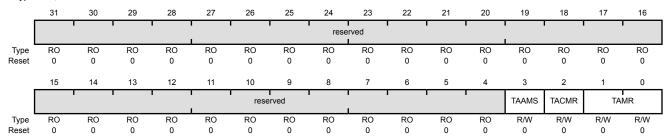
Name

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x004

Bit/Field

Type R/W, reset 0x0000.0000



31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TAAMS	R/W	0	GPTM TimerA Alternate Mode Select

Reset

Description

The TAAMS values are defined as follows:

Value Description

Capture mode is enabled.

1 PWM mode is enabled.

Note: To enable PWM mode, you must also clear the TACMR

bit and set the TAMR field to 0x2.

2 TACMR R/W 0 GPTM TimerA Capture Mode

Type

The TACMR values are defined as follows:

Value Description

D Edge-Count mode

1 Edge-Time mode

Bit/Field	Name	Type	Reset	Description
1:0	TAMR	R/W	0x0	GPTM TimerA Mode
				The TAMR values are defined as follows:
				Value Description
				0x0 Reserved
				0x1 One-Shot Timer mode
				0x2 Periodic Timer mode
				0x3 Capture mode
				The Timer mode is based on the timer configuration defined by bits 2:0 in the GPTMCFG register (16-or 32-bit).
				In 16-bit timer configuration, ${\tt TAMR}$ controls the 16-bit timer modes for TimerA.
				In 32-bit timer configuration, this register controls the mode and the contents of GPTMTBMR are ignored.

Register 3: GPTM TimerB Mode (GPTMTBMR), offset 0x008

This register configures the GPTM based on the configuration selected in the **GPTMCFG** register. When in 16-bit PWM mode, set the TBAMS bit to 0x1, the TBCMR bit to 0x0, and the TBMR field to 0x2.

GPTM TimerB Mode (GPTMTBMR)

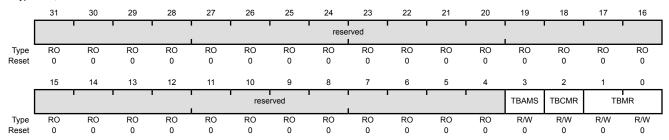
Name

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x008

Bit/Field

Type R/W, reset 0x0000.0000



31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TBAMS	R/W	0	GPTM TimerB Alternate Mode Select

Reset

Description

The TBAMS values are defined as follows:

Value Description

Capture mode is enabled.

1 PWM mode is enabled.

Note: To enable PWM mode, you must also clear the TBCMR

bit and set the $\ensuremath{\mathtt{TBMR}}$ field to 0x2.

2 TBCMR R/W 0 GPTM TimerB Capture Mode

Type

The TBCMR values are defined as follows:

Value Description

0 Edge-Count mode

1 Edge-Time mode

Bit/Field	Name	Type	Reset	Description
1:0	TBMR	R/W	0x0	GPTM TimerB Mode The TBMR values are defined as follows: Value Description
				0x0 Reserved 0x1 One-Shot Timer mode 0x2 Periodic Timer mode 0x3 Capture mode
				The timer mode is based on the timer configuration defined by bits 2:0 in the GPTMCFG register. In 16-bit timer configuration, these bits control the 16-bit timer modes for TimerB. In 32-bit timer configuration, this register's contents are ignored and GPTMTAMR is used.

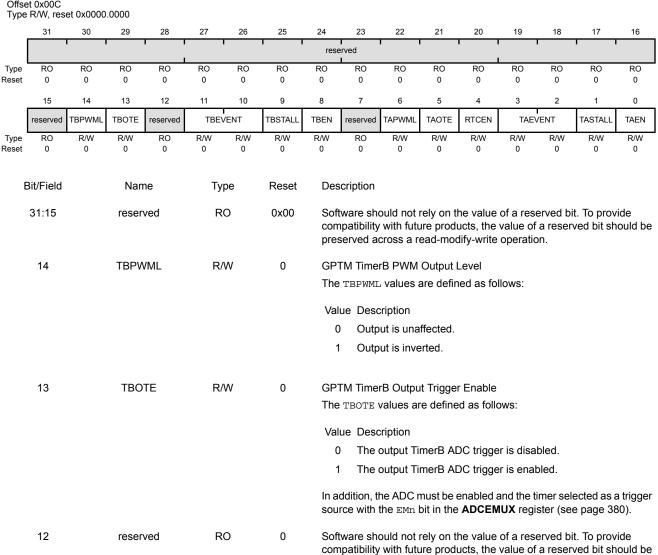
Register 4: GPTM Control (GPTMCTL), offset 0x00C

This register is used alongside the **GPTMCFG** and **GMTMTnMR** registers to fine-tune the timer configuration, and to enable other features such as timer stall and the output trigger. The output trigger can be used to initiate transfers on the ADC module.

GPTM Control (GPTMCTL)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x00C



preserved across a read-modify-write operation.

Bit/Field	Name	Туре	Reset	Description
11:10	TBEVENT	R/W	0x0	GPTM TimerB Event Mode
				The TBEVENT values are defined as follows:
				Value Description
				0x0 Positive edge
				0x1 Negative edge
				0x2 Reserved
				0x3 Both edges
9	TBSTALL	R/W	0	GPTM Timer B Stall Enable
				The TBSTALL values are defined as follows:
				Value Description
				Timer B continues counting while the processor is halted by the debugger.
				1 Timer B freezes counting while the processor is halted by the debugger.
				If the processor is executing normally, the ${\tt TBSTALL}$ bit is ignored.
8	TBEN	R/W	0	GPTM TimerB Enable
				The TBEN values are defined as follows:
				Value Description
				0 TimerB is disabled.
				1 TimerB is enabled and begins counting or the capture logic is enabled based on the GPTMCFG register.
7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	TAPWML	R/W	0	GPTM TimerA PWM Output Level
				The TAPWML values are defined as follows:
				Value Description
				0 Output is unaffected.
				1 Output is inverted.
5	TAOTE	R/W	0	GPTM TimerA Output Trigger Enable
				The TAOTE values are defined as follows:
				Value Description
				0 The output TimerA ADC trigger is disabled.
				1 The output TimerA ADC trigger is enabled.

July 15, 2014 323

In addition, the ADC must be enabled and the timer selected as a trigger source with the ${\tt EMn}$ bit in the ADCEMUX register (see page 380).

Bit/Field	Name	Туре	Reset	Description
4	RTCEN	R/W	0	GPTM RTC Enable
				The RTCEN values are defined as follows:
				Value Description
				0 RTC counting is disabled.
				1 RTC counting is enabled.
3:2	TAEVENT	R/W	0x0	GPTM TimerA Event Mode
				The TAEVENT values are defined as follows:
				Value Description
				0x0 Positive edge
				0x1 Negative edge
				0x2 Reserved
				0x3 Both edges
1	TASTALL	R/W	0	GPTM Timer A Stall Enable
				The TASTALL values are defined as follows:
				Value Description
				Timer A continues counting while the processor is halted by the debugger.
				Timer A freezes counting while the processor is halted by the debugger.
				If the processor is executing normally, the ${\tt TASTALL}$ bit is ignored.
0	TAEN	R/W	0	GPTM TimerA Enable
				The TAEN values are defined as follows:
				Value Description
				0 TimerA is disabled.

TimerA is enabled and begins counting or the capture logic is enabled based on the **GPTMCFG** register.

Register 5: GPTM Interrupt Mask (GPTMIMR), offset 0x018

This register allows software to enable/disable GPTM controller-level interrupts. Writing a 1 enables the interrupt, while writing a 0 disables it.

GPTM Interrupt Mask (GPTMIMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x018

Type R/W, reset 0x0000.0000

8

7:4

TBTOIM

reserved

R/W

RO

0

0

GPTM TimerB Time-Out Interrupt Mask
The TBTOIM values are defined as follows:

Interrupt is disabled.

Interrupt is enabled.

Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be

preserved across a read-modify-write operation.

Value Description

30

								rese	rvea							
Type L	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		'	reserved			CBEIM	СВМІМ	твтоім		rese	rved	'	RTCIM	CAEIM	CAMIM	TATOIM
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	R/W 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	R/W 0	R/W 0
Reset	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
Bit/Field		Nam	10	Type Re			Description									
L	olivi i c iu		ivaiii	ic	ıy	þe	Reset	Desi	cription							
	31:11		reserv	/ed	R	0	0x00				,	he value				
										•	ucts, the			ed bit sh	ould be	
								pres	erved ad	ross a r	ead-mod	dify-write	operation	ori.		
10		CBEI	IM	R/	W	0	GPT	M Captu	ıreB Eve	ent Interr	rupt Mas	k				
								The	CBEIM V	alues ar	e define	ed as follo	ows:			
								\ /=I.	D							
									ue Desc	•						
								0 Interrupt is disabled.1 Interrupt is enabled.								
								1	Interr	upt is er	nabled.					
	9		CBM	IM	R/	W	0	GPTM CaptureB Match Interrupt Mask								
								The	CBMIM V	alues ar	e define	ed as follo	ows:			
								Valu	ue Desc	ription						
								0	Interr	upt is di	sabled.					
								1		upt is er						
								·								

Bit/Field	Name	Туре	Reset	Description
3	RTCIM	R/W	0	GPTM RTC Interrupt Mask The RTCIM values are defined as follows:
				Value Description 0 Interrupt is disabled. 1 Interrupt is enabled.
2	CAEIM	R/W	0	GPTM CaptureA Event Interrupt Mask The CAEIM values are defined as follows:
				Value Description 0 Interrupt is disabled. 1 Interrupt is enabled.
1	CAMIM	R/W	0	GPTM CaptureA Match Interrupt Mask The CAMIM values are defined as follows:
				Value Description 0 Interrupt is disabled. 1 Interrupt is enabled.
0	TATOIM	R/W	0	GPTM TimerA Time-Out Interrupt Mask The TATOIM values are defined as follows:
				Value Description 0 Interrupt is disabled. 1 Interrupt is enabled.

Register 6: GPTM Raw Interrupt Status (GPTMRIS), offset 0x01C

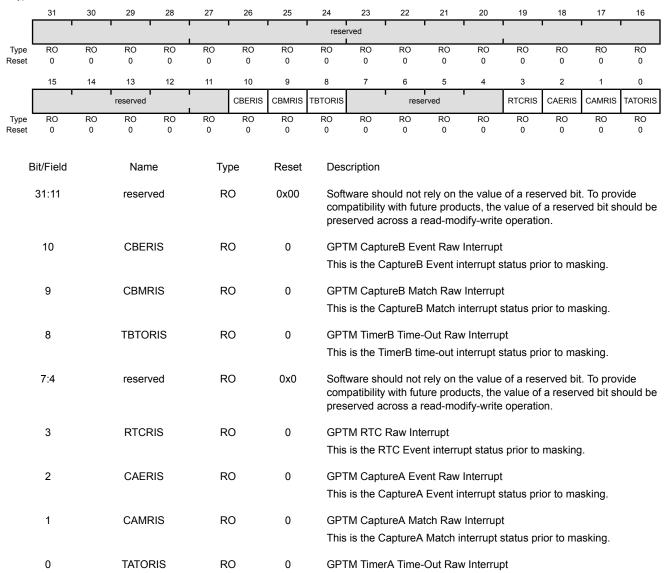
This register shows the state of the GPTM's internal interrupt signal. These bits are set whether or not the interrupt is masked in the **GPTMIMR** register. Each bit can be cleared by writing a 1 to its corresponding bit in **GPTMICR**.

GPTM Raw Interrupt Status (GPTMRIS)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x01C

Type RO, reset 0x0000.0000



This the TimerA time-out interrupt status prior to masking.

Register 7: GPTM Masked Interrupt Status (GPTMMIS), offset 0x020

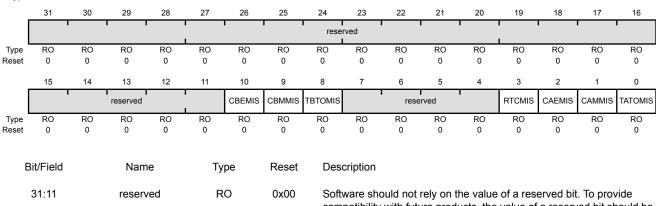
This register show the state of the GPTM's controller-level interrupt. If an interrupt is unmasked in **GPTMIMR**, and there is an event that causes the interrupt to be asserted, the corresponding bit is set in this register. All bits are cleared by writing a 1 to the corresponding bit in **GPTMICR**.

GPTM Masked Interrupt Status (GPTMMIS)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x020

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:11	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
10	CBEMIS	RO	0	GPTM CaptureB Event Masked Interrupt This is the CaptureB event interrupt status after masking.
9	CBMMIS	RO	0	GPTM CaptureB Match Masked Interrupt This is the CaptureB match interrupt status after masking.
8	TBTOMIS	RO	0	GPTM TimerB Time-Out Masked Interrupt This is the TimerB time-out interrupt status after masking.
7:4	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	RTCMIS	RO	0	GPTM RTC Masked Interrupt This is the RTC event interrupt status after masking.
2	CAEMIS	RO	0	GPTM CaptureA Event Masked Interrupt This is the CaptureA event interrupt status after masking.
1	CAMMIS	RO	0	GPTM CaptureA Match Masked Interrupt This is the CaptureA match interrupt status after masking.
0	TATOMIS	RO	0	GPTM TimerA Time-Out Masked Interrupt This is the TimerA time-out interrupt status after masking.

Register 8: GPTM Interrupt Clear (GPTMICR), offset 0x024

This register is used to clear the status bits in the GPTMRIS and GPTMMIS registers. Writing a 1 to a bit clears the corresponding bit in the **GPTMRIS** and **GPTMMIS** registers.

GPTM Interrupt Clear (GPTMICR)

Timer0 base: 0x4003.0000
Timer1 base: 0x4003.1000
Timer2 base: 0x4003.2000
Offset 0x024
Type W1C, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	'				' '	•	!	rese	rved	•	•	•		•	•	•
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
recoct	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	10		reserved	12		CBECINT	CBMCINT	1		ı	erved		RTCCINT	1	CAMCINT	TATOCINT
Туре	RO	RO	RO	RO	RO	W1C	W1C	W1C	RO	RO	RO	RO	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	Bit/Field		Name		Ty	Type Reset		Description								
	31:11					•		C-#		عمم اماريم		احداد		ائط لمصرسم	. Ta	مامان
	31:11		reserved		RO 0		0x00				•		e of a res e value of		•	
								pres	served a	cross a r	ead-mo	dify-writ	e operation	on.		
	10		CBECINT			1C	0	GPTM CaptureB Event Interrupt Clear								
								The	CBECIN	T value	s are det	fined as	follows:			
								Val	ue Desc	cription						
								0	The	interrupt	is unaffe	ected.				
							1	The	interrupt	is cleare	ed.					
	9		CBMCINT W1C			1C	0	GP1	GPTM CaptureB Match Interrupt Clear							
								The CBMCINT values are defined as follows:								
								Val	ue Desc	cription						
								0	The	interrupt	is unaffe	ected.				
								1	The	interrupt	is cleare	ed.				
	8		ТВТОС	INT	W	1C	0	GP1	⊺M Time	rB Time-	Out Inte	rrupt Cl	ear			
								The	TBTOCI	NT valu	es are d	efined a	s follows	:		
								Val	ue Desc	cription						
								0	The	interrupt	is unaffe	ected.				
								1	The	interrupt	is cleare	ed.				
	7:4		reserv	red	R	.0	0x0	com	patibility	with fut	ure prod	ucts, the	e of a res e value of e operation	a reserv		

Bit/Field	Name	Туре	Reset	Description
3	RTCCINT	W1C	0	GPTM RTC Interrupt Clear The RTCCINT values are defined as follows:
				Value Description O The interrupt is unaffected. 1 The interrupt is cleared.
2	CAECINT	W1C	0	GPTM CaptureA Event Interrupt Clear The CAECINT values are defined as follows: Value Description 0 The interrupt is unaffected.
				1 The interrupt is cleared.
1	CAMCINT	W1C	0	GPTM CaptureA Match Interrupt Clear The CAMCINT values are defined as follows:
				Value Description O The interrupt is unaffected. The interrupt is cleared.
0	TATOCINT	W1C	0	GPTM TimerA Time-Out Interrupt Clear The TATOCINT values are defined as follows:
				Value Description 0 The interrupt is unaffected. 1 The interrupt is cleared.

Register 9: GPTM TimerA Interval Load (GPTMTAILR), offset 0x028

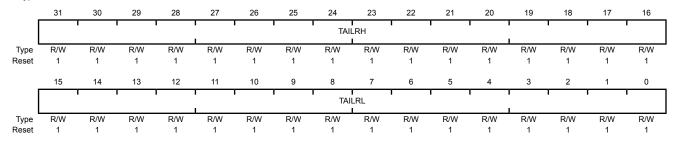
This register is used to load the starting count value into the timer. When GPTM is configured to one of the 32-bit modes, **GPTMTAILR** appears as a 32-bit register (the upper 16-bits correspond to the contents of the **GPTM TimerB Interval Load (GPTMTBILR)** register). In 16-bit mode, the upper 16 bits of this register read as 0s and have no effect on the state of **GPTMTBILR**.

GPTM TimerA Interval Load (GPTMTAILR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x028

Type R/W, reset 0xFFF.FFF



Bit/Field	Name	Туре	Reset	Description
31:16	TAILRH	R/W	0xFFFF	GPTM TimerA Interval Load Register High
				When configured for 32-bit mode via the GPTMCFG register, the GPTM TimerB Interval Load (GPTMTBILR) register loads this value on a write. A read returns the current value of GPTMTBILR .
				In 16-bit mode, this field reads as 0 and does not have an effect on the state of GPTMTBILR .
15:0	TAILRL	R/W	0xFFFF	GPTM TimerA Interval Load Register Low

For both 16- and 32-bit modes, writing this field loads the counter for TimerA. A read returns the current value of **GPTMTAILR**.

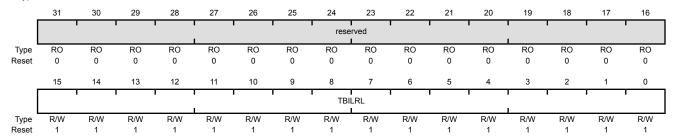
Register 10: GPTM TimerB Interval Load (GPTMTBILR), offset 0x02C

This register is used to load the starting count value into TimerB. When the GPTM is configured to a 32-bit mode, **GPTMTBILR** returns the current value of TimerB and ignores writes.

GPTM TimerB Interval Load (GPTMTBILR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x02C

Type R/W, reset 0x0000.FFFF



Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	TBILRL	R/W	0xFFFF	GPTM TimerB Interval Load Register

When the GPTM is not configured as a 32-bit timer, a write to this field updates **GPTMTBILR**. In 32-bit mode, writes are ignored, and reads return the current value of **GPTMTBILR**.

Register 11: GPTM TimerA Match (GPTMTAMATCHR), offset 0x030

This register is used in 32-bit Real-Time Clock mode and 16-bit PWM and Input Edge Count modes.

GPTM TimerA Match (GPTMTAMATCHR)

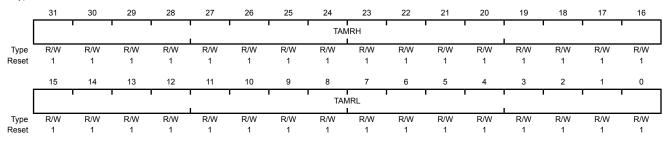
Name

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x030

Bit/Field

Type R/W, reset 0xFFFF.FFFF



Description

31:16	TAMRH	R/W	0xFFFF	GPTM TimerA Match Register High
				When configured for 22 hit Dool Time Cla

Reset

When configured for 32-bit Real-Time Clock (RTC) mode via the **GPTMCFG** register, this value is compared to the upper half of **GPTMTAR**, to determine match events.

In 16-bit mode, this field reads as 0 and does not have an effect on the state of **GPTMTBMATCHR**.

15:0 TAMRL R/W 0xFFFF GPTM TimerA Match Register Low

Type

When configured for 32-bit Real-Time Clock (RTC) mode via the **GPTMCFG** register, this value is compared to the lower half of **GPTMTAR**, to determine match events.

When configured for PWM mode, this value along with **GPTMTAILR**, determines the duty cycle of the output PWM signal.

When configured for Edge Count mode, this value along with **GPTMTAILR**, determines how many edge events are counted. The total number of edge events counted is equal to the value in **GPTMTAILR** minus this value.

Register 12: GPTM TimerB Match (GPTMTBMATCHR), offset 0x034

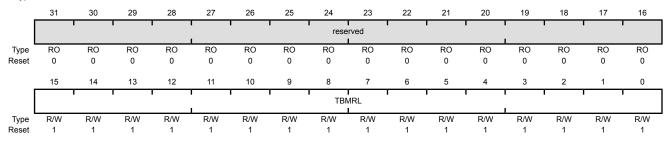
This register is used in 16-bit PWM and Input Edge Count modes.

GPTM TimerB Match (GPTMTBMATCHR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x034

Type R/W, reset 0x0000.FFFF



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	TBMRL	R/W	0xFFFF	GPTM TimerB Match Register Low

When configured for PWM mode, this value along with **GPTMTBILR**, determines the duty cycle of the output PWM signal.

When configured for Edge Count mode, this value along with **GPTMTBILR**, determines how many edge events are counted. The total number of edge events counted is equal to the value in **GPTMTBILR** minus this value.

Register 13: GPTM TimerA Prescale (GPTMTAPR), offset 0x038

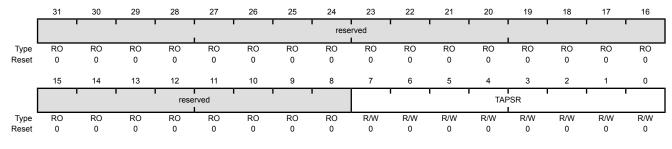
This register allows software to extend the range of the 16-bit timers when operating in one-shot or periodic mode.

GPTM TimerA Prescale (GPTMTAPR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x038

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	TAPSR	R/W	0x00	GPTM TimerA Prescale

The register loads this value on a write. A read returns the current value of the register.

Refer to Table 8-4 on page 309 for more details and an example.

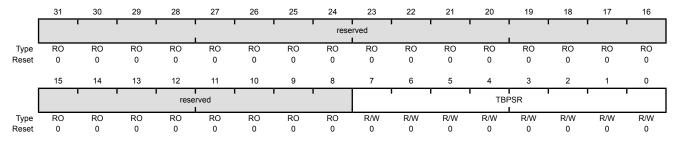
Register 14: GPTM TimerB Prescale (GPTMTBPR), offset 0x03C

This register allows software to extend the range of the 16-bit timers when operating in one-shot or periodic mode.

GPTM TimerB Prescale (GPTMTBPR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x03C

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	TBPSR	R/W	0x00	GPTM TimerB Prescale

The register loads this value on a write. A read returns the current value of this register.

Refer to Table 8-4 on page 309 for more details and an example.

Register 15: GPTM TimerA Prescale Match (GPTMTAPMR), offset 0x040

This register effectively extends the range of GPTMTAMATCHR to 24 bits when operating in 16-bit one-shot or periodic mode.

GPTM TimerA Prescale Match (GPTMTAPMR)

TAPSMR

R/W

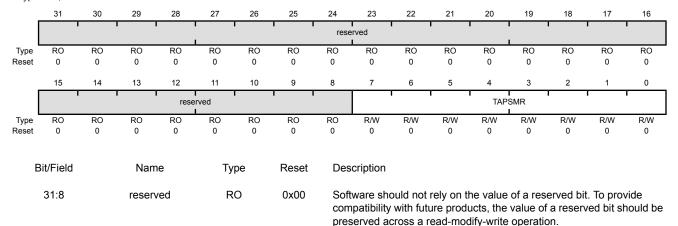
0x00

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x040

7:0

Type R/W, reset 0x0000.0000



GPTM TimerA Prescale Match

This value is used alongside **GPTMTAMATCHR** to detect timer match events while using a prescaler.

Register 16: GPTM TimerB Prescale Match (GPTMTBPMR), offset 0x044

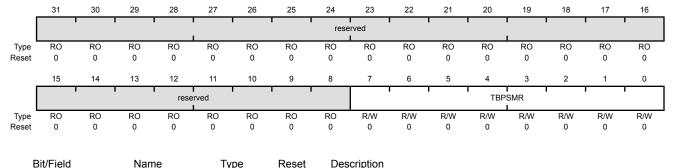
This register effectively extends the range of GPTMTBMATCHR to 24 bits when operating in 16-bit one-shot or periodic mode.

GPTM TimerB Prescale Match (GPTMTBPMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000

Offset 0x044

Type R/W, reset 0x0000.0000



Dit/Tield	Name	Туре	Neset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	TBPSMR	R/W	0x00	GPTM TimerB Prescale Match

This value is used alongside **GPTMTBMATCHR** to detect timer match events while using a prescaler.

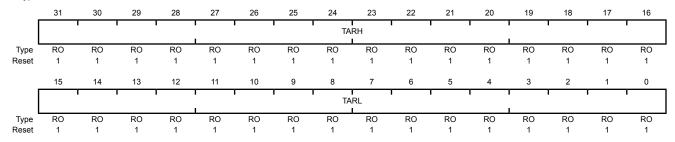
Register 17: GPTM TimerA (GPTMTAR), offset 0x048

This register shows the current value of the TimerA counter in all cases except for Input Edge Count mode. When in this mode, this register contains the number of edges that have occurred.

GPTM TimerA (GPTMTAR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x048

Offset 0x048
Type RO, reset 0xFFF.FFF



Bit/Field	Name	Type	Reset	Description
31:16	TARH	RO	0xFFFF	GPTM TimerA Register High If the GPTMCFG is in a 32-bit mode, TimerB value is read. If the GPTMCFG is in a 16-bit mode, this is read as zero.
15:0	TARL	RO	0xFFFF	GPTM TimerA Register Low

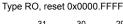
A read returns the current value of the **GPTM TimerA Count Register**, except in Input Edge-Count mode, when it returns the number of edges that have occurred.

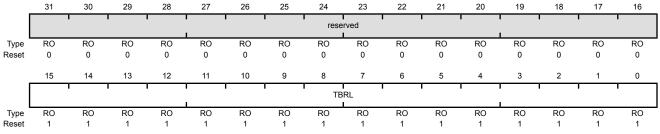
Register 18: GPTM TimerB (GPTMTBR), offset 0x04C

This register shows the current value of the TimerB counter in all cases except for Input Edge Count mode. When in this mode, this register contains the number of edges that have occurred.

GPTM TimerB (GPTMTBR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Offset 0x04C Type RO, reset 0x0000.FFFF





Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	TBRL	RO	0xFFFF	GPTM TimerB

A read returns the current value of the GPTM TimerB Count Register, except in Input Edge-Count mode, when it returns the number of edges that have occurred.

9 Watchdog Timer

A watchdog timer can generate nonmaskable interrupts (NMIs) or a reset when a time-out value is reached. The watchdog timer is used to regain control when a system has failed due to a software error or due to the failure of an external device to respond in the expected way.

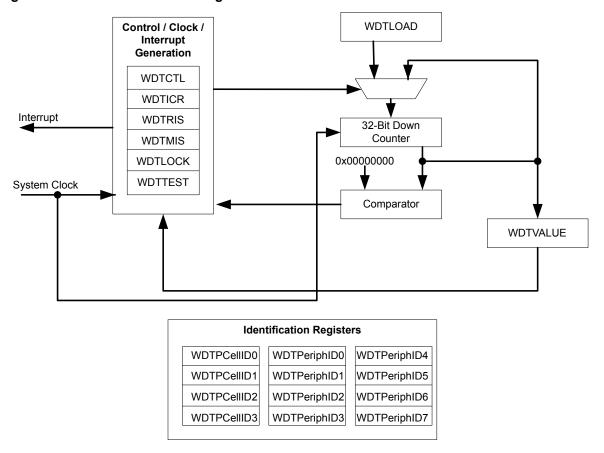
The Stellaris® Watchdog Timer module has the following features:

- 32-bit down counter with a programmable load register
- Separate watchdog clock with an enable
- Programmable interrupt generation logic with interrupt masking
- Lock register protection from runaway software
- Reset generation logic with an enable/disable
- User-enabled stalling when the controller asserts the CPU Halt flag during debug

The Watchdog Timer can be configured to generate an interrupt to the controller on its first time-out, and to generate a reset signal on its second time-out. Once the Watchdog Timer has been configured, the lock register can be written to prevent the timer configuration from being inadvertently altered.

9.1 Block Diagram

Figure 9-1. WDT Module Block Diagram



9.2 Functional Description

The Watchdog Timer module generates the first time-out signal when the 32-bit counter reaches the zero state after being enabled; enabling the counter also enables the watchdog timer interrupt. After the first time-out event, the 32-bit counter is re-loaded with the value of the **Watchdog Timer Load (WDTLOAD)** register, and the timer resumes counting down from that value. Once the Watchdog Timer has been configured, the **Watchdog Timer Lock (WDTLOCK)** register is written, which prevents the timer configuration from being inadvertently altered by software.

If the timer counts down to its zero state again before the first time-out interrupt is cleared, and the reset signal has been enabled (via the WatchdogResetEnable function), the Watchdog timer asserts its reset signal to the system. If the interrupt is cleared before the 32-bit counter reaches its second time-out, the 32-bit counter is loaded with the value in the **WDTLOAD** register, and counting resumes from that value.

If **WDTLOAD** is written with a new value while the Watchdog Timer counter is counting, then the counter is loaded with the new value and continues counting.

Writing to **WDTLOAD** does not clear an active interrupt. An interrupt must be specifically cleared by writing to the **Watchdog Interrupt Clear (WDTICR)** register.

The Watchdog module interrupt and reset generation can be enabled or disabled as required. When the interrupt is re-enabled, the 32-bit counter is preloaded with the load register value and not its last state.

9.3 Initialization and Configuration

To use the WDT, its peripheral clock must be enabled by setting the WDT bit in the **RCGC0** register. The Watchdog Timer is configured using the following sequence:

- 1. Load the WDTLOAD register with the desired timer load value.
- 2. If the Watchdog is configured to trigger system resets, set the RESEN bit in the WDTCTL register.
- 3. Set the INTEN bit in the WDTCTL register to enable the Watchdog and lock the control register.

If software requires that all of the watchdog registers are locked, the Watchdog Timer module can be fully locked by writing any value to the **WDTLOCK** register. To unlock the Watchdog Timer, write a value of 0x1ACC.E551.

9.4 Register Map

Table 9-1 on page 343 lists the Watchdog registers. The offset listed is a hexadecimal increment to the register's address, relative to the Watchdog Timer base address of 0x4000.0000.

Table 9-1. Watchdog Timer Register Map

Offset	Name	Type	Reset	Description	See page
0x000	WDTLOAD	R/W	0xFFFF.FFFF	Watchdog Load	345
0x004	WDTVALUE	RO	0xFFFF.FFFF	Watchdog Value	346
0x008	WDTCTL	R/W	0x0000.0000	Watchdog Control	347
0x00C	WDTICR	WO	-	Watchdog Interrupt Clear	348
0x010	WDTRIS	RO	0x0000.0000	Watchdog Raw Interrupt Status	349
0x014	WDTMIS	RO	0x0000.0000	Watchdog Masked Interrupt Status	350
0x418	WDTTEST	R/W	0x0000.0000	Watchdog Test	351
0xC00	WDTLOCK	R/W	0x0000.0000	Watchdog Lock	352
0xFD0	WDTPeriphID4	RO	0x0000.0000	Watchdog Peripheral Identification 4	353
0xFD4	WDTPeriphID5	RO	0x0000.0000	Watchdog Peripheral Identification 5	354
0xFD8	WDTPeriphID6	RO	0x0000.0000	Watchdog Peripheral Identification 6	355
0xFDC	WDTPeriphID7	RO	0x0000.0000	Watchdog Peripheral Identification 7	356
0xFE0	WDTPeriphID0	RO	0x0000.0005	Watchdog Peripheral Identification 0	357
0xFE4	WDTPeriphID1	RO	0x0000.0018	Watchdog Peripheral Identification 1	358
0xFE8	WDTPeriphID2	RO	0x0000.0018	Watchdog Peripheral Identification 2	359

Table 9-1. Watchdog Timer Register Map (continued)

Offset	Name	Type	Reset	Description	See page
0xFEC	WDTPeriphID3	RO	0x0000.0001	Watchdog Peripheral Identification 3	360
0xFF0	WDTPCellID0	RO	0x0000.000D	Watchdog PrimeCell Identification 0	361
0xFF4	WDTPCellID1	RO	0x0000.00F0	Watchdog PrimeCell Identification 1	362
0xFF8	WDTPCellID2	RO	0x0000.0005	Watchdog PrimeCell Identification 2	363
0xFFC	WDTPCellID3	RO	0x0000.00B1	Watchdog PrimeCell Identification 3	364

9.5 Register Descriptions

The remainder of this section lists and describes the WDT registers, in numerical order by address offset.

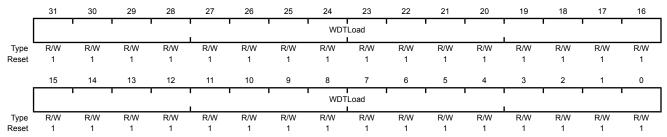
Register 1: Watchdog Load (WDTLOAD), offset 0x000

This register is the 32-bit interval value used by the 32-bit counter. When this register is written, the value is immediately loaded and the counter restarts counting down from the new value. If the WDTLOAD register is loaded with 0x0000.0000, an interrupt is immediately generated.

Watchdog Load (WDTLOAD)

Base 0x4000.0000

Offset 0x000 Type R/W, reset 0xFFFF.FFF



Bit/Field Name Reset Description Type 31:0 WDTLoad R/W 0xFFFF.FFFF Watchdog Load Value

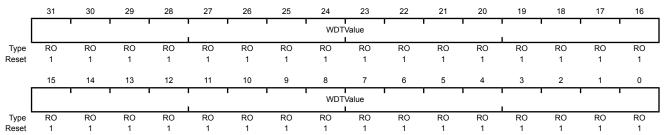
Register 2: Watchdog Value (WDTVALUE), offset 0x004

This register contains the current count value of the timer.

Watchdog Value (WDTVALUE)

Base 0x4000.0000 Offset 0x004

Type RO, reset 0xFFFF.FFF



Bit/Field Name Type Reset Description

31:0 WDTValue RO 0xFFF.FFFF Watchdog Value

Current value of the 32-bit down counter.

be

Register 3: Watchdog Control (WDTCTL), offset 0x008

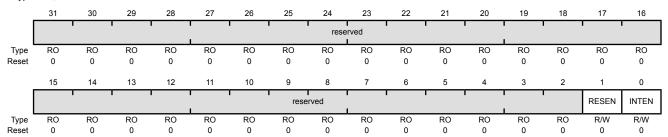
This register is the watchdog control register. The watchdog timer can be configured to generate a reset signal (on second time-out) or an interrupt on time-out.

When the watchdog interrupt has been enabled, all subsequent writes to the control register are ignored. The only mechanism that can re-enable writes is a hardware reset.

Watchdog Control (WDTCTL)

Base 0x4000.0000 Offset 0x008

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	RESEN	R/W	0	Watchdog Reset Enable The RESEN values are defined as follows:
				Value Description 0 Disabled. 1 Enable the Watchdog module reset output.
0	INTEN	R/W	0	Watchdog Interrupt Enable The INTEN values are defined as follows:

Value Description

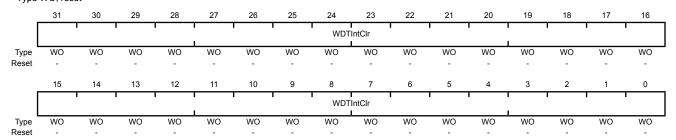
- 0 Interrupt event disabled (once this bit is set, it can only be cleared by a hardware reset).
- 1 Interrupt event enabled. Once enabled, all writes are ignored.

Register 4: Watchdog Interrupt Clear (WDTICR), offset 0x00C

This register is the interrupt clear register. A write of any value to this register clears the Watchdog interrupt and reloads the 32-bit counter from the **WDTLOAD** register. Value for a read or reset is indeterminate.

Watchdog Interrupt Clear (WDTICR)

Base 0x4000.0000 Offset 0x00C Type WO, reset -



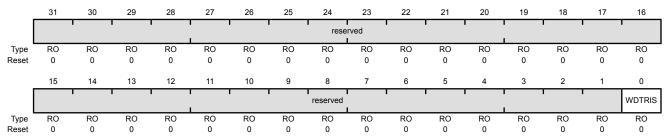
Bit/Field	Name	Type	Reset	Description
31:0	WDTIntClr	WO	-	Watchdog Interrupt Clear

Register 5: Watchdog Raw Interrupt Status (WDTRIS), offset 0x010

This register is the raw interrupt status register. Watchdog interrupt events can be monitored via this register if the controller interrupt is masked.

Watchdog Raw Interrupt Status (WDTRIS)

Base 0x4000.0000 Offset 0x010 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	WDTRIS	RO	0	Watchdog Raw Interrupt Status

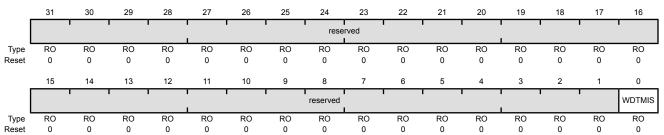
Gives the raw interrupt state (prior to masking) of WDTINTR.

Register 6: Watchdog Masked Interrupt Status (WDTMIS), offset 0x014

This register is the masked interrupt status register. The value of this register is the logical AND of the raw interrupt bit and the Watchdog interrupt enable bit.

Watchdog Masked Interrupt Status (WDTMIS)

Base 0x4000.0000 Offset 0x014 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	WDTMIS	RO	0	Watchdog Masked Interrupt Status

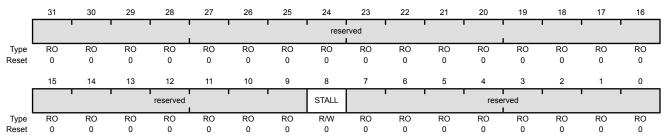
Gives the masked interrupt state (after masking) of the WDTINTR interrupt.

Register 7: Watchdog Test (WDTTEST), offset 0x418

This register provides user-enabled stalling when the microcontroller asserts the CPU halt flag during debug.

Watchdog Test (WDTTEST)

Base 0x4000.0000 Offset 0x418 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:9	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
8	STALL	R/W	0	Watchdog Stall Enable
				When set to 1, if the Stellaris microcontroller is stopped with a debugger, the watchdog timer stops counting. Once the microcontroller is restarted, the watchdog timer resumes counting.
7:0	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

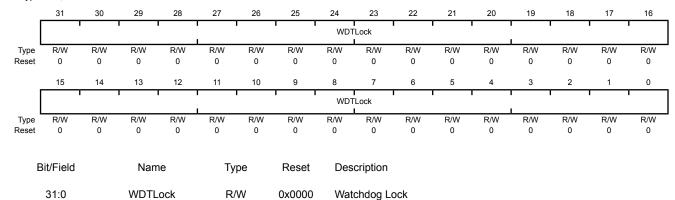
Register 8: Watchdog Lock (WDTLOCK), offset 0xC00

Writing 0x1ACC.E551 to the **WDTLOCK** register enables write access to all other registers. Writing any other value to the **WDTLOCK** register re-enables the locked state for register writes to all the other registers. Reading the **WDTLOCK** register returns the lock status rather than the 32-bit value written. Therefore, when write accesses are disabled, reading the **WDTLOCK** register returns 0x0000.0001 (when locked; otherwise, the returned value is 0x0000.0000 (unlocked)).

Watchdog Lock (WDTLOCK)

Base 0x4000.0000 Offset 0xC00

Type R/W, reset 0x0000.0000



A write of the value 0x1ACC.E551 unlocks the watchdog registers for write access. A write of any other value reapplies the lock, preventing any register updates.

A read of this register returns the following values:

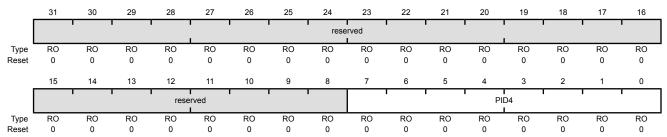
Value Description 0x0000.0001 Locked 0x0000.0000 Unlocked

Register 9: Watchdog Peripheral Identification 4 (WDTPeriphID4), offset 0xFD0

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 4 (WDTPeriphID4)

Base 0x4000.0000 Offset 0xFD0 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x00	WDT Peripheral ID Register[7:0]

Register 10: Watchdog Peripheral Identification 5 (WDTPeriphID5), offset 0xFD4

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

WDT Peripheral ID Register[15:8]

Watchdog Peripheral Identification 5 (WDTPeriphID5)

PID5

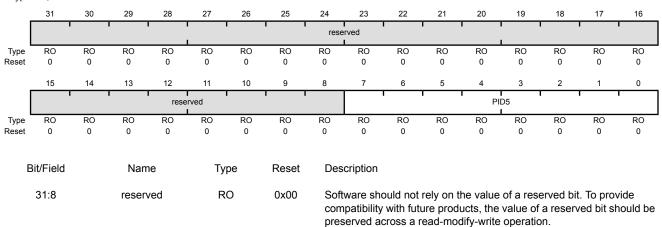
RO

0x00

Base 0x4000.0000

7:0

Offset 0xFD4
Type RO, reset 0x0000.0000

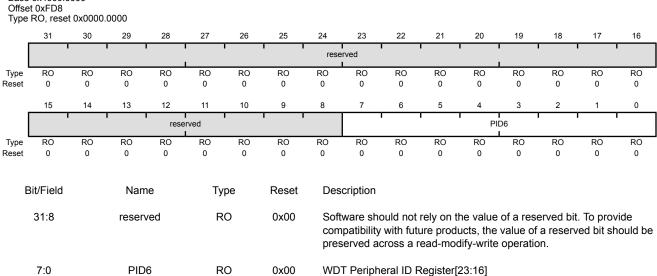


Register 11: Watchdog Peripheral Identification 6 (WDTPeriphID6), offset 0xFD8

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 6 (WDTPeriphID6)

Base 0x4000.0000



Register 12: Watchdog Peripheral Identification 7 (WDTPeriphID7), offset 0xFDC

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 7 (WDTPeriphID7)

PID7

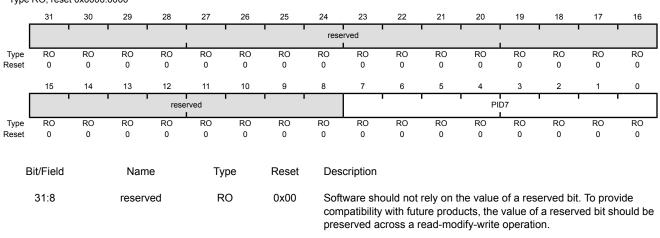
RO

0x00

Base 0x4000.0000

7:0

Offset 0xFDC Type RO, reset 0x0000.0000



WDT Peripheral ID Register[31:24]

Register 13: Watchdog Peripheral Identification 0 (WDTPeriphID0), offset 0xFE0

The **WDTPeriphIDn** registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral ID Register[7:0]

Watchdog Peripheral Identification 0 (WDTPeriphID0)

PID0

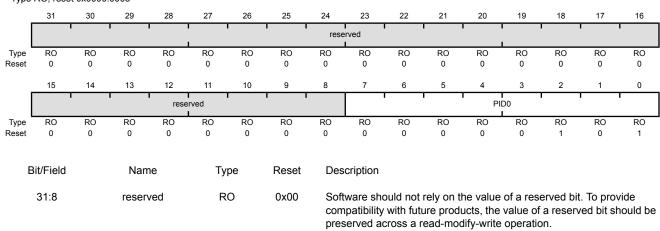
RO

0x05

Base 0x4000.0000

7:0

Offset 0xFE0
Type RO, reset 0x0000.0005



Register 14: Watchdog Peripheral Identification 1 (WDTPeriphID1), offset 0xFE4

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 1 (WDTPeriphID1)

PID1

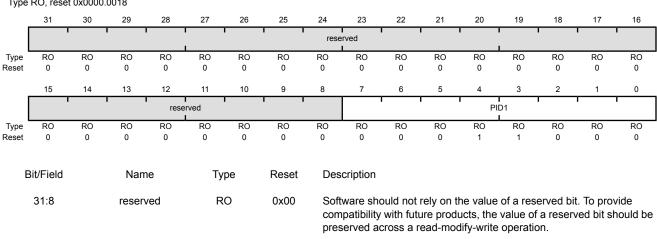
RO

0x18

Base 0x4000.0000

7:0

Offset 0xFE4
Type RO, reset 0x0000.0018



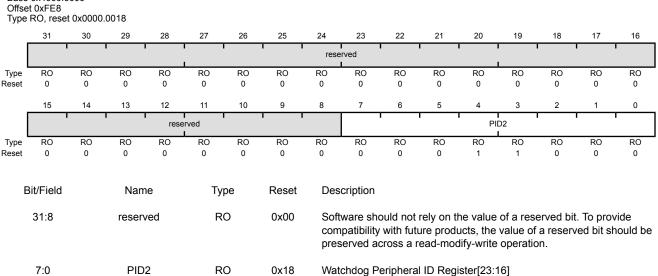
Watchdog Peripheral ID Register[15:8]

Register 15: Watchdog Peripheral Identification 2 (WDTPeriphID2), offset 0xFE8

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 2 (WDTPeriphID2)

Base 0x4000.0000



Register 16: Watchdog Peripheral Identification 3 (WDTPeriphID3), offset 0xFEC

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral ID Register[31:24]

Watchdog Peripheral Identification 3 (WDTPeriphID3)

PID3

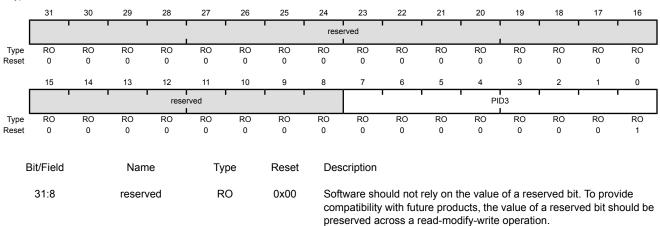
RO

0x01

Base 0x4000.0000

7:0

Offset 0xFEC Type RO, reset 0x0000.0001

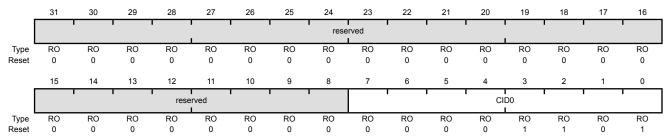


Register 17: Watchdog PrimeCell Identification 0 (WDTPCellID0), offset 0xFF0

The WDTPCellIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog PrimeCell Identification 0 (WDTPCellID0)

Base 0x4000.0000 Offset 0xFF0 Type RO, reset 0x0000.000D



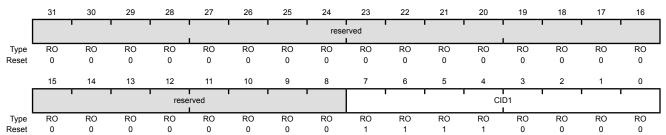
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	Watchdog PrimeCell ID Register[7:0]

Register 18: Watchdog PrimeCell Identification 1 (WDTPCellID1), offset 0xFF4

The WDTPCellIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog PrimeCell Identification 1 (WDTPCellID1)

Base 0x4000.0000 Offset 0xFF4 Type RO, reset 0x0000.00F0



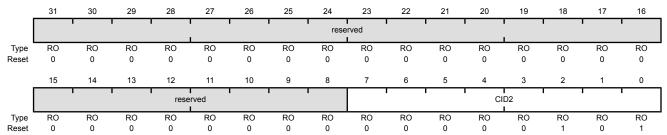
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	Watchdog PrimeCell ID Register[15:8]

Register 19: Watchdog PrimeCell Identification 2 (WDTPCellID2), offset 0xFF8

The WDTPCellIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog PrimeCell Identification 2 (WDTPCellID2)

Base 0x4000.0000 Offset 0xFF8 Type RO, reset 0x0000.0005



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	Watchdog PrimeCell ID Register[23:16]

Register 20: Watchdog PrimeCell Identification 3 (WDTPCellID3), offset 0xFFC

The WDTPCellIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog PrimeCell Identification 3 (WDTPCellID3)

Base 0x4000.0000 Offset 0xFFC Type RO, reset 0x0000.00B1



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID3	RO	0xB1	Watchdog PrimeCell ID Register[31:24]

10 Analog-to-Digital Converter (ADC)

An analog-to-digital converter (ADC) is a peripheral that converts a continuous analog voltage to a discrete digital number.

The Stellaris[®] ADC module features 10-bit conversion resolution and supports three input channels, plus an internal temperature sensor. The ADC module contains four programmable sequencer which allows for the sampling of multiple analog input sources without controller intervention. Each sample sequence provides flexible programming with fully configurable input source, trigger events, interrupt generation, and sequence priority.

The Stellaris ADC module provides the following features:

- Three analog input channels
- Single-ended and differential-input configurations
- On-chip internal temperature sensor
- Sample rate of 250 thousand samples/second
- Flexible, configurable analog-to-digital conversion
- Four programmable sample conversion sequences from one to eight entries long, with corresponding conversion result FIFOs
- Flexible trigger control
 - Controller (software)
 - Timers
 - Analog Comparators
 - PWM
 - GPIO
- Hardware averaging of up to 64 samples for improved accuracy
- Converter uses an internal 3-V reference
- Power and ground for the analog circuitry is separate from the digital power and ground

10.1 Block Diagram

Figure 10-1 on page 366 provides details on the internal configuration of the ADC controls and data registers.

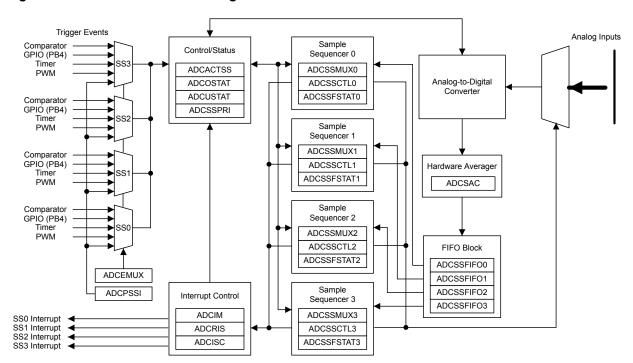


Figure 10-1. ADC Module Block Diagram

10.2 Signal Description

Table 10-1 on page 366 and Table 10-2 on page 366 list the external signals of the ADC module and describe the function of each. The signals are analog functions for some GPIO signals. The column in the table below titled "Pin Assignment" lists the GPIO pin placement for the ADC signals. The AINX analog signals are not 5-V tolerant and go through an isolation circuit before reaching their circuitry. These signals are configured by clearing the corresponding DEN bit in the **GPIO Digital Enable (GPIODEN)** register. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 10-1. ADC Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ADC0	1	I	Analog	Analog-to-digital converter input 0.
ADC1	2	I	Analog	Analog-to-digital converter input 1.
ADC2	5	I	Analog	Analog-to-digital converter input 2.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 10-2. ADC Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ADC0	B1	I	Analog	Analog-to-digital converter input 0.
ADC1	A1	I	Analog	Analog-to-digital converter input 1.
ADC2	В3	I	Analog	Analog-to-digital converter input 2.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

10.3 Functional Description

The Stellaris ADC collects sample data by using a programmable sequence-based approach instead of the traditional single or double-sampling approaches found on many ADC modules. Each *sample sequence* is a fully programmed series of consecutive (back-to-back) samples, allowing the ADC to collect data from multiple input sources without having to be re-configured or serviced by the controller. The programming of each sample in the sample sequence includes parameters such as the input source and mode (differential versus single-ended input), interrupt generation on sample completion, and the indicator for the last sample in the sequence.

10.3.1 Sample Sequencers

The sampling control and data capture is handled by the sample sequencers. All of the sequencers are identical in implementation except for the number of samples that can be captured and the depth of the FIFO. Table 10-3 on page 367 shows the maximum number of samples that each sequencer can capture and its corresponding FIFO depth. In this implementation, each FIFO entry is a 32-bit word, with the lower 10 bits containing the conversion result.

Table 10-3. Samples and FIFO Depth of Sequencers

Sequencer Number of Samples

Sequencer	Number of Samples	Depth of FIFO
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

For a given sample sequence, each sample is defined by two 4-bit nibbles in the ADC Sample Sequence Input Multiplexer Select (ADCSSMUXn) and ADC Sample Sequence Control (ADCSSCTLn) registers, where "n" corresponds to the sequence number. The ADCSSMUXn nibbles select the input pin, while the ADCSSCTLn nibbles contain the sample control bits corresponding to parameters such as temperature sensor selection, interrupt enable, end of sequence, and differential input mode. Sample sequencers are enabled by setting the respective ASENn bit in the ADC Active Sample Sequencer (ADCACTSS) register, and should be configured before being enabled.

When configuring a sample sequence, multiple uses of the same input pin within the same sequence is allowed. In the **ADCSSCTLn** register, the <code>IEn</code> bits can be set for any combination of samples, allowing interrupts to be generated after every sample in the sequence if necessary. Also, the <code>END</code> bit can be set at any point within a sample sequence. For example, if Sequencer 0 is used, the <code>END</code> bit can be set in the nibble associated with the fifth sample, allowing Sequencer 0 to complete execution of the sample sequence after the fifth sample.

After a sample sequence completes execution, the result data can be retrieved from the **ADC Sample Sequence Result FIFO (ADCSSFIFOn)** registers. The FIFOs are simple circular buffers that read a single address to "pop" result data. For software debug purposes, the positions of the FIFO head and tail pointers are visible in the **ADC Sample Sequence FIFO Status (ADCSSFSTATN)** registers along with FULL and EMPTY status flags. Overflow and underflow conditions are monitored using the **ADCOSTAT** and **ADCUSTAT** registers.

10.3.2 Module Control

Outside of the sample sequencers, the remainder of the control logic is responsible for tasks such as:

- Interrupt generation
- Sequence prioritization
- Trigger configuration

Most of the ADC control logic runs at the ADC clock rate of 14-18 MHz. The internal ADC divider is configured automatically by hardware when the system XTAL is selected. The automatic clock divider configuration targets 16.667 MHz operation for all Stellaris devices.

10.3.2.1 Interrupts

The register configurations of the sample sequencers dictate which events generate raw interrupts, but do not have control over whether the interrupt is actually sent to the interrupt controller. The ADC module's interrupt signals are controlled by the state of the MASK bits in the ADC Interrupt Mask (ADCIM) register. Interrupt status can be viewed at two locations: the ADC Raw Interrupt Status (ADCRIS) register, which shows the raw status of the various interrupt signals, and the ADC Interrupt Status and Clear (ADCISC) register, which shows active interrupts that are enabled by the ADCIM register. Sequencer interrupts are cleared by writing a 1 to the corresponding IN bit in ADCISC.

10.3.2.2 Prioritization

When sampling events (triggers) happen concurrently, they are prioritized for processing by the values in the ADC Sample Sequencer Priority (ADCSSPRI) register. Valid priority values are in the range of 0-3, with 0 being the highest priority and 3 being the lowest. Multiple active sample sequencer units with the same priority do not provide consistent results, so software must ensure that all active sample sequencer units have a unique priority value.

10.3.2.3 Sampling Events

Sample triggering for each sample sequencer is defined in the **ADC Event Multiplexer Select (ADCEMUX)** register. The external peripheral triggering sources vary by Stellaris family member, but all devices share the "Controller" and "Always" triggers. Software can initiate sampling by setting the SSx bits in the **ADC Processor Sample Sequence Initiate (ADCPSSI)** register.

Care must be taken when using the "Always" trigger. If a sequence's priority is too high, it is possible to starve other lower priority sequences.

10.3.3 Hardware Sample Averaging Circuit

Higher precision results can be generated using the hardware averaging circuit, however, the improved results are at the cost of throughput. Up to 64 samples can be accumulated and averaged to form a single data entry in the sequencer FIFO. Throughput is decreased proportionally to the number of samples in the averaging calculation. For example, if the averaging circuit is configured to average 16 samples, the throughput is decreased by a factor of 16.

By default the averaging circuit is off and all data from the converter passes through to the sequencer FIFO. The averaging hardware is controlled by the **ADC Sample Averaging Control (ADCSAC)** register (see page 387). There is a single averaging circuit and all input channels receive the same amount of averaging whether they are single-ended or differential.

10.3.4 Analog-to-Digital Converter

The converter itself generates a 10-bit output value for selected analog input. Special analog pads are used to minimize the distortion on the input. An internal 3 V reference is used by the converter

resulting in sample values ranging from 0x000 at 0 V input to 0x3FF at 3 V input when in single-ended input mode.

10.3.5 Differential Sampling

In addition to traditional single-ended sampling, the ADC module supports differential sampling of two analog input channels. To enable differential sampling, software must set the Dn bit in the **ADCSSCTL0n** register in a step's configuration nibble.

When a sequence step is configured for differential sampling, its corresponding value in the **ADCSSMUXn** register must be set to one of the four differential pairs, numbered 0-3. Differential pair 0 samples analog inputs 0 and 1; differential pair 1 samples analog inputs 2 and 3; and so on (see Table 10-4 on page 369). The ADC does not support other differential pairings such as analog input 0 with analog input 3. The number of differential pairs supported is dependent on the number of analog inputs (see Table 10-4 on page 369).

Table 10-4. Differential Sampling Pairs

Differential Pair	Analog Inputs
0	0 and 1
1	2 and 3

The voltage sampled in differential mode is the difference between the odd and even channels:

 ΔV (differential voltage) = V_{IN} (even channels) – V_{IN} (odd channels), therefore:

- If $\Delta V = 0$, then the conversion result = 0x1FF
- If $\Delta V > 0$, then the conversion result > 0x1FF (range is 0x1FF–0x3FF)
- If $\Delta V < 0$, then the conversion result < 0x1FF (range is 0–0x1FF)

The differential pairs assign polarities to the analog inputs: the even-numbered input is always positive, and the odd-numbered input is always negative. In order for a valid conversion result to appear, the negative input must be in the range of \pm 1.5 V of the positive input. If an analog input is greater than 3 V or less than 0 V (the valid range for analog inputs), the input voltage is clipped, meaning it appears as either 3 V or 0 V, respectively, to the ADC.

Figure 10-2 on page 370 shows an example of the negative input centered at 1.5 V. In this configuration, the differential range spans from -1.5 V to 1.5 V. Figure 10-3 on page 370 shows an example where the negative input is centered at -0.75 V, meaning inputs on the positive input saturate past a differential voltage of -0.75 V since the input voltage is less than 0 V. Figure 10-4 on page 371 shows an example of the negative input centered at 2.25 V, where inputs on the positive channel saturate past a differential voltage of 0.75 V since the input voltage would be greater than 3 V.



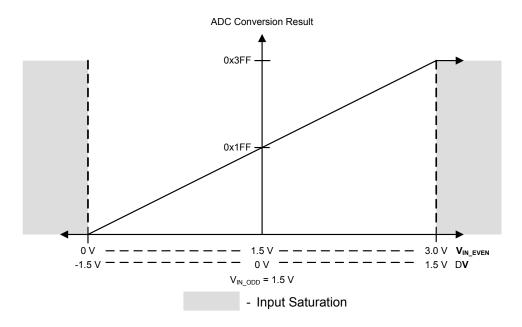
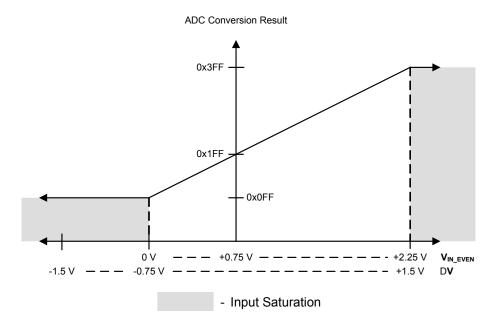


Figure 10-3. Differential Sampling Range, V_{IN_ODD} = 0.75 V



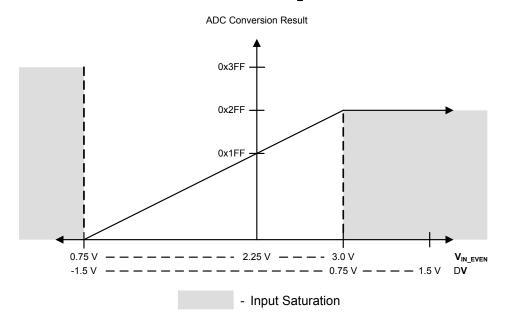


Figure 10-4. Differential Sampling Range, $V_{IN\ ODD}$ = 2.25 V

10.3.6 Test Modes

There is a user-available test mode that allows for loopback operation within the digital portion of the ADC module. This can be useful for debugging software without having to provide actual analog stimulus. This mode is available through the **ADC Test Mode Loopback (ADCTMLB)** register (see page 400).

10.3.7 Internal Temperature Sensor

The temperature sensor's primary purpose is to notify the system that the internal temperature is too high or low for reliable operation.

The temperature sensor does not have a separate enable, since it also contains the bandgap reference and must always be enabled. The reference is supplied to other analog modules; not just the ADC.

The internal temperature sensor provides an analog temperature reading as well as a reference voltage. The voltage at the output terminal SENSO is given by the following equation:

$$SENSO = 2.7 - ((T + 55) / 75)$$

This relation is shown in Figure 10-5 on page 372.

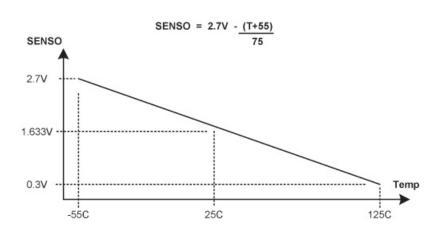


Figure 10-5. Internal Temperature Sensor Characteristic

10.4 Initialization and Configuration

In order for the ADC module to be used, the PLL must be enabled and using a supported crystal frequency (see the **RCC** register). Using unsupported frequencies can cause faulty operation in the ADC module.

10.4.1 Module Initialization

Initialization of the ADC module is a simple process with very few steps. The main steps include enabling the clock to the ADC and reconfiguring the sample sequencer priorities (if needed).

The initialization sequence for the ADC is as follows:

- 1. Enable the ADC clock by writing a value of 0x0001.0000 to the **RCGC0** register (see page 209).
- 2. If required by the application, reconfigure the sample sequencer priorities in the **ADCSSPRI** register. The default configuration has Sample Sequencer 0 with the highest priority, and Sample Sequencer 3 as the lowest priority.

10.4.2 Sample Sequencer Configuration

Configuration of the sample sequencers is slightly more complex than the module initialization since each sample sequence is completely programmable.

The configuration for each sample sequencer should be as follows:

- 1. Ensure that the sample sequencer is disabled by writing a 0 to the corresponding ASENn bit in the **ADCACTSS** register. Programming of the sample sequencers is allowed without having them enabled. Disabling the sequencer during programming prevents erroneous execution if a trigger event were to occur during the configuration process.
- 2. Configure the trigger event for the sample sequencer in the ADCEMUX register.
- **3.** For each sample in the sample sequence, configure the corresponding input source in the **ADCSSMUXn** register.

- **4.** For each sample in the sample sequence, configure the sample control bits in the corresponding nibble in the **ADCSSCTLn** register. When programming the last nibble, ensure that the END bit is set. Failure to set the END bit causes unpredictable behavior.
- 5. If interrupts are to be used, write a 1 to the corresponding MASK bit in the ADCIM register.
- **6.** Enable the sample sequencer logic by writing a 1 to the corresponding ASENn bit in the **ADCACTSS** register.

10.5 Register Map

Table 10-5 on page 373 lists the ADC registers. The offset listed is a hexadecimal increment to the register's address, relative to the ADC base address of 0x4003.8000.

Note that the ADC module clock must be enabled before the registers can be programmed (see page 209). There must be a delay of 3 system clocks after the ADC module clock is enabled before any ADC module registers are accessed.

Table 10-5. ADC Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	ADCACTSS	R/W	0x0000.0000	ADC Active Sample Sequencer	375
0x004	ADCRIS	RO	0x0000.0000	ADC Raw Interrupt Status	376
800x0	ADCIM	R/W	0x0000.0000	ADC Interrupt Mask	377
0x00C	ADCISC	R/W1C	0x0000.0000	ADC Interrupt Status and Clear	378
0x010	ADCOSTAT	R/W1C	0x0000.0000	ADC Overflow Status	379
0x014	ADCEMUX	R/W	0x0000.0000	ADC Event Multiplexer Select	380
0x018	ADCUSTAT	R/W1C	0x0000.0000	ADC Underflow Status	383
0x020	ADCSSPRI	R/W	0x0000.3210	ADC Sample Sequencer Priority	384
0x028	ADCPSSI	WO	-	ADC Processor Sample Sequence Initiate	386
0x030	ADCSAC	R/W	0x0000.0000	ADC Sample Averaging Control	387
0x040	ADCSSMUX0	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 0	388
0x044	ADCSSCTL0	R/W	0x0000.0000	ADC Sample Sequence Control 0	390
0x048	ADCSSFIFO0	RO	-	ADC Sample Sequence Result FIFO 0	393
0x04C	ADCSSFSTAT0	RO	0x0000.0100	ADC Sample Sequence FIFO 0 Status	394
0x060	ADCSSMUX1	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 1	395
0x064	ADCSSCTL1	R/W	0x0000.0000	ADC Sample Sequence Control 1	396
0x068	ADCSSFIFO1	RO	-	ADC Sample Sequence Result FIFO 1	393
0x06C	ADCSSFSTAT1	RO	0x0000.0100	ADC Sample Sequence FIFO 1 Status	394
0x080	ADCSSMUX2	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 2	395
0x084	ADCSSCTL2	R/W	0x0000.0000	ADC Sample Sequence Control 2	396
0x088	ADCSSFIFO2	RO	-	ADC Sample Sequence Result FIFO 2	393

Table 10-5. ADC Register Map (continued)

Offset	Name	Type	Reset	Description	See page
0x08C	ADCSSFSTAT2	RO	0x0000.0100	ADC Sample Sequence FIFO 2 Status	394
0x0A0	ADCSSMUX3	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 3	398
0x0A4	ADCSSCTL3	R/W	0x0000.0002	ADC Sample Sequence Control 3	399
0x0A8	ADCSSFIFO3	RO	-	ADC Sample Sequence Result FIFO 3	393
0x0AC	ADCSSFSTAT3	RO	0x0000.0100	ADC Sample Sequence FIFO 3 Status	394
0x100	ADCTMLB	R/W	0x0000.0000	ADC Test Mode Loopback	400

10.6 Register Descriptions

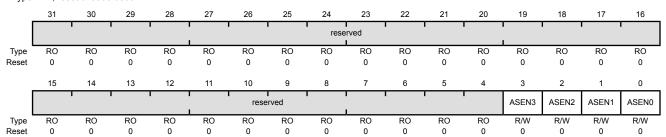
The remainder of this section lists and describes the ADC registers, in numerical order by address offset.

Register 1: ADC Active Sample Sequencer (ADCACTSS), offset 0x000

This register controls the activation of the sample sequencers. Each sample sequencer can be enabled or disabled independently.

ADC Active Sample Sequencer (ADCACTSS)

Base 0x4003.8000 Offset 0x000 Type R/W, reset 0x0000.0000



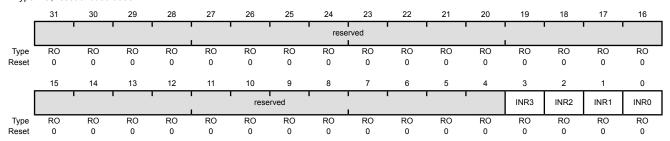
Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	ASEN3	R/W	0	ADC SS3 Enable Specifies whether Sample Sequencer 3 is enabled. If set, the sample sequence logic for Sequencer 3 is active. Otherwise, the sequencer is inactive.
2	ASEN2	R/W	0	ADC SS2 Enable Specifies whether Sample Sequencer 2 is enabled. If set, the sample sequence logic for Sequencer 2 is active. Otherwise, the sequencer is inactive.
1	ASEN1	R/W	0	ADC SS1 Enable Specifies whether Sample Sequencer 1 is enabled. If set, the sample sequence logic for Sequencer 1 is active. Otherwise, the sequencer is inactive.
0	ASEN0	R/W	0	ADC SS0 Enable Specifies whether Sample Sequencer 0 is enabled. If set, the sample sequence logic for Sequencer 0 is active. Otherwise, the sequencer is inactive.

Register 2: ADC Raw Interrupt Status (ADCRIS), offset 0x004

This register shows the status of the raw interrupt signal of each sample sequencer. These bits may be polled by software to look for interrupt conditions without having to generate controller interrupts.

ADC Raw Interrupt Status (ADCRIS)

Base 0x4003.8000 Offset 0x004 Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	INR3	RO	0	SS3 Raw Interrupt Status
				This bit is set by hardware when a sample with its respective ADCSSCTL3 IE bit has completed conversion. This bit is cleared by setting the IN3 bit in the ADCISC register.
2	INR2	RO	0	SS2 Raw Interrupt Status
				This bit is set by hardware when a sample with its respective ADCSSCTL2 IE bit has completed conversion. This bit is cleared by setting the IN2 bit in the ADCISC register.
1	INR1	RO	0	SS1 Raw Interrupt Status
				This bit is set by hardware when a sample with its respective ADCSSCTL1 IE bit has completed conversion. This bit is cleared by setting the IN1 bit in the ADCISC register.
0	INR0	RO	0	SS0 Raw Interrupt Status
				This bit is set by hardware when a sample with its respective ADCSSCTL0 IE bit has completed conversion. This bit is cleared by

setting the IN30 bit in the ADCISC register.

Register 3: ADC Interrupt Mask (ADCIM), offset 0x008

Reset

This register controls whether the sample sequencer raw interrupt signals are promoted to controller interrupts. Each raw interrupt signal can be masked independently.

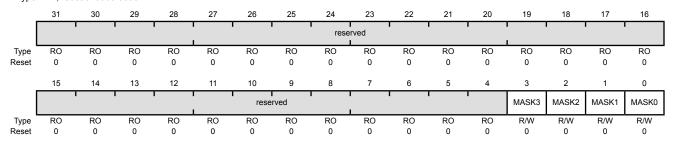
ADC Interrupt Mask (ADCIM)

Name

Type

Bit/Field

Base 0x4003.8000 Offset 0x008 Type R/W, reset 0x0000.0000



Description

31:4	reserved	RO	0x000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	MASK3	R/W	0	SS3 Interrupt Mask
				When set, this bit allows the raw interrupt signal from Sample Sequencer 3 (ADCRIS register INR3 bit) to be promoted to a controller interrupt.
				When clear, the status of Sample Sequencer 3 does not affect the SS3 interrupt status.
2	MASK2	R/W	0	SS2 Interrupt Mask
				When set, this bit allows the raw interrupt signal from Sample Sequencer 2 (ADCRIS register INR2 bit) to be promoted to a controller interrupt.
				When clear, the status of Sample Sequencer 2 does not affect the SS2 interrupt status.
1	MASK1	R/W	0	SS1 Interrupt Mask
				When set, this bit allows the raw interrupt signal from Sample Sequencer 1 (ADCRIS register INR1 bit) to be promoted to a controller interrupt.
				When clear, the status of Sample Sequencer 1 does not affect the SS1 interrupt status.
0	MASK0	R/W	0	SS0 Interrupt Mask
				When set, this bit allows the raw interrupt signal from Sample Sequencer 0 (ADCRIS register INR0 bit) to be promoted to a controller interrupt.
				When clear, the status of Sample Sequencer 0 does not affect the SS0

interrupt status.

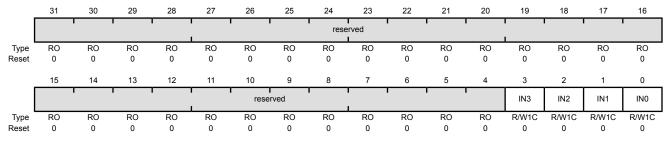
Register 4: ADC Interrupt Status and Clear (ADCISC), offset 0x00C

This register provides the mechanism for clearing sample sequence interrupt conditions and shows the status of controller interrupts generated by the sample sequencers. When read, each bit field is the logical AND of the respective INR and MASK bits. Sample sequence nterrupts are cleared by setting the corresponding bit position. If software is polling the **ADCRIS** instead of generating interrupts, the sample sequence INR bits are still cleared via the **ADCISC** register, even if the IN bit is not set.

ADC Interrupt Status and Clear (ADCISC)

Base 0x4003.8000 Offset 0x00C

Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	IN3	R/W1C	0	SS3 Interrupt Status and Clear
				This bit is set when both the INR3 bit in the ADCRIS register and the MASK3 bit in the ADCIM register are set, providing a level-based interrupt to the controller.
				This bit is cleared by writing a 1. Clearing this bit also clears the ${\tt INR3}$ bit.
2	IN2	R/W1C	0	SS2 Interrupt Status and Clear
				This bit is set when both the INR2 bit in the ADCRIS register and the MASK2 bit in the ADCIM register are set, providing a level-based interrupt to the controller.
				This bit is cleared by writing a 1. Clearing this bit also clears the ${\tt INR2}$ bit.
1	IN1	R/W1C	0	SS1 Interrupt Status and Clear
				This bit is set when both the INR1 bit in the ADCRIS register and the MASK1 bit in the ADCIM register are set, providing a level-based interrupt to the controller.
				This bit is cleared by writing a 1. Clearing this bit also clears the ${\tt INR1}$ bit.
0	INO	R/W1C	0	SS0 Interrupt Status and Clear
				This bit is set when both the INRO bit in the ADCRIS register and the MASKO bit in the ADCIM register are set, providing a level-based interrupt

to the controller.

This bit is cleared by writing a 1. Clearing this bit also clears the INRO

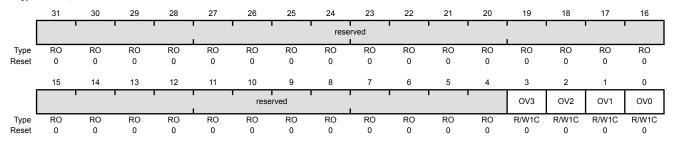
Register 5: ADC Overflow Status (ADCOSTAT), offset 0x010

This register indicates overflow conditions in the sample sequencer FIFOs. Once the overflow condition has been handled by software, the condition can be cleared by writing a 1 to the corresponding bit position.

ADC Overflow Status (ADCOSTAT)

Base 0x4003.8000

Offset 0x010 Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	OV3	R/W1C	0	SS3 FIFO Overflow When set, this bit specifies that the FIFO for Sample Sequencer 3 has hit an overflow condition where the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped. This bit is cleared by writing a 1.
2	OV2	R/W1C	0	SS2 FIFO Overflow When set, this bit specifies that the FIFO for Sample Sequencer 2 has hit an overflow condition where the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped. This bit is cleared by writing a 1.
1	OV1	R/W1C	0	SS1 FIFO Overflow When set, this bit specifies that the FIFO for Sample Sequencer 1 has hit an overflow condition where the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped. This bit is cleared by writing a 1.
0	OV0	R/W1C	0	SS0 FIFO Overflow When set, this bit specifies that the FIFO for Sample Sequencer 0 has hit an overflow condition where the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped.

July 15, 2014 379

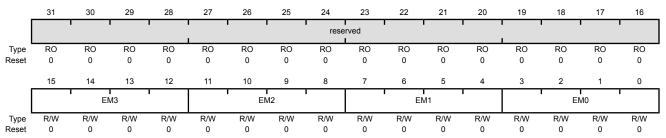
This bit is cleared by writing a 1.

Register 6: ADC Event Multiplexer Select (ADCEMUX), offset 0x014

The ADCEMUX selects the event (trigger) that initiates sampling for each sample sequencer. Each sample sequencer can be configured with a unique trigger source.

ADC Event Multiplexer Select (ADCEMUX)

Base 0x4003.8000 Offset 0x014 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:12	EM3	R/W	0x0	SS3 Trigger Select

Value

This field selects the trigger source for Sample Sequencer 3.

The valid configurations for this field are:

Event

0x0	Controller (default)
0x1	Analog Comparator 0
0x2	Analog Comparator 1
0x3	Reserved
0x4	External (GPIO PB4)
0x5	Timer
	In addition, the trigger must be enabled with the ${\tt TnOTE}$ bit in the ${\tt GPTMCTL}$ register (see page 322).
0x6	PWM0
	The PWM module 0 trigger can be configured with the PWM0 Interrupt and Trigger Enable (PWM0INTEN) register, see page 598.
0x7	PWM1
	The PWM module 1 trigger can be configured with the PWM1INTEN register, see page 598.

0x9-0xE reserved 0xF Always (continuously sample)

reserved

0x8

Bit/Field	Name	Туре	Reset	Description	on
11:8	EM2	R/W	0x0	This field	ger Select selects the trigger source for Sample Sequencer 2. configurations for this field are:
					·
				Value	Event
				0x0	Controller (default)
				0x1	Analog Comparator 0
				0x2	Analog Comparator 1
				0x3	Reserved
				0x4	External (GPIO PB4)
				0x5	Timer
					In addition, the trigger must be enabled with the Thote bit in the GPTMCTL register (see page 322).
				0x6	PWM0
					The PWM module 0 trigger can be configured with the PWM0 Interrupt and Trigger Enable (PWM0INTEN) register, see page 598.
				0x7	PWM1
					The PWM module 1 trigger can be configured with the PWM1INTEN register, see page 598.
				0x8	reserved
				0x9-0xE	reserved
				0xF	Always (continuously sample)
7:4	EM1	R/W	0x0	SS1 Trigg	ger Select
				This field	selects the trigger source for Sample Sequencer 1.
				The valid	configurations for this field are:
				Value	Event
				0x0	Controller (default)
				0x1	Analog Comparator 0
				0x2	Analog Comparator 1
				0x3	Reserved
				0x4	External (GPIO PB4)
				0x5	Timer
					In addition, the trigger must be enabled with the ${ t TnOTE}$ bit in the GPTMCTL register (see page 322).
				0x6	PWM0
					The PWM module 0 trigger can be configured with the PWM0 Interrupt and Trigger Enable (PWM0INTEN) register, see page 598.
				0x7	PWM1
					The PWM module 1 trigger can be configured with the PWM1INTEN register, see page 598.
				0x8	reserved
				0x9-0xE	reserved
				0xF	Always (continuously sample)

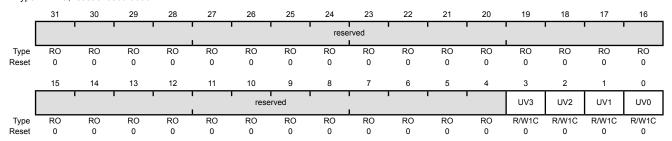
Bit/Field	Name	Туре	Reset	Descript	ion
3:0	EM0	R/W	0x0	This field	ger Select d selects the trigger source for Sample Sequencer 0. d configurations for this field are:
				Value	Event
				0x0	Controller (default)
				0x1	Analog Comparator 0
				0x2	Analog Comparator 1
				0x3	Reserved
				0x4	External (GPIO PB4)
				0x5	Timer
					In addition, the trigger must be enabled with the ${ t TnOTE}$ bit in the GPTMCTL register (see page 322).
				0x6	PWM0
					The PWM module 0 trigger can be configured with the PWM0 Interrupt and Trigger Enable (PWM0INTEN) register, see page 598.
				0x7	PWM1
					The PWM module 1 trigger can be configured with the PWM1INTEN register, see page 598.
				0x8	reserved
				0x9-0xE	reserved
				0xF	Always (continuously sample)

Register 7: ADC Underflow Status (ADCUSTAT), offset 0x018

This register indicates underflow conditions in the sample sequencer FIFOs. The corresponding underflow condition is cleared by writing a 1 to the relevant bit position.

ADC Underflow Status (ADCUSTAT)

Base 0x4003.8000 Offset 0x018 Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	UV3	R/W1C	0	SS3 FIFO Underflow
				When set, this bit specifies that the FIFO for Sample Sequencer 3 has hit an underflow condition where the FIFO is empty and a read was requested. The problematic read does not move the FIFO pointers, and 0s are returned.
				This bit is cleared by writing a 1.
2	UV2	R/W1C	0	SS2 FIFO Underflow
				When set, this bit specifies that the FIFO for Sample Sequencer 2 has hit an underflow condition where the FIFO is empty and a read was requested. The problematic read does not move the FIFO pointers, and 0s are returned.
				This bit is cleared by writing a 1.
1	UV1	R/W1C	0	SS1 FIFO Underflow
				When set, this bit specifies that the FIFO for Sample Sequencer 1 has hit an underflow condition where the FIFO is empty and a read was requested. The problematic read does not move the FIFO pointers, and 0s are returned.
				This bit is cleared by writing a 1.
0	UV0	R/W1C	0	SS0 FIFO Underflow
				When set, this bit specifies that the FIFO for Sample Sequencer 0 has hit an underflow condition where the FIFO is empty and a read was requested. The problematic read does not move the FIFO pointers, and

This bit is cleared by writing a 1.

0s are returned.

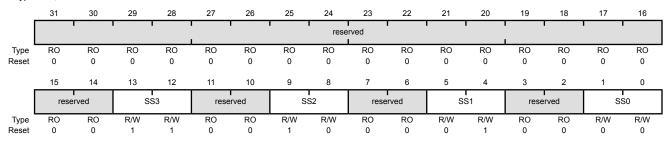
Register 8: ADC Sample Sequencer Priority (ADCSSPRI), offset 0x020

This register sets the priority for each of the sample sequencers. Out of reset, Sequencer 0 has the highest priority, and Sequencer 3 has the lowest priority. When reconfiguring sequence priorities, each sequence must have a unique priority for the ADC to operate properly.

ADC Sample Sequencer Priority (ADCSSPRI)

Base 0x4003.8000

Offset 0x020 Type R/W, reset 0x0000.3210



Bit/Field	Name	Туре	Reset	Description
31:14	reserved	RO	0x0000.0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13:12	SS3	R/W	0x3	SS3 Priority This field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 3. A priority encoding of 0 is highest and 3 is lowest. The priorities assigned to the sequencers must be uniquely mapped. The ADC may not operate properly if two or more fields are equal.
11:10	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9:8	SS2	R/W	0x2	SS2 Priority This field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 2. A priority encoding of 0 is highest and 3 is lowest. The priorities assigned to the sequencers must be uniquely mapped. The ADC may not operate properly if two or more fields are equal.
7:6	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:4	SS1	R/W	0x1	SS1 Priority This field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 1. A priority encoding of 0 is highest and 3 is lowest. The priorities assigned to the sequencers must be uniquely mapped. The ADC may not operate properly if two or more fields are equal.
3:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

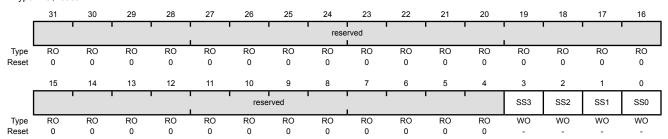
Bit/Field	Name	Type	Reset	Description
1:0	SS0	R/W	0x0	SS0 Priority This field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 0. A priority encoding of 0 is highest and 3 is lowest. The priorities assigned to the sequencers must be uniquely mapped. The ADC may not operate properly if two or more fields are equal.

Register 9: ADC Processor Sample Sequence Initiate (ADCPSSI), offset 0x028

This register provides a mechanism for application software to initiate sampling in the sample sequencers. Sample sequences can be initiated individually or in any combination. When multiple sequences are triggered simultaneously, the priority encodings in **ADCSSPRI** dictate execution order.

ADC Processor Sample Sequence Initiate (ADCPSSI)

Base 0x4003.8000 Offset 0x028 Type WO, reset -



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	SS3	WO	-	SS3 Initiate
				When set, this bit triggers sampling on Sample Sequencer 3 if the sequencer is enabled in the ADCACTSS register.
				Only a write by software is valid; a read of this register returns no meaningful data.
2	SS2	WO	_	SS2 Initiate
				When set, this bit triggers sampling on Sample Sequencer 2 if the sequencer is enabled in the ADCACTSS register.
				Only a write by software is valid; a read of this register returns no meaningful data.
1	SS1	WO	-	SS1 Initiate
				When set, this bit triggers sampling on Sample Sequencer 1 if the sequencer is enabled in the ADCACTSS register.
				Only a write by software is valid; a read of this register returns no meaningful data.
0	SS0	WO	-	SS0 Initiate
				When set, this bit triggers sampling on Sample Sequencer 0 if the sequencer is enabled in the ADCACTSS register.
				Only a write by software is valid; a read of this register returns no

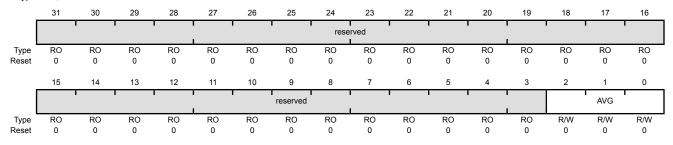
meaningful data.

Register 10: ADC Sample Averaging Control (ADCSAC), offset 0x030

This register controls the amount of hardware averaging applied to conversion results. The final conversion result stored in the FIFO is averaged from 2^{AVG} consecutive ADC samples at the specified ADC speed. If AVG is 0, the sample is passed directly through without any averaging. If AVG=6, then 64 consecutive ADC samples are averaged to generate one result in the sequencer FIFO. An AVG = 7 provides unpredictable results.

ADC Sample Averaging Control (ADCSAC)

Base 0x4003.8000 Offset 0x030 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:3	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2:0	AVG	R/W	0x0	Hardware Averaging Control

Specifies the amount of hardware averaging that will be applied to ADC samples. The AVG field can be any value between 0 and 6. Entering a value of 7 creates unpredictable results.

Value Description 0x0 No hardware oversampling 2x hardware oversampling 0x1 0x2 4x hardware oversampling 0x3 8x hardware oversampling 16x hardware oversampling 0x4 32x hardware oversampling 0x5 64x hardware oversampling 0x6 Reserved 0x7

Register 11: ADC Sample Sequence Input Multiplexer Select 0 (ADCSSMUX0), offset 0x040

This register defines the analog input configuration for each sample in a sequence executed with Sample Sequencer 0. This register is 32 bits wide and contains information for eight possible samples.

ADC Sample Sequence Input Multiplexer Select 0 (ADCSSMUX0)

Base 0x4003.8000 Offset 0x040 Type R/W, reset 0x0000.0000

,,	,																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	rese	rved	МС	JX7	rese	erved	М	JX6	rese	erved	м	JX5	rese	rved	МС	IX4		
Type Reset	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0		
Neset																		
Γ	15 rese	14 nved	13 MI	12 JX3	11	10 rved	9 MI	8 JX2	7	6 erved	5 MI	4 JX1	3	2 rved	1 ML	0		
[
Type Reset	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0		
В	sit/Field		Nan	ne	Ту	pe	Reset	Des	cription									
	31:30		reser	ved	R	0	0	com	patibility	with fut	ure prod	he value ucts, the dify-write	value of	a reserv				
	29:28		MUX	< 7	R	W	0x0	8th	th Sample Input Select									
								with sam	the san pled for correspo	nple sequ the analo	uencer. I og-to-digi	the eight t specifie tal conve ample, a	s which rsion. Th	of the ar	nalog inp set here i	uts is ndicates		
	27:26		reser	ved	R	0	0	com	patibility	with fut	ure prod	he value ucts, the dify-write	value of	a reserv				
	25:24		MUX	K 6	R	W	0x0	7th	Sample	Input Se	lect							
								exe	cuted wi	th the sa	ımple se	during the seventh sample of a sequence le sequencer. It specifies which of the analoge analog-to-digital conversion.						
	23:22		reser	ved	R	Ю.	0	com	patibility	with fut	ure prod	he value ucts, the dify-write	value of	a reserv				
	21:20		MUX	< 5	R	W	0x0	6th	Sample	Input Se	lect							
			MUX5		R/W			The with	The MUX5 field is used during the sixth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.									
19:18			reser	ved	R	0	0	com	patibility	with fut	ure prod	he value ucts, the dify-write	value of	a reserv				

Bit/Field	Name	Туре	Reset	Description
17:16	MUX4	R/W	0x0	5th Sample Input Select The MUX4 field is used during the fifth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.
15:14	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13:12	MUX3	R/W	0x0	4th Sample Input Select The MUX3 field is used during the fourth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.
11:10	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9:8	MUX2	R/W	0x0	3rd Sample Input Select The MUX72 field is used during the third sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.
7:6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:4	MUX1	R/W	0x0	2nd Sample Input Select The MUX1 field is used during the second sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1:0	MUX0	R/W	0x0	1st Sample Input Select The MUX0 field is used during the first sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion.

Register 12: ADC Sample Sequence Control 0 (ADCSSCTL0), offset 0x044

This register contains the configuration information for each sample for a sequence executed with a sample sequencer. When configuring a sample sequence, the END bit must be set at some point, whether it be after the first sample, last sample, or any sample in between. This register is 32-bits wide and contains information for eight possible samples.

ADC Sample Sequence Control 0 (ADCSSCTL0)

Base 0x4003.8000

Offset 0x044 Type R/W, reset 0x0000.0000

-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TS7	IE7	END7	D7	TS6	IE6	END6	D6	TS5	IE5	END5	D5	TS4	IE4	END4	D4
Туре	R/W	R/W	R/W	R/W												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
Туре	R/W	R/W	R/W	R/W												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		_		
Bit/Field	Name	Type	Reset	Description
31	TS7	R/W	0	8th Sample Temp Sensor Select
				This bit is used during the eighth sample of the sample sequence and and specifies the input source of the sample.
				When set, the temperature sensor is read.
				When clear, the input pin specified by the ADCSSMUX register is read.
30	IE7	R/W	0	8th Sample Interrupt Enable
				This bit is used during the eighth sample of the sample sequence and specifies whether the raw interrupt signal (INRO bit) is asserted at the end of the sample's conversion. If the MASKO bit in the ADCIM register is set, the interrupt is promoted to a controller-level interrupt.
				When this bit is set, the raw interrupt is asserted.
				When this bit is clear, the raw interrupt is not asserted.
				It is legal to have multiple samples within a sequence generate interrupts.
29	END7	R/W	0	8th Sample is End of Sequence
				The END7 bit indicates that this is the last sample of the sequence. It is possible to end the sequence on any sample position. Samples defined after the sample containing a set END are not requested for conversion even though the fields may be non-zero. It is required that software write the END bit somewhere within the sequence. (Sample Sequencer 3, which only has a single sample in the sequence, is hardwired to have the END0 bit set.)
				Setting this bit indicates that this sample is the last in the sequence.
28	D7	R/W	0	8th Sample Diff Input Select
				The D7 bit indicates that the analog input is to be differentially sampled. The corresponding ADCSSMUXx nibble must be set to the pair number "i", where the paired inputs are "2i and 2i+1". The temperature sensor does not have a differential option. When set, the analog inputs are differentially sampled.
27	TS6	R/W	0	7th Sample Temp Sensor Select Same definition as TS7 but used during the seventh sample.

Bit/Field	Name	Туре	Reset	Description
26	IE6	R/W	0	7th Sample Interrupt Enable Same definition as IE7 but used during the seventh sample.
25	END6	R/W	0	7th Sample is End of Sequence Same definition as END7 but used during the seventh sample.
24	D6	R/W	0	7th Sample Diff Input Select Same definition as D7 but used during the seventh sample.
23	TS5	R/W	0	6th Sample Temp Sensor Select Same definition as TS7 but used during the sixth sample.
22	IE5	R/W	0	6th Sample Interrupt Enable Same definition as IE7 but used during the sixth sample.
21	END5	R/W	0	6th Sample is End of Sequence Same definition as END7 but used during the sixth sample.
20	D5	R/W	0	6th Sample Diff Input Select Same definition as D7 but used during the sixth sample.
19	TS4	R/W	0	5th Sample Temp Sensor Select Same definition as TS7 but used during the fifth sample.
18	IE4	R/W	0	5th Sample Interrupt Enable Same definition as IE7 but used during the fifth sample.
17	END4	R/W	0	5th Sample is End of Sequence Same definition as END7 but used during the fifth sample.
16	D4	R/W	0	5th Sample Diff Input Select Same definition as D7 but used during the fifth sample.
15	TS3	R/W	0	4th Sample Temp Sensor Select Same definition as TS7 but used during the fourth sample.
14	IE3	R/W	0	4th Sample Interrupt Enable Same definition as IE7 but used during the fourth sample.
13	END3	R/W	0	4th Sample is End of Sequence Same definition as END7 but used during the fourth sample.
12	D3	R/W	0	4th Sample Diff Input Select Same definition as D7 but used during the fourth sample.
11	TS2	R/W	0	3rd Sample Temp Sensor Select Same definition as TS7 but used during the third sample.
10	IE2	R/W	0	3rd Sample Interrupt Enable Same definition as IE7 but used during the third sample.
9	END2	R/W	0	3rd Sample is End of Sequence Same definition as END7 but used during the third sample.

July 15, 2014 391

Bit/Field	Name	Туре	Reset	Description
8	D2	R/W	0	3rd Sample Diff Input Select Same definition as D7 but used during the third sample.
7	TS1	R/W	0	2nd Sample Temp Sensor Select Same definition as TS7 but used during the second sample.
6	IE1	R/W	0	2nd Sample Interrupt Enable Same definition as IE7 but used during the second sample.
5	END1	R/W	0	2nd Sample is End of Sequence Same definition as END7 but used during the second sample.
4	D1	R/W	0	2nd Sample Diff Input Select Same definition as D7 but used during the second sample.
3	TS0	R/W	0	1st Sample Temp Sensor Select Same definition as TS7 but used during the first sample.
2	IE0	R/W	0	1st Sample Interrupt Enable Same definition as IE7 but used during the first sample.
1	END0	R/W	0	1st Sample is End of Sequence Same definition as END7 but used during the first sample.
0	D0	R/W	0	1st Sample Diff Input Select Same definition as D7 but used during the first sample.

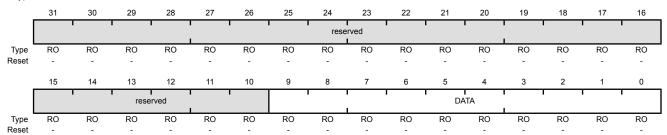
Register 13: ADC Sample Sequence Result FIFO 0 (ADCSSFIFO0), offset 0x048 Register 14: ADC Sample Sequence Result FIFO 1 (ADCSSFIFO1), offset 0x068 Register 15: ADC Sample Sequence Result FIFO 2 (ADCSSFIFO2), offset 0x088 Register 16: ADC Sample Sequence Result FIFO 3 (ADCSSFIFO3), offset 0x0A8

Important: This register is read-sensitive. See the register description for details.

This register contains the conversion results for samples collected with the sample sequencer (the ADCSSFIFO0 register is used for Sample Sequencer 0, ADCSSFIFO1 for Sequencer 1, ADCSSFIFO2 for Sequencer 2, and ADCSSFIFO3 for Sequencer 3). Reads of this register return conversion result data in the order sample 0, sample 1, and so on, until the FIFO is empty. If the FIFO is not properly handled by software, overflow and underflow conditions are registered in the ADCOSTAT and ADCUSTAT registers.

ADC Sample Sequence Result FIFO 0 (ADCSSFIFO0)

Base 0x4003.8000 Offset 0x048 Type RO, reset -



Bit/Field	Name	Type	Reset	Description
31:10	reserved	RO	-	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9:0	DATA	RO	-	Conversion Result Data

Register 17: ADC Sample Sequence FIFO 0 Status (ADCSSFSTAT0), offset 0x04C

Register 18: ADC Sample Sequence FIFO 1 Status (ADCSSFSTAT1), offset 0x06C

Register 19: ADC Sample Sequence FIFO 2 Status (ADCSSFSTAT2), offset 0x08C

Register 20: ADC Sample Sequence FIFO 3 Status (ADCSSFSTAT3), offset 0x0AC

This register provides a window into the sample sequencer, providing full/empty status information as well as the positions of the head and tail pointers. The reset value of 0x100 indicates an empty FIFO. The ADCSSFSTAT0 register provides status on FIFO0, ADCSSFSTAT1 on FIFO1, ADCSSFSTAT2 on FIFO2, and ADCSSFSTAT3 on FIFO3.

ADC Sample Sequence FIFO 0 Status (ADCSSFSTAT0)

Base 0x4003.8000 Offset 0x04C Type RO, reset 0x0000.0100

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		' '						rese	rved							
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved FUI			FULL		reserved		EMPTY		HP	TR			TP	TR	
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 0							

Bit/Field	Name	Туре	Reset	Description
31:13	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	FULL	RO	0	FIFO Full
				When set, this bit indicates that the FIFO is currently full.
11:9	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
8	EMPTY	RO	1	FIFO Empty
				When set, this bit indicates that the FIFO is currently empty.
7:4	HPTR	RO	0x0	FIFO Head Pointer
				This field contains the current "head" pointer index for the FIFO, that is, the next entry to be written.
3:0	TPTR	RO	0x0	FIFO Tail Pointer
				This field contains the current "tail" pointer index for the FIFO, that is, the next entry to be read.

Register 21: ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1), offset 0x060

Register 22: ADC Sample Sequence Input Multiplexer Select 2 (ADCSSMUX2), offset 0x080

This register defines the analog input configuration for each sample in a sequence executed with Sample Sequencer 1 or 2. These registers are 16-bits wide and contain information for four possible samples. See the **ADCSSMUX0** register on page 388 for detailed bit descriptions. The **ADCSSMUX1** register affects Sample Sequencer 1 and the **ADCSSMUX2** register affects Sample Sequencer 2.

ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1)

Base 0x4003.8000 Offset 0x060

Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					1			rese	rved							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved		М	JX3	rese	rved	MU	JX2	rese	rved	MU	JX1	rese	rved	MU	JX0
Type	RO	RO	R/W	R/W	RO	RO	R/W	R/W	RO	RO	R/W	R/W	RO	RO	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31:14	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13:12	MUX3	R/W	0x0	4th Sample Input Select
11:10	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9:8	MUX2	R/W	0x0	3rd Sample Input Select
7:6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:4	MUX1	R/W	0x0	2nd Sample Input Select
3:2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1:0	MUX0	R/W	0x0	1st Sample Input Select

Register 23: ADC Sample Sequence Control 1 (ADCSSCTL1), offset 0x064 Register 24: ADC Sample Sequence Control 2 (ADCSSCTL2), offset 0x084

These registers contain the configuration information for each sample for a sequence executed with Sample Sequencer 1 or 2. When configuring a sample sequence, the END bit must be set at some point, whether it be after the first sample, last sample, or any sample in between. These registers are 16-bits wide and contain information for four possible samples. See the **ADCSSCTL0** register on page 390 for detailed bit descriptions. The **ADCSSCTL1** register configures Sample Sequencer 1 and the **ADCSSCTL2** register configures Sample Sequencer 2.

ADC Sample Sequence Control 1 (ADCSSCTL1)

Base 0x4003.8000 Offset 0x064

Type R/W, reset 0x0000.0000

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
		•	'					rese	rved) 		'				
Type *Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0			
Reset																			
Γ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Į	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0			
Type Reset	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0			
В	Bit/Field		Nam	ne	Ту	ре	Reset	Des	cription										
	31:16		reserv	/ed	R	0	0x0000	Soft	Software should not rely on the value of a reserved bit. To provide										
								com	patibility	with futu	ıre produ	ucts, the	value of	a reserv	ed bit sh				
								preserved across a read-modify-write operation.											
	15		TS	3	R/	W	0	4th Sample Temp Sensor Select											
								Sam	Same definition as TS7 but used during the fourth sample.										
	14		IE3	,	R/	۱۸/	0	1th	Sample	Interrupt	Enable								
	14		IES)	r./	vv	U			interrupt tion as I		ead duri	na the fo	urth cam	nnla				
								Jan	ie deiliili	lion as 1	E / Dut u	seu uum	ing the lo	urur san	ipic.				
	13		END)3	R/	W	0	4th	Sample	is End of	Sequen	ice							
								Sam	ne definit	tion as E	ND7 but	used du	ring the t	fourth sa	imple.				
	12		D3		R/	W	0	4th	Sample	Diff Input	Select								
								Sam	ne definit	ti on as D	7 but us	ed during	g the fou	rth samp	ole.				
	11		TS	n	R/		0	Ord	Campla	Town Co	noor Co	loot							
	11		132	2	K/	VV	U		•	Temp Setion as T			na the th	ird camr	مام				
								Jan	ie deiliili	lion as 1	5 / Dul u	seu uum	ing the th	iiu sairip	Jic.				
	10		IE2	2	R/	W	0			Interrupt									
								San	ne definit	tion as I	E7 but u	sed duri	ng the th	ird samp	ole.				
	9		END)2	R/	W	0	3rd	Sample	is End of	Sequen	ice							
								Sam	ne definit	tion as E	ND7 but	used du	ring the t	third san	nple.				
	8		D2		R/	۱۸/	0	2rd	Sampla	Diff Input	t Salact								
	0		DZ		K/	v v	U		•	tion as D		ed durin	n the thir	d sample	e				
								Gan	ic delilli	ion as D	, but us	ca aann	9 410 4111	a sampi	.				
	7		TS	1	R/	W	0			Temp Se									
								Same definition as TS7 but used during the second sample.											

Bit/Field	Name	Туре	Reset	Description
6	IE1	R/W	0	2nd Sample Interrupt Enable Same definition as IE7 but used during the second sample.
5	END1	R/W	0	2nd Sample is End of Sequence Same definition as END7 but used during the second sample.
4	D1	R/W	0	2nd Sample Diff Input Select Same definition as D7 but used during the second sample.
3	TS0	R/W	0	1st Sample Temp Sensor Select Same definition as TS7 but used during the first sample.
2	IE0	R/W	0	1st Sample Interrupt Enable Same definition as IE7 but used during the first sample.
1	END0	R/W	0	1st Sample is End of Sequence Same definition as END7 but used during the first sample.
0	D0	R/W	0	1st Sample Diff Input Select Same definition as D7 but used during the first sample.

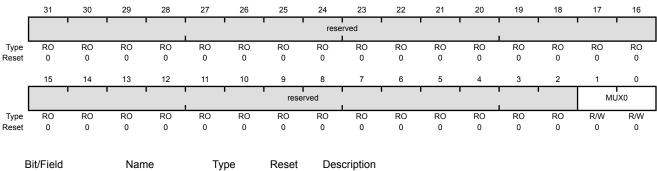
Register 25: ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3), offset 0x0A0

This register defines the analog input configuration for a sample executed with Sample Sequencer 3. This register is 4-bits wide and contains information for one possible sample. See the ADCSSMUX0 register on page 388 for detailed bit descriptions.

ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3)

Base 0x4003.8000

Offset 0x0A0
Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1:0	MUX0	R/W	0	1st Sample Input Select

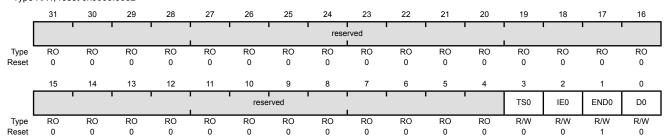
Register 26: ADC Sample Sequence Control 3 (ADCSSCTL3), offset 0x0A4

This register contains the configuration information for a sample executed with Sample Sequencer 3. The END bit is always set since there is only one sample in this sequencer. This register is 4-bits wide and contains information for one possible sample. See the **ADCSSCTL0** register on page 390 for detailed bit descriptions.

ADC Sample Sequence Control 3 (ADCSSCTL3)

Base 0x4003.8000

Offset 0x0A4
Type R/W, reset 0x0000.0002



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TS0	R/W	0	1st Sample Temp Sensor Select Same definition as TS7 but used during the first sample.
2	IE0	R/W	0	1st Sample Interrupt Enable Same definition as IE7 but used during the first sample.
1	END0	R/W	1	1st Sample is End of Sequence Same definition as END7 but used during the first sample. Since this sequencer has only one entry, this bit must be set.
0	D0	R/W	0	1st Sample Diff Input Select Same definition as D7 but used during the first sample.

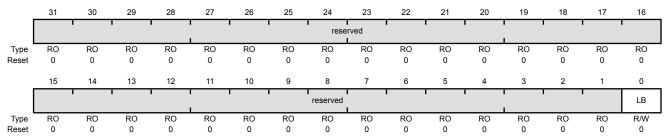
Register 27: ADC Test Mode Loopback (ADCTMLB), offset 0x100

This register provides loopback operation within the digital logic of the ADC, which can be useful in debugging software without having to provide actual analog stimulus. This test mode is entered by writing a value of 0x0000.0001 to this register. When data is read from the FIFO in loopback mode, the read-only portion of this register is returned.

ADC Test Mode Loopback (ADCTMLB)

Base 0x4003.8000

Offset 0x100 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	LB	R/W	0	Loopback Mode Enable

When set, forces a loopback within the digital block to provide information on input and unique numbering. The **ADCSSFIFOn** registers do not provide sample data, but instead provide the 10-bit loopback data as shown below.

Bit/Field	Name	Description
9:6	CNT	Continuous Sample Counter
		Continuous sample counter that is initialized to 0 and counts each sample as it processed. This helps provide a unique value for the data received.
5	CONT	Continuation Sample Indicator
		When set, indicates that this is a continuation sample. For example, if two sequencers were to run back-to-back, this indicates that the controller kept continuously sampling at full rate.
4	DIFF	Differential Sample Indicator
		When set, indicates that this is a differential sample.
3	TS	Temp Sensor Sample Indicator
		When set, indicates that this is a temperature sensor sample.
2:0	MUX	Analog Input Indicator
		Indicates which analog input is to be sampled.

11 Universal Asynchronous Receivers/Transmitters (UARTs)

Each Stellaris® Universal Asynchronous Receiver/Transmitter (UART) has the following features:

- Two fully programmable 16C550-type UARTs with IrDA support
- Separate 16x8 transmit (TX) and receive (RX) FIFOs to reduce CPU interrupt service loading
- Programmable baud-rate generator allowing speeds up to 3.125 Mbps
- Programmable FIFO length, including 1-byte deep operation providing conventional double-buffered interface
- FIFO trigger levels of 1/8, 1/4, 1/2, 3/4, and 7/8
- Standard asynchronous communication bits for start, stop, and parity
- Line-break generation and detection
- Fully programmable serial interface characteristics
 - 5, 6, 7, or 8 data bits
 - Even, odd, stick, or no-parity bit generation/detection
 - 1 or 2 stop bit generation
- IrDA serial-IR (SIR) encoder/decoder providing
 - Programmable use of IrDA Serial Infrared (SIR) or UART input/output
 - Support of IrDA SIR encoder/decoder functions for data rates up to 115.2 Kbps half-duplex
 - Support of normal 3/16 and low-power (1.41-2.23 μs) bit durations
 - Programmable internal clock generator enabling division of reference clock by 1 to 256 for low-power mode bit duration

11.1 Block Diagram

Figure 11-1. UART Module Block Diagram

System Clock Interrupt **TxFIFO Interrupt Control** 16 x 8 **UARTIFLS** UARTIM **UARTMIS** Identification **UARTRIS** Registers **UARTICR** Transmitter UARTPCellID0 UnTx (with SIR UARTPCellID1 Transmit **Baud Rate** UARTPCellID2 Encoder) Generator UARTPCellID3 **UARTDR** UARTPeriphID0 **UARTIBRD** UARTPeriphID1 UARTFBRD Receiver UnRx UARTPeriphID2 (with SIR UARTPeriphID3 Receive UARTPeriphID4 Decoder) **RxFIFO** Control/Status UARTPeriphID5 16 x 8 UARTPeriphID6 UARTPeriphID7 UARTRSR/ECR **UARTFR** UARTLCRH UARTCTL **UARTILPR**

11.2 Signal Description

Table 11-1 on page 402 and Table 11-2 on page 403 list the external signals of the UART module and describe the function of each. The UART signals are alternate functions for some GPIO signals and default to be GPIO signals at reset, with the exception of the U0Rx and U0Tx pins which default to the UART function. The column in the table below titled "Pin Assignment" lists the possible GPIO pin placements for these UART signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the UART function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 11-1. UART Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
U0Rx	26	I		UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
UOTx	27	0		UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
Ulrx	12	I		UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.

Table 11-1. UART Signals (100LQFP) (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
UlTx	13	0		UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 11-2. UART Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
UORx	L3	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
UOTx	M3	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
UlRx	H2	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
UlTx	H1	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

11.3 Functional Description

Each Stellaris UART performs the functions of parallel-to-serial and serial-to-parallel conversions. It is similar in functionality to a 16C550 UART, but is not register compatible.

The UART is configured for transmit and/or receive via the TXE and RXE bits of the **UART Control** (**UARTCTL**) register (see page 421). Transmit and receive are both enabled out of reset. Before any control registers are programmed, the UART must be disabled by clearing the UARTEN bit in **UARTCTL**. If the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

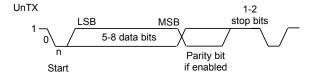
The UART peripheral also includes a serial IR (SIR) encoder/decoder block that can be connected to an infrared transceiver to implement an IrDA SIR physical layer. The SIR function is programmed using the UARTCTL register.

11.3.1 Transmit/Receive Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit, and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration in the control registers. See Figure 11-2 on page 403 for details.

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and their status accompanies the data that is written to the receive FIFO.

Figure 11-2. UART Character Frame



11.3.2 Baud-Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit period. Having a fractional baud-rate divider allows the UART to generate all the standard baud rates.

The 16-bit integer is loaded through the **UART Integer Baud-Rate Divisor (UARTIBRD)** register (see page 417) and the 6-bit fractional part is loaded with the **UART Fractional Baud-Rate Divisor (UARTFBRD)** register (see page 418). The baud-rate divisor (BRD) has the following relationship to the system clock (where *BRDI* is the integer part of the *BRD* and *BRDF* is the fractional part, separated by a decimal place.)

```
BRD = BRDI + BRDF = UARTSysClk / (16 * Baud Rate)
```

where UARTSysClk is the system clock connected to the UART.

The 6-bit fractional number (that is to be loaded into the DIVFRAC bit field in the **UARTFBRD** register) can be calculated by taking the fractional part of the baud-rate divisor, multiplying it by 64, and adding 0.5 to account for rounding errors:

```
UARTFBRD[DIVFRAC] = integer(BRDF * 64 + 0.5)
```

The UART generates an internal baud-rate reference clock at 16x the baud-rate (referred to as Baud16). This reference clock is divided by 16 to generate the transmit clock, and is used for error detection during receive operations.

Along with the **UART Line Control**, **High Byte (UARTLCRH)** register (see page 419), the **UARTIBRD** and **UARTFBRD** registers form an internal 30-bit register. This internal register is only updated when a write operation to **UARTLCRH** is performed, so any changes to the baud-rate divisor must be followed by a write to the **UARTLCRH** register for the changes to take effect.

To update the baud-rate registers, there are four possible sequences:

- UARTIBRD write, UARTFBRD write, and UARTLCRH write
- UARTFBRD write, UARTIBRD write, and UARTLCRH write
- UARTIBRD write and UARTLCRH write
- UARTFBRD write and UARTLCRH write

11.3.3 Data Transmission

Data received or transmitted is stored in two 16-byte FIFOs, though the receive FIFO has an extra four bits per character for status information. For transmission, data is written into the transmit FIFO. If the UART is enabled, it causes a data frame to start transmitting with the parameters indicated in the **UARTLCRH** register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the **UART Flag (UARTFR)** register (see page 414) is asserted as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit is negated only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. The UART can indicate that it is busy even though the UART may no longer be enabled.

When the receiver is idle (the UnRx is continuously 1) and the data input goes Low (a start bit has been received), the receive counter begins running and data is sampled on the eighth cycle of Baud16 (described in "Transmit/Receive Logic" on page 403).

The start bit is valid and recognized if UnRx is still low on the eighth cycle of Baud16, otherwise it is ignored. After a valid start bit is detected, successive data bits are sampled on every 16th cycle of Baud16 (that is, one bit period later) according to the programmed length of the data characters. The parity bit is then checked if parity mode was enabled. Data length and parity are defined in the **UARTLCRH** register.

Lastly, a valid stop bit is confirmed if UnRx is High, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO, with any error bits associated with that word.

11.3.4 **Serial IR (SIR)**

The UART peripheral includes an IrDA serial-IR (SIR) encoder/decoder block. The IrDA SIR block provides functionality that converts between an asynchronous UART data stream, and half-duplex serial SIR interface. No analog processing is performed on-chip. The role of the SIR block is to provide a digital encoded output and decoded input to the UART. The UART signal pins can be connected to an infrared transceiver to implement an IrDA SIR physical layer link. The SIR block has two modes of operation:

- In normal IrDA mode, a zero logic level is transmitted as high pulse of 3/16th duration of the selected baud rate bit period on the output pin, while logic one levels are transmitted as a static LOW signal. These levels control the driver of an infrared transmitter, sending a pulse of light for each zero. On the reception side, the incoming light pulses energize the photo transistor base of the receiver, pulling its output LOW. This drives the UART input pin LOW.
- In low-power IrDA mode, the width of the transmitted infrared pulse is set to three times the period of the internally generated IrLPBaud16 signal (1.63 µs, assuming a nominal 1.8432 MHz frequency) by changing the appropriate bit in the **UARTCR** register. See page 416 for more information on IrDA low-power pulse-duration configuration.

Figure 11-3 on page 405 shows the UART transmit and receive signals, with and without IrDA modulation.

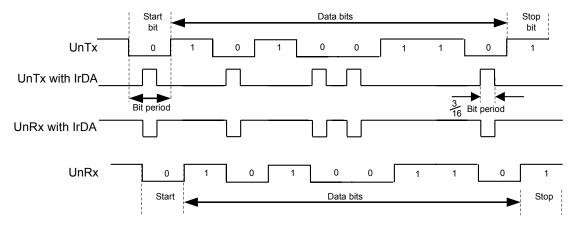


Figure 11-3. IrDA Data Modulation

In both normal and low-power IrDA modes:

- During transmission, the UART data bit is used as the base for encoding
- During reception, the decoded bits are transferred to the UART receive logic

The IrDA SIR physical layer specifies a half-duplex communication link, with a minimum 10 ms delay between transmission and reception. This delay must be generated by software because it is not automatically supported by the UART. The delay is required because the infrared receiver electronics might become biased, or even saturated from the optical power coupled from the adjacent transmitter LED. This delay is known as latency, or receiver setup time.

If the application does not require the use of the UnRx signal, the GPIO pin that has the UnRx signal as an alternate function must be configured as the UnRx signal and pulled High.

11.3.5 FIFO Operation

The UART has two 16-entry FIFOs; one for transmit and one for receive. Both FIFOs are accessed via the **UART Data (UARTDR)** register (see page 410). Read operations of the **UARTDR** register return a 12-bit value consisting of 8 data bits and 4 error flags while write operations place 8-bit data in the transmit FIFO.

Out of reset, both FIFOs are disabled and act as 1-byte-deep holding registers. The FIFOs are enabled by setting the FEN bit in **UARTLCRH** (page 419).

FIFO status can be monitored via the **UART Flag (UARTFR)** register (see page 414) and the **UART Receive Status (UARTRSR)** register. Hardware monitors empty, full and overrun conditions. The **UARTFR** register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits) and the **UARTRSR** register shows overrun status via the OE bit.

The trigger points at which the FIFOs generate interrupts is controlled via the **UART Interrupt FIFO Level Select (UARTIFLS)** register (see page 423). Both FIFOs can be individually configured to trigger interrupts at different levels. Available configurations include 1/8, ½, ½, ¾, and 7/8. For example, if the ¼ option is selected for the receive FIFO, the UART generates a receive interrupt after 4 data bytes are received. Out of reset, both FIFOs are configured to trigger an interrupt at the ½ mark.

11.3.6 Interrupts

The UART can generate interrupts when the following conditions are observed:

- Overrun Error
- Break Error
- Parity Error
- Framing Error
- Receive Timeout
- Transmit (when condition defined in the TXIFLSEL bit in the **UARTIFLS** register is met)
- Receive (when condition defined in the RXIFLSEL bit in the **UARTIFLS** register is met)

All of the interrupt events are ORed together before being sent to the interrupt controller, so the UART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine by reading the **UART Masked Interrupt Status (UARTMIS)** register (see page 428).

The interrupt events that can trigger a controller-level interrupt are defined in the **UART Interrupt**Mask (UARTIM) register (see page 425) by setting the corresponding IM bit to 1. If interrupts are

not used, the raw interrupt status is always visible via the **UART Raw Interrupt Status (UARTRIS)** register (see page 427).

Interrupts are always cleared (for both the **UARTMIS** and **UARTRIS** registers) by setting the corresponding bit in the **UART Interrupt Clear (UARTICR)** register (see page 429).

The receive interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed trigger level, the RXRIS bit is set. The receive interrupt is cleared by reading data from the receive FIFO until it becomes less than the trigger level, or by clearing the interrupt by writing a 1 to the RXIC bit.
- If the FIFOs are disabled (have a depth of one location) and data is received thereby filling the location, the RXRIS bit is set. The receive interrupt is cleared by performing a single read of the receive FIFO, or by clearing the interrupt by writing a 1 to the RXIC bit.

The transmit interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the transmit FIFO progresses through the programmed trigger level, the TXRIS bit is set. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts will be generated. The transmit interrupt is cleared by writing data to the transmit FIFO until it becomes greater than the trigger level, or by clearing the interrupt by writing a 1 to the TXIC bit.
- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitters single location, the TXRIS bit is set. It is cleared by performing a single write to the transmit FIFO, or by clearing the interrupt by writing a 1 to the TXIC bit.

11.3.7 Loopback Operation

The UART can be placed into an internal loopback mode for diagnostic or debug work. This is accomplished by setting the LBE bit in the **UARTCTL** register (see page 421). In loopback mode, data transmitted on UnTx is received on the UnRx input.

11.3.8 IrDA SIR block

The IrDA SIR block contains an IrDA serial IR (SIR) protocol encoder/decoder. When enabled, the SIR block uses the \mathtt{UnTx} and \mathtt{UnRx} pins for the SIR protocol, which should be connected to an IR transceiver.

The SIR block can receive and transmit, but it is only half-duplex so it cannot do both at the same time. Transmission must be stopped before data can be received. The IrDA SIR physical layer specifies a minimum 10-ms delay between transmission and reception.

11.4 Initialization and Configuration

To use the UARTs, the peripheral clock must be enabled by setting the $\mathtt{UART0}$ or $\mathtt{UART1}$ bits in the $\mathtt{RCGC1}$ register.

This section discusses the steps that are required to use a UART module. For this example, the UART clock is assumed to be 20 MHz and the desired UART configuration is:

- 115200 baud rate
- Data length of 8 bits
- One stop bit

- No parity
- FIFOs disabled
- No interrupts

The first thing to consider when programming the UART is the baud-rate divisor (BRD), since the **UARTIBRD** and **UARTFBRD** registers must be written before the **UARTLCRH** register. Using the equation described in "Baud-Rate Generation" on page 404, the BRD can be calculated:

```
BRD = 20,000,000 / (16 * 115,200) = 10.8507
```

which means that the DIVINT field of the **UARTIBRD** register (see page 417) should be set to 10. The value to be loaded into the **UARTFBRD** register (see page 418) is calculated by the equation:

```
UARTFBRD[DIVFRAC] = integer(0.8507 * 64 + 0.5) = 54
```

With the BRD values in hand, the UART configuration is written to the module in the following order:

- 1. Disable the UART by clearing the UARTEN bit in the UARTCTL register.
- 2. Write the integer portion of the BRD to the **UARTIBRD** register.
- 3. Write the fractional portion of the BRD to the **UARTFBRD** register.
- **4.** Write the desired serial parameters to the **UARTLCRH** register (in this case, a value of 0x0000.0060).
- 5. Enable the UART by setting the UARTEN bit in the UARTCTL register.

11.5 Register Map

Table 11-3 on page 408 lists the UART registers. The offset listed is a hexadecimal increment to the register's address, relative to that UART's base address:

UART0: 0x4000.C000UART1: 0x4000.D000

Note that the UART module clock must be enabled before the registers can be programmed (see page 215). There must be a delay of 3 system clocks after the UART module clock is enabled before any UART module registers are accessed.

Note: The UART must be disabled (see the UARTEN bit in the **UARTCTL** register on page 421) before any of the control registers are reprogrammed. When the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

Table 11-3. UART Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	UARTDR	R/W	0x0000.0000	UART Data	410
0x004	UARTRSR/UARTECR	R/W	0x0000.0000	UART Receive Status/Error Clear	412
0x018	UARTFR	RO	0x0000.0090	UART Flag	414

Table 11-3. UART Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0x020	UARTILPR	R/W	0x0000.0000	UART IrDA Low-Power Register	416
0x024	UARTIBRD	R/W	0x0000.0000	UART Integer Baud-Rate Divisor	417
0x028	UARTFBRD	R/W	0x0000.0000	UART Fractional Baud-Rate Divisor	418
0x02C	UARTLCRH	R/W	0x0000.0000	UART Line Control	419
0x030	UARTCTL	R/W	0x0000.0300	UART Control	421
0x034	UARTIFLS	R/W	0x0000.0012	UART Interrupt FIFO Level Select	423
0x038	UARTIM	R/W	0x0000.0000	UART Interrupt Mask	425
0x03C	UARTRIS	RO	0x0000.0000	UART Raw Interrupt Status	427
0x040	UARTMIS	RO	0x0000.0000	UART Masked Interrupt Status	428
0x044	UARTICR	W1C	0x0000.0000	UART Interrupt Clear	429
0xFD0	UARTPeriphID4	RO	0x0000.0000	UART Peripheral Identification 4	431
0xFD4	UARTPeriphID5	RO	0x0000.0000	UART Peripheral Identification 5	432
0xFD8	UARTPeriphID6	RO	0x0000.0000	UART Peripheral Identification 6	433
0xFDC	UARTPeriphID7	RO	0x0000.0000	UART Peripheral Identification 7	434
0xFE0	UARTPeriphID0	RO	0x0000.0011	UART Peripheral Identification 0	435
0xFE4	UARTPeriphID1	RO	0x0000.0000	UART Peripheral Identification 1	436
0xFE8	UARTPeriphID2	RO	0x0000.0018	UART Peripheral Identification 2	437
0xFEC	UARTPeriphID3	RO	0x0000.0001	UART Peripheral Identification 3	438
0xFF0	UARTPCellID0	RO	0x0000.000D	UART PrimeCell Identification 0	439
0xFF4	UARTPCellID1	RO	0x0000.00F0	UART PrimeCell Identification 1	440
0xFF8	UARTPCellID2	RO	0x0000.0005	UART PrimeCell Identification 2	441
0xFFC	UARTPCellID3	RO	0x0000.00B1	UART PrimeCell Identification 3	442

11.6 Register Descriptions

The remainder of this section lists and describes the UART registers, in numerical order by address offset.

Register 1: UART Data (UARTDR), offset 0x000

Important: This register is read-sensitive. See the register description for details.

This register is the data register (the interface to the FIFOs).

When FIFOs are enabled, data written to this location is pushed onto the transmit FIFO. If FIFOs are disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.

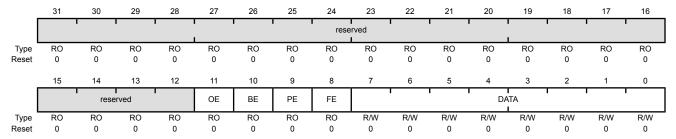
For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If FIFOs are disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

UART Data (UARTDR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000

Offset 0x000

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11	OE	RO	0	UART Overrun Error
				The OE values are defined as follows:
				Value Description
				0 There has been no data loss due to a FIFO overrun.
				New data was received when the FIFO was full, resulting in data loss.
10	BE	RO	0	UART Break Error
				This bit is not to 1 when a break condition is detected indicating that

This bit is set to 1 when a break condition is detected, indicating that the receive data input was held I ow for longer than a full-word

the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).

In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the received data input goes to a 1 (marking state) and the next valid start bit is received.

Bit/Field	Name	Туре	Reset	Description
9	PE	RO	0	UART Parity Error This bit is set to 1 when the parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. In FIFO mode, this error is associated with the character at the top of the FIFO.
8	FE	RO	0	UART Framing Error This bit is set to 1 when the received character does not have a valid stop bit (a valid stop bit is 1).
7:0	DATA	R/W	0	Data Transmitted or Received When written, the data that is to be transmitted via the UART. When read, the data that was received by the UART.

Register 2: UART Receive Status/Error Clear (UARTRSR/UARTECR), offset 0x004

The **UARTRSR/UARTECR** register is the receive status register/error clear register.

In addition to the **UARTDR** register, receive status can also be read from the **UARTRSR** register. If the status is read from this register, then the status information corresponds to the entry read from **UARTDR** prior to reading **UARTRSR**. The status information for overrun is set immediately when an overrun condition occurs.

The **UARTRSR** register cannot be written.

A write of any value to the **UARTECR** register clears the framing, parity, break, and overrun errors. All the bits are cleared to 0 on reset.

Reads

UART Receive Status/Error Clear (UARTRSR/UARTECR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x004

Type RO, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	1	ı	ı		1	rese	rved) 			
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												OE	BE	PE	FE
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bit/Field	Name	Type	Reset	Description
31:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	OE	RO	0	UART Overrun Error
				When this bit is set to 1, data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR .
				The FIFO contents remain valid since no further data is written when the FIFO is full, only the contents of the shift register are overwritten. The CPU must now read the data in order to empty the FIFO.
2	BE	RO	0	UART Break Error

This bit is set to 1 when a break condition is detected, indicating that the received data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).

This bit is cleared to 0 by a write to **UARTECR**.

In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.

Bit/Field	Name	Туре	Reset	Description
1	PE	RO	0	UART Parity Error
				This bit is set to 1 when the parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register.
				This bit is cleared to 0 by a write to UARTECR .
0	FE	RO	0	UART Framing Error
				This bit is set to 1 when the received character does not have a valid stop bit (a valid stop bit is 1).

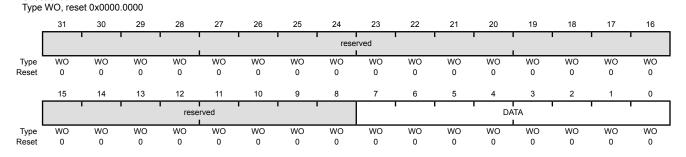
This bit is cleared to 0 by a write to **UARTECR**.

In FIFO mode, this error is associated with the character at the top of the FIFO.

Writes

UART Receive Status/Error Clear (UARTRSR/UARTECR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x004



Bit/Field	Name	Type	Reset	Description
31:8	reserved	WO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DATA	WO	0	Error Clear

 $\boldsymbol{\mathsf{A}}$ write to this register of any data clears the framing, parity, break, and overrun flags.

Register 3: UART Flag (UARTFR), offset 0x018

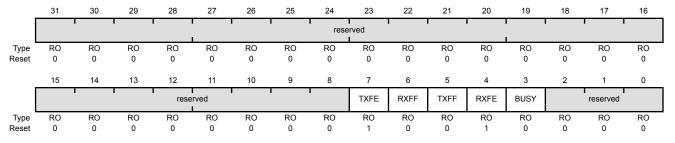
The **UARTFR** register is the flag register. After reset, the TXFF, RXFF, and BUSY bits are 0, and TXFE and RXFE bits are 1.

UART Flag (UARTFR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000

Offset 0x018

Type RO, reset 0x0000.0090



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7	TXFE	RO	1	UART Transmit FIFO Empty
				The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
				If the FIFO is disabled (FEN is 0), this bit is set when the transmit holding register is empty.
				If the FIFO is enabled (FEN is 1), this bit is set when the transmit FIFO is empty.
6	RXFF	RO	0	UART Receive FIFO Full
				The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
				If the FIFO is disabled, this bit is set when the receive holding register is full.
				If the FIFO is enabled, this bit is set when the receive FIFO is full.
5	TXFF	RO	0	UART Transmit FIFO Full
				The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
				If the FIFO is disabled, this bit is set when the transmit holding register is full.
				If the FIFO is enabled, this bit is set when the transmit FIFO is full.
4	RXFE	RO	1	UART Receive FIFO Empty
				The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register.
				If the FIFO is disabled, this bit is set when the receive holding register is empty.
				KU 5150: 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

If the FIFO is enabled, this bit is set when the receive FIFO is empty.

Bit/Field	Name	Туре	Reset	Description
3	BUSY	RO	0	UART Busy When this bit is 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register.
				This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled).
2:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 4: UART IrDA Low-Power Register (UARTILPR), offset 0x020

The **UARTILPR** register is an 8-bit read/write register that stores the low-power counter divisor value used to derive the low-power SIR pulse width clock by dividing down the system clock (SysClk). All the bits are cleared to 0 when reset.

The internal IrlPBaud16 clock is generated by dividing down SysClk according to the low-power divisor value written to **UARTILPR**. The duration of SIR pulses generated when low-power mode is enabled is three times the period of the IrlPBaud16 clock. The low-power divisor value is calculated as follows:

ILPDVSR = SysClk / F_{IrLPBaud16}

where $F_{IrLPBaud16}$ is nominally 1.8432 MHz.

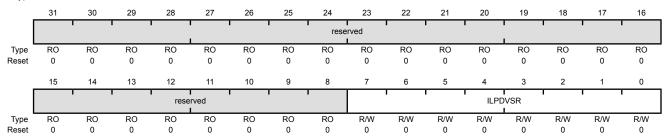
You must choose the divisor so that $1.42\,\mathrm{MHz} < \mathrm{F}_{\mathtt{IrlPBaud16}} < 2.12\,\mathrm{MHz}$, which results in a low-power pulse duration of $1.41-2.11\,\mu s$ (three times the period of $\mathtt{IrlPBaud16}$). The minimum frequency of $\mathtt{IrlPBaud16}$ ensures that pulses less than one period of $\mathtt{IrlPBaud16}$ are rejected, but that pulses greater than $1.4\,\mu s$ are accepted as valid pulses.

Note: Zero is an illegal value. Programming a zero value results in no IrLPBaud16 pulses being generated.

UART IrDA Low-Power Register (UARTILPR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x020

Type R/W, reset 0x0000.0000



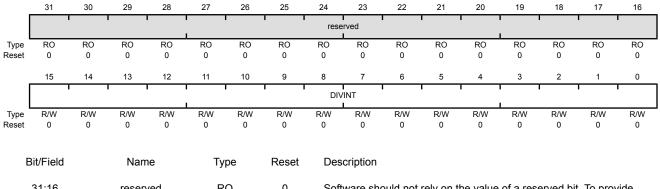
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	ILPDVSR	R/W	0x00	IrDA Low-Power Divisor
				This is an 8-bit low-power divisor value.

Register 5: UART Integer Baud-Rate Divisor (UARTIBRD), offset 0x024

The **UARTIBRD** register is the integer part of the baud-rate divisor value. All the bits are cleared on reset. The minimum possible divide ratio is 1 (when **UARTIBRD**=0), in which case the **UARTFBRD** register is ignored. When changing the **UARTIBRD** register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the **UARTLCRH** register. See "Baud-Rate Generation" on page 404 for configuration details.

UART Integer Baud-Rate Divisor (UARTIBRD)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x024 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	DIVINT	R/W	0x0000	Integer Baud-Rate Divisor

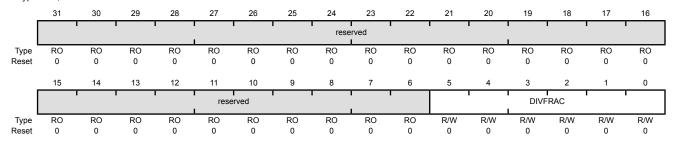
Register 6: UART Fractional Baud-Rate Divisor (UARTFBRD), offset 0x028

The **UARTFBRD** register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the UARTFBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the **UARTLCRH** register. See "Baud-Rate Generation" on page 404 for configuration details.

UART Fractional Baud-Rate Divisor (UARTFBRD)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x028

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	DIVFRAC	R/W	0x000	Fractional Baud-Rate Divisor

Register 7: UART Line Control (UARTLCRH), offset 0x02C

The **UARTLCRH** register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.

When updating the baud-rate divisor (UARTIBRD and/or UARTIFRD), the UARTLCRH register must also be written. The write strobe for the baud-rate divisor registers is tied to the UARTLCRH register.

reserved

UART Line Control (UARTLCRH)

28

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x02C

31

3

STP2

R/W

0

Type R/W, reset 0x0000.0000

Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO 0	RO	RO	RO	RO	RO			
Reset	0	0	0	0	0	0	0	0	0	0	U	0	0	0	0	0			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
			reserved						SPS	WL	EN	FEN	STP2	EPS	PEN	BRK			
Type	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W	R/W 0	R/W 0	R/W			
Reset	U	U	U	U	U	U	U	U	U	U	U	U	0	U	U	0			
_):4/E: _1_		Na		т.		Deset	Description											
	Bit/Field		Nar	iie	Ту	pe	Reset	Des	cription										
	31:8		reser	ved	R	0	0	Soft	ware sho	ould not	rely on t	he value	of a rese	erved bit	. To prov	ride			
									patibility		•				ed bit sh	ould be			
								pres	served a	cross a r	ead-mod	airy-write	operatio	n.					
	7		SP	S	R/	R/W 0			UART Stick Parity Select										
									en bits 1,										
									checked					t and 2 i	s cleared	d, the			
									ty bit is ti					J					
								VVIIE	en this bi	l is cleai	rea, suck	panty is	disable	J.					
	6:5		WLI	ΞN	R/	W	0	UART Word Length											
									bits indi		number	of data b	oits trans	mitted o	r receive	d in a			
								fran	ne as foll	ows:									
								Val	ue Desc	ription									
								0x3 8 bits											
								0x2 7 bits											
								0x1 6 bits											
								0x	0 5 bits	(defaul	t)								
										`	,								
	4		FE	N	R/	W	0	UAF	RT Enabl	e FIFOs	;								
	•			1.0	- •	J	If this bit is set to 1, transmit and receive FIFO buffers are enable						d (FIFO						
								mode).					oro ur	5 5114510					
								When cleared to 0, FIFOs are disabled (Character mode). The FIFOs								FIFOs			
								bec	ome 1-by	/te-deep	holding	registers	S .						

UART Two Stop Bits Select

If this bit is set to 1, two stop bits are transmitted at the end of a frame. The receive logic does not check for two stop bits being received.

Bit/Field	Name	Туре	Reset	Description
2	EPS	R/W	0	UART Even Parity Select
				If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits.
				When cleared to 0, then odd parity is performed, which checks for an odd number of 1s.
				This bit has no effect when parity is disabled by the ${\tt PEN}$ bit.
1	PEN	R/W	0	UART Parity Enable
				If this bit is set to 1, parity checking and generation is enabled; otherwise, parity is disabled and no parity bit is added to the data frame.
0	BRK	R/W	0	UART Send Break
				If this bit is set to 1, a Low level is continually output on the ${\tt UnTX}$ output, after completing transmission of the current character. For the proper execution of the break command, the software must set this bit for at least two frames (character periods). For normal use, this bit must be cleared to 0.

Register 8: UART Control (UARTCTL), offset 0x030

The **UARTCTL** register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set to 1.

To enable the UART module, the UARTEN bit must be set to 1. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping.

Note: The **UARTCTL** register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the **UARTCTL** register.

- 1. Disable the UART.
- 2. Wait for the end of transmission or reception of the current character.
- 3. Flush the transmit FIFO by disabling bit 4 (FEN) in the line control register (UARTLCRH).
- **4.** Reprogram the control register.
- 5. Enable the UART.

UART Control (UARTCTL)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x030

Type R/W, reset 0x0000.0300

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	'		•	'		'	•	rese	rved				'	•		•
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved							TXE	LBE		rese	rved		SIRLP	SIREN	UARTEN
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 1	R/W 1	R/W 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	R/W 0

pe set	0	0	0	0	0	0	R/W 1	1 1	0 0	0	0	0	0	0 0	0 0	0	
В	it/Field		Nam	ne	Ту	pe	Reset	Desc	cription								
;	31:10		reserv	ved	R	0	0	com	patibility	with futu	ıre produ	ne value ucts, the lify-write	value of	a reserv	•	vide nould be	
	9		RXI	E	R/	W	1	If this	s bit is so JART is	,	he receiv in the m	ve sectio				d. When e current	
								Note	: To	enable	receptio	n, the UA	RTEN bi	t must al	so be se	et.	
	8		TXI	E	R/	W	1	UAR	T Transı	mit Enat	ole						
								the U	JART is	disabled		niddle of				d. When etes the	
								Note	: To	enable	transmis	sion, the	UARTE	N bit mus	st also be	e set.	

Bit/Field	Name	Туре	Reset	Description
7	LBE	R/W	0	UART Loop Back Enable If this bit is set to 1, the ${\tt UnTX}$ path is fed through the ${\tt UnRX}$ path.
6:3	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	SIRLP	R/W	0	UART SIR Low Power Mode
				This bit selects the IrDA encoding mode. If this bit is cleared to 0, low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period. If this bit is set to 1, low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate. Setting this bit uses less power, but might reduce transmission distances. See page 416 for more information.
1	SIREN	R/W	0	UART SIR Enable
				If this bit is set to 1, the IrDA SIR block is enabled, and the UART will transmit and receive data using SIR protocol.
0	UARTEN	R/W	0	UART Enable If this bit is set to 1, the UART is enabled. When the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.

Register 9: UART Interrupt FIFO Level Select (UARTIFLS), offset 0x034

The **UARTIFLS** register is the interrupt FIFO level select register. You can use this register to define the FIFO level at which the TXRIS and RXRIS bits in the UARTRIS register are triggered.

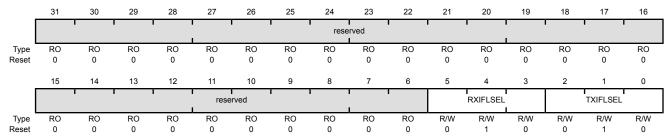
The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered as the module is receiving the 9th character.

Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

UART Interrupt FIFO Level Select (UARTIFLS)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x034

Type R/W, reset 0x0000.0012



Bit/Field	Name	Туре	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:3	RXIFLSEL	R/W	0x2	UART Receive Interrupt FIFO Level Select

The trigger points for the receive interrupt are as follows:

Value	Description
0x0	RX FIFO ≥ 1/8 full
0x1	RX FIFO ≥ ¼ full
0x2	RX FIFO $\geq \frac{1}{2}$ full (default)
0x3	RX FIFO ≥ ¾ full
0x4	RX FIFO ≥ 7/8 full
0x5-0x7	Reserved

Bit/Field	Name	Туре	Reset	Description
2:0	TXIFLSEL	R/W	0x2	UART Transmit Interrupt FIFO Level Select The trigger points for the transmit interrupt are as follows: Value Description 0x0 TX FIFO ≤ ⅓ empty 0x1 TX FIFO ≤ ⅓ empty 0x2 TX FIFO ≤ ⅓ empty (default) 0x3 TX FIFO ≤ ¼ empty 0x4 TX FIFO ≤ ⅓ empty
				0x5-0x7 Reserved

Register 10: UART Interrupt Mask (UARTIM), offset 0x038

The **UARTIM** register is the interrupt mask set/clear register.

On a read, this register gives the current value of the mask on the relevant interrupt. Writing a 1 to a bit allows the corresponding raw interrupt signal to be routed to the interrupt controller. Writing a 0 prevents the raw interrupt signal from being sent to the interrupt controller.

UART Interrupt Mask (UARTIM)

30

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x038

Type R/W, reset 0x0000.0000

								rese	erved									
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0		
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	ľ		reserved			OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM		rese	erved	•		
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	RO 0	RO 0	RO 0	RO 0		
E	Bit/Field		Nam	е	Туре		Reset	Des	Description									
	31:11 10 9		OEIM BEIM			RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.										
						/W	0	UART Overrun Error Interrupt Mask On a read, the current mask for the OEIM interrupt is returned. Setting this bit to 1 promotes the OEIM interrupt to the interrupt controller.										
						/W	0	UART Break Error Interrupt Mask On a read, the current mask for the BEIM interrupt is returned. Setting this bit to 1 promotes the BEIM interrupt to the interrupt controller.										
	8		PEIM			R/W 0			a read, tl	ne currei		Mask for the PI ne PEIM		•				
	7		FEIN	И	R	/W	0	On	a read, tl	ne currei		t Mask for the Fine Fine		•				
	6		RTIM	И	R	/W	0	On	a read, tl	ne currei	nt mask	errupt Ma for the R'	тім inte	•		ontroller.		
	5		TXIN	Л	R	/W	0	On	a read, tl	ne currei		sk for the TI ne TXIM		•		ontroller.		
	4		RXII	М	R	/W	0				rupt Mas	k for the R	хім inte	rrupt is r	eturned.			

Setting this bit to 1 promotes the \mathtt{RXIM} interrupt to the interrupt controller.

Bit/Field	Name	Type	Reset	Description
3:0	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 11: UART Raw Interrupt Status (UARTRIS), offset 0x03C

The **UARTRIS** register is the raw interrupt status register. On a read, this register gives the current raw status value of the corresponding interrupt. A write has no effect.

UART Raw Interrupt Status (UARTRIS)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x03C Type RO, reset 0x0000.0000

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			1					rese	rved							
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			reserved		! I	OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	'	rese	rved	
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bit/Field	Name	Type	Reset	Description
31:11	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
10	OERIS	RO	0	UART Overrun Error Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
9	BERIS	RO	0	UART Break Error Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
8	PERIS	RO	0	UART Parity Error Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
7	FERIS	RO	0	UART Framing Error Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
6	RTRIS	RO	0	UART Receive Time-Out Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
5	TXRIS	RO	0	UART Transmit Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
4	RXRIS	RO	0	UART Receive Raw Interrupt Status Gives the raw interrupt state (prior to masking) of this interrupt.
3:0	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

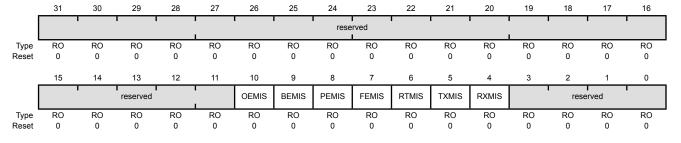
Register 12: UART Masked Interrupt Status (UARTMIS), offset 0x040

The **UARTMIS** register is the masked interrupt status register. On a read, this register gives the current masked status value of the corresponding interrupt. A write has no effect.

UART Masked Interrupt Status (UARTMIS)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x040

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:11	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
10	OEMIS	RO	0	UART Overrun Error Masked Interrupt Status Gives the masked interrupt state of this interrupt.
9	BEMIS	RO	0	UART Break Error Masked Interrupt Status Gives the masked interrupt state of this interrupt.
8	PEMIS	RO	0	UART Parity Error Masked Interrupt Status Gives the masked interrupt state of this interrupt.
7	FEMIS	RO	0	UART Framing Error Masked Interrupt Status Gives the masked interrupt state of this interrupt.
6	RTMIS	RO	0	UART Receive Time-Out Masked Interrupt Status Gives the masked interrupt state of this interrupt.
5	TXMIS	RO	0	UART Transmit Masked Interrupt Status Gives the masked interrupt state of this interrupt.
4	RXMIS	RO	0	UART Receive Masked Interrupt Status Gives the masked interrupt state of this interrupt.
3:0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

19

18

16

Register 13: UART Interrupt Clear (UARTICR), offset 0x044

The **UARTICR** register is the interrupt clear register. On a write of 1, the corresponding interrupt (both raw interrupt and masked interrupt, if enabled) is cleared. A write of 0 has no effect.

UART Interrupt Clear (UARTICR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x044

31

Type W1C, reset 0x0000.0000

30

28

	01	30	20	20	21	20	20	27	20	~~	21	20	10	10	17	10	
	1		1	İ		1	1	rese	rved		1	1			1		
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10 	9	8	7	6	5	4	3	2	1 1	0	
			reserved		ı	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC			erved		
Type Reset	RO 0	RO 0	RO 0	RO 0	RO 0	W1C 0	W1C 0	W1C 0	W1C 0	W1C 0	W1C 0	W1C 0	RO 0	RO 0	RO 0	RO 0	
E	Bit/Field		Nam	ne	Type Res			Description									
	31:11		reserv	ved	R	.0	0x00	0x00 Software should not rely on the value of a reserve compatibility with future products, the value of a									
												ucts, the dify-write			ed bit sh	ould be	
												-	орстан	JII.			
	10		OEIC			1C	0		rrun Erro								
								The	OEIC V	alues are	edefined	as follow	vs:				
								Val	ue Desc	ription							
								0	No e	ffect on t	the inter	upt.					
								1	Clea	rs interru	ıpt.						
	9		BEI	С	W	1C	0	Brea	ak Error	Interrupt	Clear						
								The	BEIC V	alues are	defined	as follow	vs:				
								Val	ue Desc	ription							
								0 No effect on the interrupt.									
								1	Clea	rs interru	ıpt.						
	8		PEI	С	W	1C	0	Parity Error Interrupt Clear									
								The	PEIC V	alues are	defined	as follow	vs:				
								Val	ue Desc	ription							
								0	No e	ffect on t	the interr	upt.					
								1 Clears interrupt.									
	7		FEI	C	\^/	1C	0	Eror	ming Erro	or Interr	ınt Clear						
	,		1 LI	•	۷V	.5	J		_			as follow	vs:				
									_								

Value Description

No effect on the interrupt.

Clears interrupt.

Bit/Field	Name	Type	Reset	Description
6	RTIC	W1C	0	Receive Time-Out Interrupt Clear The RTIC values are defined as follows: Value Description 0 No effect on the interrupt. 1 Clears interrupt.
5	TXIC	W1C	0	Transmit Interrupt Clear The TXIC values are defined as follows: Value Description 0 No effect on the interrupt. 1 Clears interrupt.
4	RXIC	W1C	0	Receive Interrupt Clear The RXIC values are defined as follows: Value Description 0 No effect on the interrupt. 1 Clears interrupt.
3:0	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

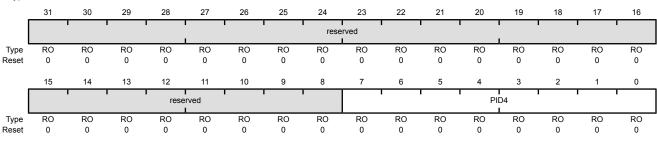
Register 14: UART Peripheral Identification 4 (UARTPeriphID4), offset 0xFD0

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 4 (UARTPeriphID4)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFD0

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x0000	UART Peripheral ID Register[7:0]

Can be used by software to identify the presence of this peripheral.

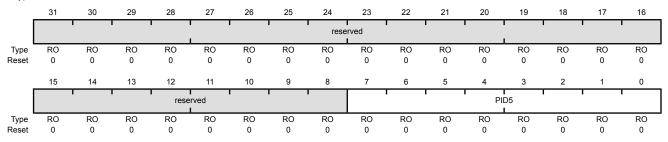
Register 15: UART Peripheral Identification 5 (UARTPeriphID5), offset 0xFD4

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 5 (UARTPeriphID5)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFD4

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID5	RO	0x0000	UART Peripheral ID Register[15:8]

Can be used by software to identify the presence of this peripheral.

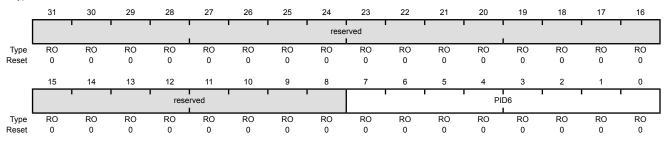
Register 16: UART Peripheral Identification 6 (UARTPeriphID6), offset 0xFD8

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 6 (UARTPeriphID6)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFD8

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID6	RO	0x0000	UART Peripheral ID Register[23:16]

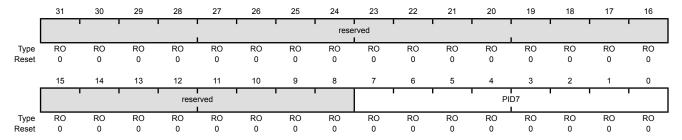
Register 17: UART Peripheral Identification 7 (UARTPeriphID7), offset 0xFDC

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 7 (UARTPeriphID7)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFDC

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID7	RO	0x0000	UART Peripheral ID Register[31:24]

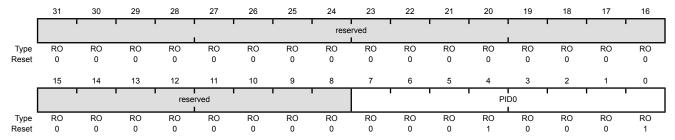
Register 18: UART Peripheral Identification 0 (UARTPeriphID0), offset 0xFE0

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 0 (UARTPeriphID0)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFE0

Type RO, reset 0x0000.0011



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID0	RO	0x11	UART Peripheral ID Register[7:0]

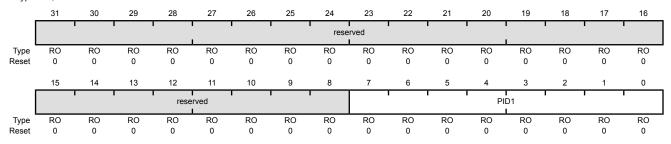
Register 19: UART Peripheral Identification 1 (UARTPeriphID1), offset 0xFE4

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 1 (UARTPeriphID1)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFE4

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID1	RO	0x00	UART Peripheral ID Register[15:8]

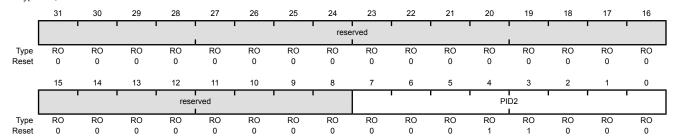
Register 20: UART Peripheral Identification 2 (UARTPeriphID2), offset 0xFE8

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 2 (UARTPeriphID2)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFE8

Type RO, reset 0x0000.0018



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID2	RO	0x18	UART Peripheral ID Register[23:16]

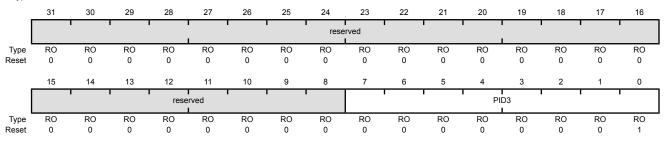
Register 21: UART Peripheral Identification 3 (UARTPeriphID3), offset 0xFEC

The **UARTPeriphIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART Peripheral Identification 3 (UARTPeriphID3)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFEC

Type RO, reset 0x0000.0001



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	UART Peripheral ID Register[31:24]

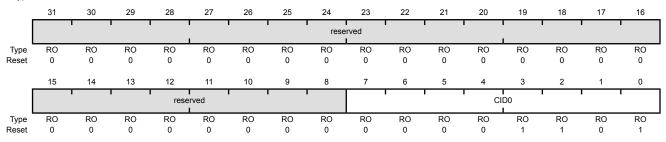
Register 22: UART PrimeCell Identification 0 (UARTPCellID0), offset 0xFF0

The **UARTPCellIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART PrimeCell Identification 0 (UARTPCellID0)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFF0

Type RO, reset 0x0000.000D



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	UART PrimeCell ID Register[7:0]

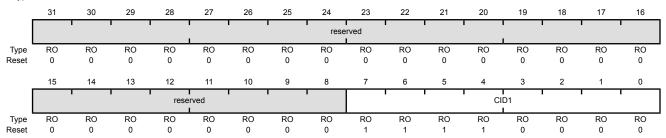
Register 23: UART PrimeCell Identification 1 (UARTPCellID1), offset 0xFF4

The **UARTPCellIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART PrimeCell Identification 1 (UARTPCellID1)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFF4

Type RO, reset 0x0000.00F0



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	UART PrimeCell ID Register[15:8]

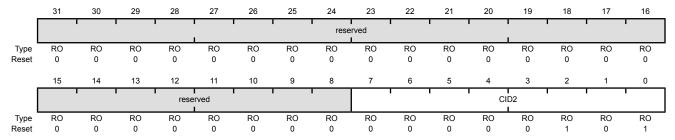
Register 24: UART PrimeCell Identification 2 (UARTPCellID2), offset 0xFF8

The **UARTPCellIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART PrimeCell Identification 2 (UARTPCellID2)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFF8

Offset 0xFF8
Type RO, reset 0x0000.0005



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	UART PrimeCell ID Register[23:16]

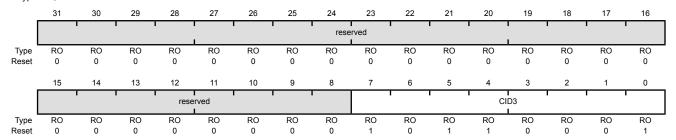
Register 25: UART PrimeCell Identification 3 (UARTPCellID3), offset 0xFFC

The **UARTPCellIDn** registers are hard-coded and the fields within the registers determine the reset values.

UART PrimeCell Identification 3 (UARTPCellID3)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0xFFC

Type RO, reset 0x0000.00B1



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID3	RO	0xB1	UART PrimeCell ID Register[31:24]

12 Synchronous Serial Interface (SSI)

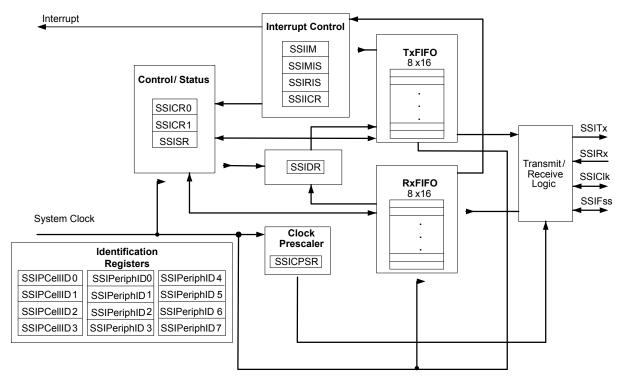
The Stellaris[®] Synchronous Serial Interface (SSI) is a master or slave interface for synchronous serial communication with peripheral devices that have either Freescale SPI, MICROWIRE, or Texas Instruments synchronous serial interfaces.

The Stellaris SSI module has the following features:

- Master or slave operation
- Programmable clock bit rate and prescale
- Separate transmit and receive FIFOs, 16 bits wide, 8 locations deep
- Programmable interface operation for Freescale SPI, MICROWIRE, or Texas Instruments synchronous serial interfaces
- Programmable data frame size from 4 to 16 bits
- Internal loopback test mode for diagnostic/debug testing

12.1 Block Diagram

Figure 12-1. SSI Module Block Diagram



12.2 Signal Description

Table 12-1 on page 444 and Table 12-2 on page 444 list the external signals of the SSI module and describe the function of each. The SSI signals are alternate functions for some GPIO signals and

default to be GPIO signals at reset., with the exception of the SSIOClk, SSIOFSS, SSIORX, and SSIOTX pins which default to the SSI function. The column in the table below titled "Pin Assignment" lists the possible GPIO pin placements for the SSI signals. The AFSEL bit in the **GPIO Alternate** Function Select (GPIOAFSEL) register (page 280) should be set to choose the SSI function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOS)" on page 258.

Table 12-1. SSI Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
SSIOClk	28	I/O	TTL	SSI module 0 clock.
SSIOFss	29	I/O	TTL	SSI module 0 frame signal.
SSIORx	30	I	TTL	SSI module 0 receive.
SSIOTx	31	0	TTL	SSI module 0 transmit.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 12-2. SSI Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
SSIOClk	M4	I/O	TTL	SSI module 0 clock.
SSIOFss	L4	I/O	TTL	SSI module 0 frame signal.
SSIORx	L5	1	TTL	SSI module 0 receive.
SSIOTx	M5	0	TTL	SSI module 0 transmit.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

12.3 Functional Description

The SSI performs serial-to-parallel conversion on data received from a peripheral device. The CPU accesses data, control, and status information. The transmit and receive paths are buffered with internal FIFO memories allowing up to eight 16-bit values to be stored independently in both transmit and receive modes.

12.3.1 Bit Rate Generation

The SSI includes a programmable bit rate clock divider and prescaler to generate the serial output clock. Bit rates are supported to 2 MHz and higher, although maximum bit rate is determined by peripheral devices.

The serial bit rate is derived by dividing down the input clock (FSysClk). The clock is first divided by an even prescale value CPSDVSR from 2 to 254, which is programmed in the **SSI Clock Prescale** (**SSICPSR**) register (see page 463). The clock is further divided by a value from 1 to 256, which is 1 + SCR, where SCR is the value programmed in the **SSI Control0** (**SSICR0**) register (see page 456).

The frequency of the output clock SSIClk is defined by:

```
SSIClk = FSysClk / (CPSDVSR * (1 + SCR))
```

Note: For master mode, the system clock must be at least two times faster than the SSIClk. For slave mode, the system clock must be at least 12 times faster than the SSIClk.

See "Synchronous Serial Interface (SSI)" on page 655 to view SSI timing parameters.

12.3.2 FIFO Operation

12.3.2.1 Transmit FIFO

The common transmit FIFO is a 16-bit wide, 8-locations deep, first-in, first-out memory buffer. The CPU writes data to the FIFO by writing the **SSI Data (SSIDR)** register (see page 460), and data is stored in the FIFO until it is read out by the transmission logic.

When configured as a master or a slave, parallel data is written into the transmit FIFO prior to serial conversion and transmission to the attached slave or master, respectively, through the SSITX pin.

In slave mode, the SSI transmits data each time the master initiates a transaction. If the transmit FIFO is empty and the master initiates, the slave transmits the 8th most recent value in the transmit FIFO. If less than 8 values have been written to the transmit FIFO since the SSI module clock was enabled using the SSI bit in the **RGCG1** register, then 0 is transmitted. Care should be taken to ensure that valid data is in the FIFO as needed. The SSI can be configured to generate an interrupt when the FIFO is empty.

12.3.2.2 Receive FIFO

The common receive FIFO is a 16-bit wide, 8-locations deep, first-in, first-out memory buffer. Received data from the serial interface is stored in the buffer until read out by the CPU, which accesses the read FIFO by reading the **SSIDR** register.

When configured as a master or slave, serial data received through the SSIRx pin is registered prior to parallel loading into the attached slave or master receive FIFO, respectively.

12.3.3 Interrupts

The SSI can generate interrupts when the following conditions are observed:

- Transmit FIFO service
- Receive FIFO service
- Receive FIFO time-out
- Receive FIFO overrun

All of the interrupt events are ORed together before being sent to the interrupt controller, so the SSI can only generate a single interrupt request to the controller at any given time. You can mask each of the four individual maskable interrupts by setting the appropriate bits in the **SSI Interrupt Mask** (**SSIIM**) register (see page 464). Setting the appropriate mask bit to 1 enables the interrupt.

Provision of the individual outputs, as well as a combined interrupt output, allows use of either a global interrupt service routine, or modular device drivers to handle interrupts. The transmit and receive dynamic dataflow interrupts have been separated from the status interrupts so that data can be read or written in response to the FIFO trigger levels. The status of the individual interrupt sources can be read from the SSI Raw Interrupt Status (SSIRIS) and SSI Masked Interrupt Status (SSIMIS) registers (see page 466 and page 467, respectively).

12.3.4 Frame Formats

Each data frame is between 4 and 16 bits long, depending on the size of data programmed, and is transmitted starting with the MSB. There are three basic frame types that can be selected:

Texas Instruments synchronous serial

- Freescale SPI
- MICROWIRE

For all three formats, the serial clock (SSIClk) is held inactive while the SSI is idle, and SSIClk transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSIClk is utilized to provide a receive timeout indication that occurs when the receive FIFO still contains data after a timeout period.

For Freescale SPI and MICROWIRE frame formats, the serial frame (SSIFss) pin is active Low, and is asserted (pulled down) during the entire transmission of the frame.

For Texas Instruments synchronous serial frame format, the SSIFss pin is pulsed for one serial clock period starting at its rising edge, prior to the transmission of each frame. For this frame format, both the SSI and the off-chip slave device drive their output data on the rising edge of SSIClk, and latch data from the other device on the falling edge.

Unlike the full-duplex transmission of the other two frame formats, the MICROWIRE format uses a special master-slave messaging technique, which operates at half-duplex. In this mode, when a frame begins, an 8-bit control message is transmitted to the off-chip slave. During this transmit, no incoming data is received by the SSI. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

12.3.4.1 Texas Instruments Synchronous Serial Frame Format

Figure 12-2 on page 446 shows the Texas Instruments synchronous serial frame format for a single transmitted frame.

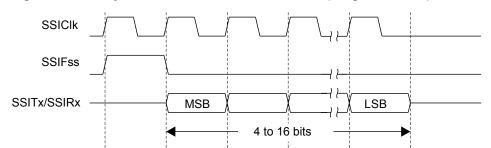


Figure 12-2. TI Synchronous Serial Frame Format (Single Transfer)

In this mode, SSIClk and SSIFss are forced Low, and the transmit data line SSITx is tristated whenever the SSI is idle. Once the bottom entry of the transmit FIFO contains data, SSIFss is pulsed High for one SSIClk period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of SSIClk, the MSB of the 4 to 16-bit data frame is shifted out on the SSITx pin. Likewise, the MSB of the received data is shifted onto the SSIRx pin by the off-chip serial slave device.

Both the SSI and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SSIClk. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SSIClk after the LSB has been latched.

Figure 12-3 on page 447 shows the Texas Instruments synchronous serial frame format when back-to-back frames are transmitted.

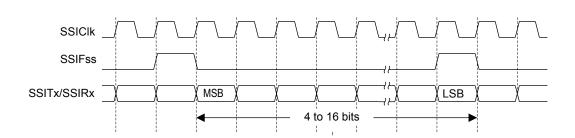


Figure 12-3. TI Synchronous Serial Frame Format (Continuous Transfer)

12.3.4.2 Freescale SPI Frame Format

The Freescale SPI interface is a four-wire interface where the SSIFss signal behaves as a slave select. The main feature of the Freescale SPI format is that the inactive state and phase of the SSIC1k signal are programmable through the SPO and SPH bits within the SSISCR0 control register.

SPO Clock Polarity Bit

When the SPO clock polarity control bit is Low, it produces a steady state Low value on the SSIC1k pin. If the SPO bit is High, a steady state High value is placed on the SSIC1k pin when data is not being transferred.

SPH Phase Control Bit

The SPH phase control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. When the SPH phase control bit is Low, data is captured on the first clock edge transition. If the SPH bit is High, data is captured on the second clock edge transition.

Freescale SPI Frame Format with SPO=0 and SPH=0 12.3.4.3

Single and continuous transmission signal sequences for Freescale SPI format with SPO=0 and SPH=0 are shown in Figure 12-4 on page 447 and Figure 12-5 on page 448.

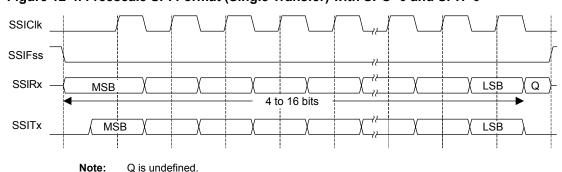


Figure 12-4. Freescale SPI Format (Single Transfer) with SPO=0 and SPH=0

July 15, 2014 447

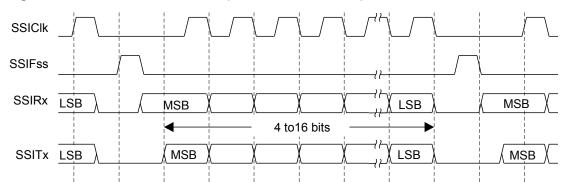


Figure 12-5. Freescale SPI Format (Continuous Transfer) with SPO=0 and SPH=0

In this configuration, during idle periods:

- SSIClk is forced Low
- SSIFss is forced High
- The transmit data line SSITx is arbitrarily forced Low
- When the SSI is configured as a master, it enables the SSIClk pad
- When the SSI is configured as a slave, it disables the SSIClk pad

If the SSI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSIFss master signal being driven Low. This causes slave data to be enabled onto the SSIRx input line of the master. The master SSITx output pad is enabled.

One half \mathtt{SSIClk} period later, valid master data is transferred to the \mathtt{SSITx} pin. Now that both the master and slave data have been set, the \mathtt{SSIClk} master clock pin goes High after one further half \mathtt{SSIClk} period.

The data is now captured on the rising and propagated on the falling edges of the SSIClk signal.

In the case of a single word transmission, after all bits of the data word have been transferred, the SSIFss line is returned to its idle High state one SSIC1k period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSIFss signal must be pulsed High between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore, the master device must raise the SSIFss pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSIFss pin is returned to its idle state one SSIClk period after the last bit has been captured.

12.3.4.4 Freescale SPI Frame Format with SPO=0 and SPH=1

The transfer signal sequence for Freescale SPI format with SPO=0 and SPH=1 is shown in Figure 12-6 on page 449, which covers both single and continuous transfers.

Figure 12-6. Freescale SPI Frame Format with SPO=0 and SPH=1

Note: Q is undefined.

In this configuration, during idle periods:

- SSIC1k is forced Low
- SSIFss is forced High
- The transmit data line SSITx is arbitrarily forced Low
- When the SSI is configured as a master, it enables the SSIClk pad
- When the SSI is configured as a slave, it disables the SSIClk pad

If the SSI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSIFss master signal being driven Low. The master SSITx output is enabled. After a further one half SSIClk period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SSIClk is enabled with a rising edge transition.

Data is then captured on the falling edges and propagated on the rising edges of the SSIC1k signal.

In the case of a single word transfer, after all bits have been transferred, the SSIFss line is returned to its idle High state one SSIClk period after the last bit has been captured.

For continuous back-to-back transfers, the SSIFss pin is held Low between successive data words and termination is the same as that of the single word transfer.

12.3.4.5 Freescale SPI Frame Format with SPO=1 and SPH=0

Q is undefined.

Note:

Single and continuous transmission signal sequences for Freescale SPI format with SPO=1 and SPH=0 are shown in Figure 12-7 on page 449 and Figure 12-8 on page 450.

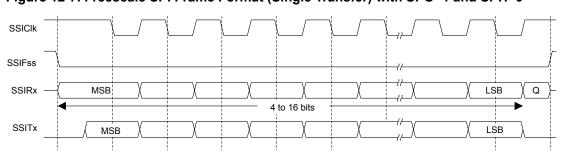


Figure 12-7. Freescale SPI Frame Format (Single Transfer) with SPO=1 and SPH=0

July 15, 2014 449

SSICIK

SSIFss

SSITx/SSIRx

LSB

MSB

4 to 16 bits

Figure 12-8. Freescale SPI Frame Format (Continuous Transfer) with SPO=1 and SPH=0

In this configuration, during idle periods:

- SSIClk is forced High
- SSIFss is forced High
- The transmit data line SSITx is arbitrarily forced Low
- When the SSI is configured as a master, it enables the SSIC1k pad
- When the SSI is configured as a slave, it disables the SSIClk pad

If the SSI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSIFss master signal being driven Low, which causes slave data to be immediately transferred onto the SSIRx line of the master. The master SSITx output pad is enabled.

One half period later, valid master data is transferred to the SSITx line. Now that both the master and slave data have been set, the SSIClk master clock pin becomes Low after one further half SSIClk period. This means that data is captured on the falling edges and propagated on the rising edges of the SSIClk signal.

In the case of a single word transmission, after all bits of the data word are transferred, the SSIFss line is returned to its idle High state one SSIClk period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSIFss signal must be pulsed High between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore, the master device must raise the SSIFss pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSIFss pin is returned to its idle state one SSIClk period after the last bit has been captured.

12.3.4.6 Freescale SPI Frame Format with SPO=1 and SPH=1

The transfer signal sequence for Freescale SPI format with SPO=1 and SPH=1 is shown in Figure 12-9 on page 451, which covers both single and continuous transfers.

Figure 12-9. Freescale SPI Frame Format with SPO=1 and SPH=1

Note: Q is undefined.

In this configuration, during idle periods:

- SSIClk is forced High
- SSIFss is forced High
- The transmit data line SSITx is arbitrarily forced Low
- When the SSI is configured as a master, it enables the SSIClk pad
- When the SSI is configured as a slave, it disables the SSIClk pad

If the SSI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSIFss master signal being driven Low. The master SSITx output pad is enabled. After a further one-half SSIClk period, both master and slave data are enabled onto their respective transmission lines. At the same time, SSIClk is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SSIClk signal.

After all bits have been transferred, in the case of a single word transmission, the SSIFss line is returned to its idle high state one SSIClk period after the last bit has been captured.

For continuous back-to-back transmissions, the SSIFss pin remains in its active Low state, until the final bit of the last word has been captured, and then returns to its idle state as described above.

For continuous back-to-back transfers, the SSIFss pin is held Low between successive data words and termination is the same as that of the single word transfer.

12.3.4.7 MICROWIRE Frame Format

Figure 12-10 on page 451 shows the MICROWIRE frame format, again for a single frame. Figure 12-11 on page 452 shows the same format when back-to-back frames are transmitted.

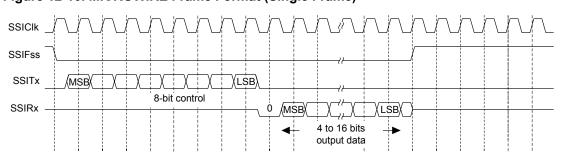


Figure 12-10. MICROWIRE Frame Format (Single Frame)

July 15, 2014 451

MICROWIRE format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SSI to the off-chip slave device. During this transmission, no incoming data is received by the SSI. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the required data. The returned data is 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

In this configuration, during idle periods:

- SSIC1k is forced Low
- SSIFss is forced High
- The transmit data line SSITx is arbitrarily forced Low

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SSIFss causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SSITx pin. SSIFss remains Low for the duration of the frame transmission. The SSIRx pin remains tristated during this transmission.

The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SSIClk. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSI. Each bit is driven onto the SSIRx line on the falling edge of SSIClk. The SSI in turn latches each bit on the rising edge of SSIClk. At the end of the frame, for single transfers, the SSIFss signal is pulled High one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SSIClk after the LSB has been latched by the receive shifter, or when the SSIFss pin goes High.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SSIFss line is continuously asserted (held Low) and transmission of data occurs back-to-back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SSIClk, after the LSB of the frame has been latched into the SSI.

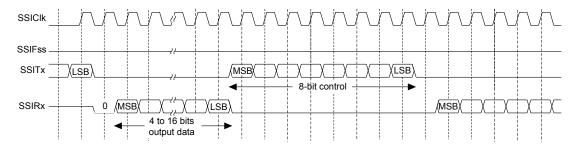
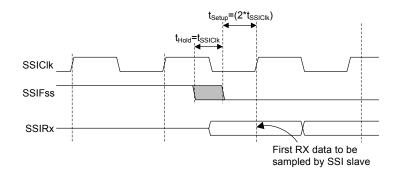


Figure 12-11. MICROWIRE Frame Format (Continuous Transfer)

In the MICROWIRE mode, the SSI slave samples the first bit of receive data on the rising edge of SSIClk after SSIFss has gone Low. Masters that drive a free-running SSIClk must ensure that the SSIFss signal has sufficient setup and hold margins with respect to the rising edge of SSIClk.

Figure 12-12 on page 453 illustrates these setup and hold time requirements. With respect to the SSIClk rising edge on which the first bit of receive data is to be sampled by the SSI slave, SSIFSS must have a setup of at least two times the period of SSIClk on which the SSI operates. With respect to the SSIClk rising edge previous to this edge, SSIFSS must have a hold of at least one SSIClk period.





12.4 Initialization and Configuration

To use the SSI, its peripheral clock must be enabled by setting the SSI bit in the **RCGC1** register. For each of the frame formats, the SSI is configured using the following steps:

- 1. Ensure that the SSE bit in the SSICR1 register is disabled before making any configuration changes.
- 2. Select whether the SSI is a master or slave:
 - **a.** For master operations, set the **SSICR1** register to 0x0000.0000.
 - **b.** For slave mode (output enabled), set the **SSICR1** register to 0x0000.0004.
 - c. For slave mode (output disabled), set the **SSICR1** register to 0x0000.000C.
- 3. Configure the clock prescale divisor by writing the **SSICPSR** register.
- **4.** Write the **SSICR0** register with the following configuration:
 - Serial clock rate (SCR)
 - Desired clock phase/polarity, if using Freescale SPI mode (SPH and SPO)
 - The protocol mode: Freescale SPI, TI SSF, MICROWIRE (FRF)
 - The data size (DSS)
- 5. Enable the SSI by setting the SSE bit in the SSICR1 register.

As an example, assume the SSI must be configured to operate with the following parameters:

Master operation

- Freescale SPI mode (SPO=1, SPH=1)
- 1 Mbps bit rate
- 8 data bits

Assuming the system clock is 20 MHz, the bit rate calculation would be:

```
FSSIClk = FSysClk / (CPSDVSR * (1 + SCR))
1x10<sup>6</sup> = 20x10<sup>6</sup> / (CPSDVSR * (1 + SCR))
```

In this case, if CPSDVSR=2, SCR must be 9.

The configuration sequence would be as follows:

- 1. Ensure that the SSE bit in the SSICR1 register is disabled.
- 2. Write the **SSICR1** register with a value of 0x0000.0000.
- 3. Write the **SSICPSR** register with a value of 0x0000.0002.
- **4.** Write the **SSICR0** register with a value of 0x0000.09C7.
- 5. The SSI is then enabled by setting the SSE bit in the SSICR1 register to 1.

12.5 Register Map

Table 12-3 on page 454 lists the SSI registers. The offset listed is a hexadecimal increment to the register's address, relative to that SSI module's base address:

■ SSI0: 0x4000.8000

Note that the SSI module clock must be enabled before the registers can be programmed (see page 215). There must be a delay of 3 system clocks after the SSI module clock is enabled before any SSI module registers are accessed.

Note: The SSI must be disabled (see the SSE bit in the **SSICR1** register) before any of the control registers are reprogrammed.

Table 12-3. SSI Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	SSICR0	R/W	0x0000.0000	SSI Control 0	456
0x004	SSICR1	R/W	0x0000.0000	SSI Control 1	458
0x008	SSIDR	R/W	0x0000.0000	SSI Data	460
0x00C	SSISR	RO	0x0000.0003	SSI Status	461
0x010	SSICPSR	R/W	0x0000.0000	SSI Clock Prescale	463
0x014	SSIIM	R/W	0x0000.0000	SSI Interrupt Mask	464
0x018	SSIRIS	RO	0x0000.0008	SSI Raw Interrupt Status	466
0x01C	SSIMIS	RO	0x0000.0000	SSI Masked Interrupt Status	467

Table 12-3. SSI Register Map (continued)

Offset	Name	Type	Reset	Description	See page
0x020	SSIICR	W1C	0x0000.0000	SSI Interrupt Clear	468
0xFD0	SSIPeriphID4	RO	0x0000.0000	SSI Peripheral Identification 4	469
0xFD4	SSIPeriphID5	RO	0x0000.0000	SSI Peripheral Identification 5	470
0xFD8	SSIPeriphID6	RO	0x0000.0000	SSI Peripheral Identification 6	471
0xFDC	SSIPeriphID7	RO	0x0000.0000	SSI Peripheral Identification 7	472
0xFE0	SSIPeriphID0	RO	0x0000.0022	SSI Peripheral Identification 0	473
0xFE4	SSIPeriphID1	RO	0x0000.0000	SSI Peripheral Identification 1	474
0xFE8	SSIPeriphID2	RO	0x0000.0018	SSI Peripheral Identification 2	475
0xFEC	SSIPeriphID3	RO	0x0000.0001	SSI Peripheral Identification 3	476
0xFF0	SSIPCellID0	RO	0x0000.000D	SSI PrimeCell Identification 0	477
0xFF4	SSIPCellID1	RO	0x0000.00F0	SSI PrimeCell Identification 1	478
0xFF8	SSIPCellID2	RO	0x0000.0005	SSI PrimeCell Identification 2	479
0xFFC	SSIPCellID3	RO	0x0000.00B1	SSI PrimeCell Identification 3	480

12.6 Register Descriptions

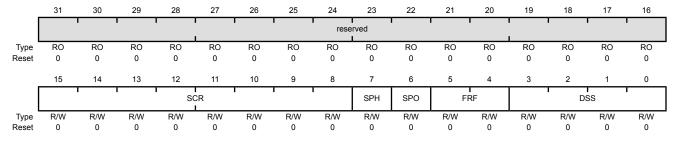
The remainder of this section lists and describes the SSI registers, in numerical order by address offset.

Register 1: SSI Control 0 (SSICR0), offset 0x000

SSICR0 is control register 0 and contains bit fields that control various functions within the SSI module. Functionality such as protocol mode, clock rate, and data size are configured in this register.

SSI Control 0 (SSICR0)

SSI0 base: 0x4000.8000 Offset 0x000 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:8	SCR	R/W	0x0000	SSI Serial Clock Rate
				The value ${\tt SCR}$ is used to generate the transmit and receive bit rate of the SSI. The bit rate is:
				BR=FSSIClk/(CPSDVSR * (1 + SCR))
				where CPSDVSR is an even value from 2-254 programmed in the SSICPSR register, and SCR is a value from 0-255.
7	SPH	R/W	0	SSI Serial Clock Phase
				This bit is only applicable to the Freescale SPI Format.
				The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge.
				When the SPH bit is 0, data is captured on the first clock edge transition. If SPH is 1, data is captured on the second clock edge transition.
6	SPO	R/W	0	SSI Serial Clock Polarity
				This bit is only applicable to the Freescale SPI Format.
				When the SPO bit is 0, it produces a steady state Low value on the SSIClk pin. If SPO is 1, a steady state High value is placed on the SSIClk pin when data is not being transferred.
5:4	FRF	R/W	0x0	SSI Frame Format Select
				The FRF values are defined as follows:
				Value Frame Format

Value Frame Format

0x0 Freescale SPI Frame Format

Texas Instruments Synchronous Serial Frame Format

MICROWIRE Frame Format 0x2

Reserved 0x3

Bit/Field	Name	Туре	Reset	Description
3:0	DSS	R/W	0x00	SSI Data Size Select The DSS values are defined as follows:
				Value Data Size
				0x0-0x2 Reserved
				0x3 4-bit data
				0x4 5-bit data
				0x5 6-bit data
				0x6 7-bit data
				0x7 8-bit data
				0x8 9-bit data
				0x9 10-bit data
				0xA 11-bit data
				0xB 12-bit data
				0xC 13-bit data
				0xD 14-bit data
				0xE 15-bit data
				0xF 16-bit data

Register 2: SSI Control 1 (SSICR1), offset 0x004

SSICR1 is control register 1 and contains bit fields that control various functions within the SSI module. Master and slave mode functionality is controlled by this register.

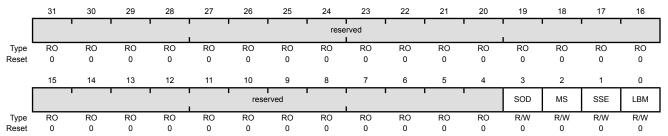
SSI Control 1 (SSICR1)

SSI0 base: 0x4000.8000

Dit/Eiold

Nomo

Offset 0x004 Type R/W, reset 0x0000.0000



Divrieiu	Name	туре	Reset	Description
31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	SOD	R/W	0	SSI Slave Mode Output Disable

Description

Dooot

This bit is relevant only in the Slave mode (MS=1). In multiple-slave systems, it is possible for the SSI master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto the serial output line. In such systems, the TXD lines from multiple slaves could be tied together. To operate in such a system, the SOD bit can be

configured so that the SSI slave does not drive the SSITx pin.

The SOD values are defined as follows:

Value Description

- SSI can drive SSITx output in Slave Output mode.
- SSI must not drive the SSITx output in Slave mode.

2 R/W MS 0 SSI Master/Slave Select

This bit selects Master or Slave mode and can be modified only when SSI is disabled (SSE=0).

The MS values are defined as follows:

Value Description

- Device configured as a master.
- Device configured as a slave.

Bit/Field	Name	Туре	Reset	Description
1	SSE	R/W	0	SSI Synchronous Serial Port Enable Setting this bit enables SSI operation. The SSE values are defined as follows:
				Value Description 0 SSI operation disabled. 1 SSI operation enabled.
				Note: This bit must be set to 0 before any control registers are reprogrammed.
0	LBM	R/W	0	SSI Loopback Mode Setting this bit enables Loopback Test mode. The LBM values are defined as follows: Value Description

- 0 Normal serial port operation enabled.
- Output of the transmit serial shift register is connected internally to the input of the receive serial shift register.

Register 3: SSI Data (SSIDR), offset 0x008

Important: This register is read-sensitive. See the register description for details.

SSIDR is the data register and is 16-bits wide. When **SSIDR** is read, the entry in the receive FIFO (pointed to by the current FIFO read pointer) is accessed. As data values are removed by the SSI receive logic from the incoming data frame, they are placed into the entry in the receive FIFO (pointed to by the current FIFO write pointer).

When **SSIDR** is written to, the entry in the transmit FIFO (pointed to by the write pointer) is written to. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the SSITX pin at the programmed bit rate.

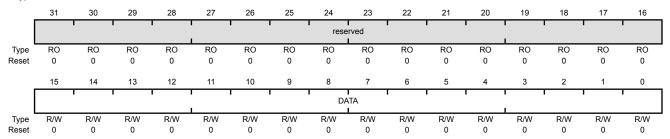
When a data size of less than 16 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits. Received data less than 16 bits is automatically right-justified in the receive buffer.

When the SSI is programmed for MICROWIRE frame format, the default size for transmit data is eight bits (the most significant byte is ignored). The receive data size is controlled by the programmer. The transmit FIFO and the receive FIFO are not cleared even when the SSE bit in the **SSICR1** register is set to zero. This allows the software to fill the transmit FIFO before enabling the SSI.

SSI Data (SSIDR)

SSI0 base: 0x4000.8000 Offset 0x008

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	DATA	R/W	0x0000	SSI Receive/Transmit Data

A read operation reads the receive FIFO. A write operation writes the transmit FIFO.

Software must right-justify data when the SSI is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by the transmit logic. The receive logic automatically right-justifies the data.

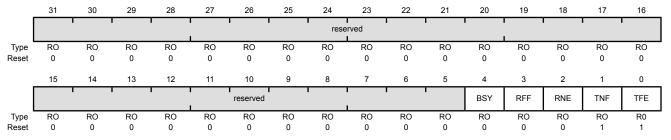
Register 4: SSI Status (SSISR), offset 0x00C

SSISR is a status register that contains bits that indicate the FIFO fill status and the SSI busy status.

SSI Status (SSISR)

SSI0 base: 0x4000.8000 Offset 0x00C

Type RO, reset 0x0000.0003



Bit/Field	Name	Туре	Reset	Description
31:5	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	BSY	RO	0	SSI Busy Bit
				The BSY values are defined as follows:
				Value Description
				0 SSI is idle.
				1 SSI is currently transmitting and/or receiving a frame, or the transmit FIFO is not empty.
3	RFF	RO	0	SSI Receive FIFO Full
				The RFF values are defined as follows:
				Value Description
				0 Receive FIFO is not full.
				1 Receive FIFO is full.
2	RNE	RO	0	SSI Receive FIFO Not Empty
				The RNE values are defined as follows:
				Value Description
				0 Receive FIFO is empty.
				1 Receive FIFO is not empty.
1	TNF	RO	1	SSI Transmit FIFO Not Full
			-	The TNF values are defined as follows:
				Value Description

July 15, 2014 461

Transmit FIFO is full. Transmit FIFO is not full.

Bit/Field	Name	Туре	Reset	Description
0	TFE	R0	1	SSI Transmit FIFO Empty The TFE values are defined as follows:
				Value Description
				0 Transmit FIFO is not empty.
				1 Transmit FIFO is empty.

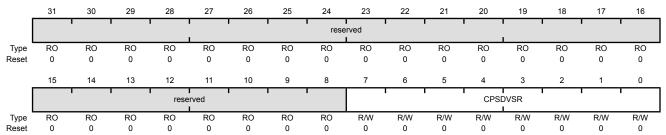
Register 5: SSI Clock Prescale (SSICPSR), offset 0x010

SSICPSR is the clock prescale register and specifies the division factor by which the system clock must be internally divided before further use.

The value programmed into this register must be an even number between 2 and 254. The least-significant bit of the programmed number is hard-coded to zero. If an odd number is written to this register, data read back from this register has the least-significant bit as zero.

SSI Clock Prescale (SSICPSR)

SSI0 base: 0x4000.8000 Offset 0x010 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CPSDVSR	R/W	0x00	SSI Clock Prescale Divisor

This value must be an even number from 2 to 254, depending on the frequency of SSIC1k. The LSB always returns 0 on reads.

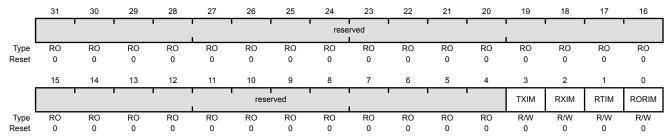
Register 6: SSI Interrupt Mask (SSIIM), offset 0x014

The SSIIM register is the interrupt mask set or clear register. It is a read/write register and all bits are cleared to 0 on reset.

On a read, this register gives the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

SSI Interrupt Mask (SSIIM)

SSI0 base: 0x4000.8000 Offset 0x014 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TXIM	R/W	0	SSI Transmit FIFO Interrupt Mask The TXIM values are defined as follows: Value Description 0 TX FIFO half-empty or less condition interrupt is masked. 1 TX FIFO half-empty or less condition interrupt is not masked.
2	RXIM	R/W	0	SSI Receive FIFO Interrupt Mask The RXIM values are defined as follows: Value Description 0 RX FIFO half-full or more condition interrupt is masked. 1 RX FIFO half-full or more condition interrupt is not masked.
1	RTIM	R/W	0	SSI Receive Time-Out Interrupt Mask The RTIM values are defined as follows:

Value Description

- RX FIFO time-out interrupt is masked.
- RX FIFO time-out interrupt is not masked.

Bit/Field	Name	Type	Reset	Description
0	RORIM	R/W	0	SSI Receive Overrun Interrupt Mask The RORIM values are defined as follows:
				Value Description

Value Description

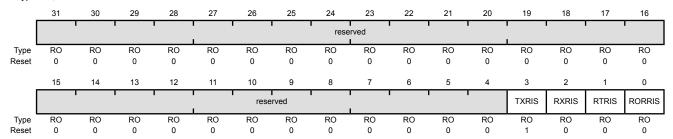
- 0 RX FIFO overrun interrupt is masked.
- 1 RX FIFO overrun interrupt is not masked.

Register 7: SSI Raw Interrupt Status (SSIRIS), offset 0x018

The SSIRIS register is the raw interrupt status register. On a read, this register gives the current raw status value of the corresponding interrupt prior to masking. A write has no effect.

SSI Raw Interrupt Status (SSIRIS)

SSI0 base: 0x4000.8000 Offset 0x018 Type RO, reset 0x0000.0008



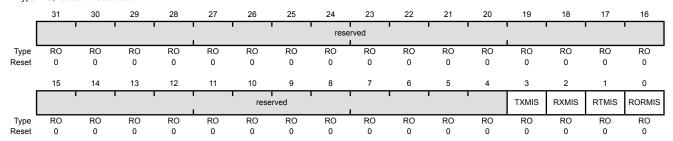
Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TXRIS	RO	1	SSI Transmit FIFO Raw Interrupt Status Indicates that the transmit FIFO is half empty or less, when set.
2	RXRIS	RO	0	SSI Receive FIFO Raw Interrupt Status Indicates that the receive FIFO is half full or more, when set.
1	RTRIS	RO	0	SSI Receive Time-Out Raw Interrupt Status Indicates that the receive time-out has occurred, when set.
0	RORRIS	RO	0	SSI Receive Overrun Raw Interrupt Status Indicates that the receive FIFO has overflowed, when set.

Register 8: SSI Masked Interrupt Status (SSIMIS), offset 0x01C

The **SSIMIS** register is the masked interrupt status register. On a read, this register gives the current masked status value of the corresponding interrupt. A write has no effect.

SSI Masked Interrupt Status (SSIMIS)

SSI0 base: 0x4000.8000 Offset 0x01C Type RO, reset 0x0000.0000



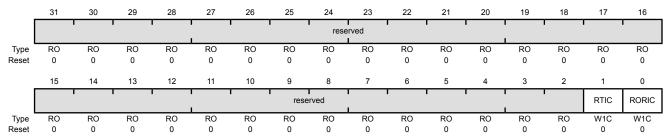
Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TXMIS	RO	0	SSI Transmit FIFO Masked Interrupt Status Indicates that the transmit FIFO is half empty or less, when set.
2	RXMIS	RO	0	SSI Receive FIFO Masked Interrupt Status Indicates that the receive FIFO is half full or more, when set.
1	RTMIS	RO	0	SSI Receive Time-Out Masked Interrupt Status Indicates that the receive time-out has occurred, when set.
0	RORMIS	RO	0	SSI Receive Overrun Masked Interrupt Status Indicates that the receive FIFO has overflowed, when set.

Register 9: SSI Interrupt Clear (SSIICR), offset 0x020

The SSIICR register is the interrupt clear register. On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.

SSI Interrupt Clear (SSIICR)

SSI0 base: 0x4000.8000 Offset 0x020 Type W1C, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	RTIC	W1C	0	SSI Receive Time-Out Interrupt Clear The RTIC values are defined as follows:
				Value Description 0 No effect on interrupt. 1 Clears interrupt.
0	RORIC	W1C	0	SSI Receive Overrun Interrupt Clear The RORIC values are defined as follows:

Value Description

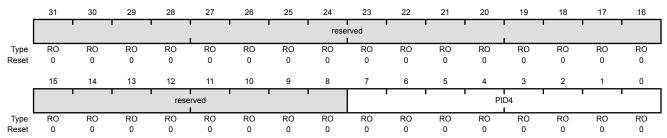
- No effect on interrupt.
- Clears interrupt.

Register 10: SSI Peripheral Identification 4 (SSIPeriphID4), offset 0xFD0

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 4 (SSIPeriphID4)

SSI0 base: 0x4000.8000 Offset 0xFD0 Type RO, reset 0x0000.0000



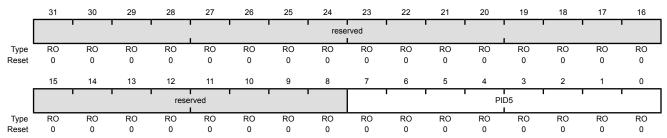
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x00	SSI Peripheral ID Register[7:0]

Register 11: SSI Peripheral Identification 5 (SSIPeriphID5), offset 0xFD4

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 5 (SSIPeriphID5)

SSI0 base: 0x4000.8000 Offset 0xFD4 Type RO, reset 0x0000.0000



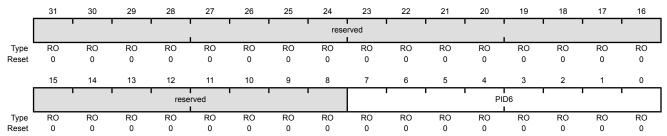
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID5	RO	0x00	SSI Peripheral ID Register[15:8]

Register 12: SSI Peripheral Identification 6 (SSIPeriphID6), offset 0xFD8

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 6 (SSIPeriphID6)

SSI0 base: 0x4000.8000 Offset 0xFD8 Type RO, reset 0x0000.0000



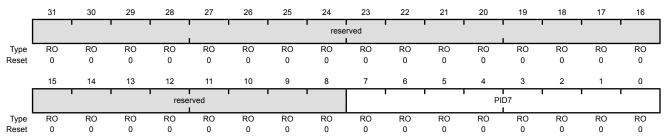
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID6	RO	0x00	SSI Peripheral ID Register[23:16]

Register 13: SSI Peripheral Identification 7 (SSIPeriphID7), offset 0xFDC

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 7 (SSIPeriphID7)

SSI0 base: 0x4000.8000 Offset 0xFDC Type RO, reset 0x0000.0000



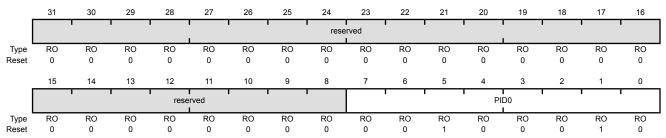
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID7	RO	0x00	SSI Peripheral ID Register[31:24]

Register 14: SSI Peripheral Identification 0 (SSIPeriphID0), offset 0xFE0

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 0 (SSIPeriphID0)

SSI0 base: 0x4000.8000 Offset 0xFE0 Type RO, reset 0x0000.0022



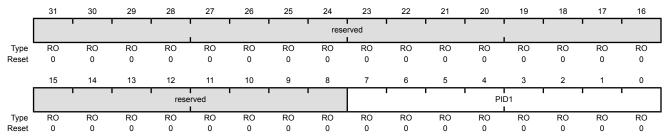
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID0	RO	0x22	SSI Peripheral ID Register[7:0]

Register 15: SSI Peripheral Identification 1 (SSIPeriphID1), offset 0xFE4

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 1 (SSIPeriphID1)

SSI0 base: 0x4000.8000 Offset 0xFE4 Type RO, reset 0x0000.0000



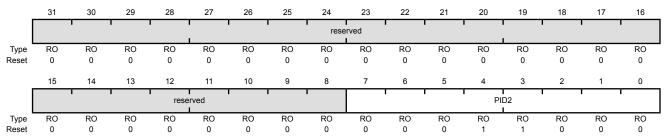
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID1	RO	0x00	SSI Peripheral ID Register [15:8]

Register 16: SSI Peripheral Identification 2 (SSIPeriphID2), offset 0xFE8

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 2 (SSIPeriphID2)

SSI0 base: 0x4000.8000 Offset 0xFE8 Type RO, reset 0x0000.0018



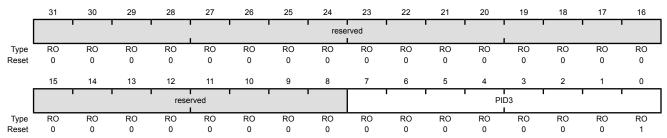
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID2	RO	0x18	SSI Peripheral ID Register [23:16]

Register 17: SSI Peripheral Identification 3 (SSIPeriphID3), offset 0xFEC

The SSIPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

SSI Peripheral Identification 3 (SSIPeriphID3)

SSI0 base: 0x4000.8000 Offset 0xFEC Type RO, reset 0x0000.0001



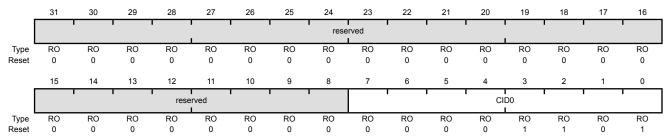
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	SSI Peripheral ID Register [31:24]

Register 18: SSI PrimeCell Identification 0 (SSIPCellID0), offset 0xFF0

The SSIPCellIDn registers are hard-coded, and the fields within the register determine the reset value.

SSI PrimeCell Identification 0 (SSIPCellID0)

SSI0 base: 0x4000.8000 Offset 0xFF0 Type RO, reset 0x0000.000D



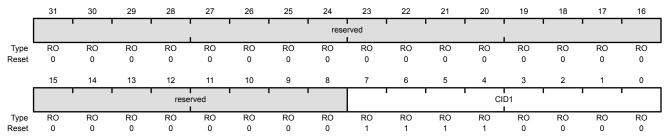
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	SSI PrimeCell ID Register [7:0]

Register 19: SSI PrimeCell Identification 1 (SSIPCellID1), offset 0xFF4

The SSIPCeIIIDn registers are hard-coded, and the fields within the register determine the reset value.

SSI PrimeCell Identification 1 (SSIPCellID1)

SSI0 base: 0x4000.8000 Offset 0xFF4 Type RO, reset 0x0000.00F0



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	SSI PrimeCell ID Register [15:8]

Register 20: SSI PrimeCell Identification 2 (SSIPCellID2), offset 0xFF8

The SSIPCellIDn registers are hard-coded, and the fields within the register determine the reset value.

SSI PrimeCell Identification 2 (SSIPCelIID2)

SSI0 base: 0x4000.8000 Offset 0xFF8 Type RO, reset 0x0000.0005



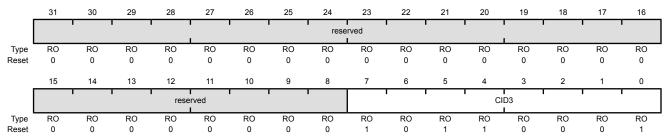
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	SSI PrimeCell ID Register [23:16]

Register 21: SSI PrimeCell Identification 3 (SSIPCelIID3), offset 0xFFC

The SSIPCeIIIDn registers are hard-coded, and the fields within the register determine the reset value.

SSI PrimeCell Identification 3 (SSIPCelIID3)

SSI0 base: 0x4000.8000 Offset 0xFFC Type RO, reset 0x0000.00B1



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID3	RO	0xB1	SSI PrimeCell ID Register [31:24]

13 Inter-Integrated Circuit (I²C) Interface

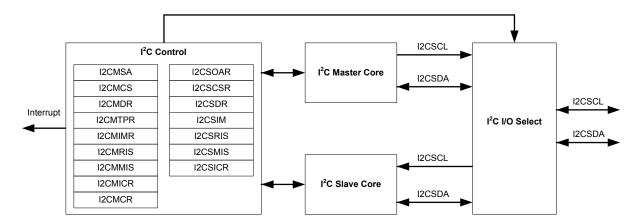
The Inter-Integrated Circuit (I^2C) bus provides bi-directional data transfer through a two-wire design (a serial data line SDA and a serial clock line SCL), and interfaces to external I^2C devices such as serial memory (RAMs and ROMs), networking devices, LCDs, tone generators, and so on. The I^2C bus may also be used for system testing and diagnostic purposes in product development and manufacture. The LM3S6432 microcontroller includes one I^2C module, providing the ability to interact (both send and receive) with other I^2C devices on the bus.

The Stellaris® I2C interface has the following features:

- Devices on the I²C bus can be designated as either a master or a slave
 - Supports both sending and receiving data as either a master or a slave
 - Supports simultaneous master and slave operation
- Four I²C modes
 - Master transmit
 - Master receive
 - Slave transmit
 - Slave receive
- Two transmission speeds: Standard (100 Kbps) and Fast (400 Kbps)
- Master and slave interrupt generation
 - Master generates interrupts when a transmit or receive operation completes (or aborts due to an error)
 - Slave generates interrupts when data has been sent or requested by a master
- Master with arbitration and clock synchronization, multimaster support, and 7-bit addressing mode

13.1 Block Diagram

Figure 13-1. I²C Block Diagram



13.2 Signal Description

Table 13-1 on page 482 and Table 13-2 on page 482 list the external signals of the I^2C interface and describe the function of each. The I^2C interface signals are alternate functions for some GPIO signals and default to be GPIO signals at reset., with the exception of the I2C0SCL and I2CSDA pins which default to the I^2C function. The column in the table below titled "Pin Assignment" lists the possible GPIO pin placements for the I^2C signals. The AFSEL bit in the **GPIO Alternate Function Select** (**GPIOAFSEL**) register (page 280) should be set to choose the I^2C function. Note that the I^2C pins should be set to open drain using the **GPIO Open Drain Select** (**GPIOODR**) register. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 13-1. I2C Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
I2C0SCL	70	I/O	OD	I ² C module 0 clock.
I2C0SDA	71	I/O	OD	I ² C module 0 data.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 13-2. I2C Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
I2C0SCL	C11	I/O	OD	I ² C module 0 clock.
I2C0SDA	C12	I/O	OD	I ² C module 0 data.

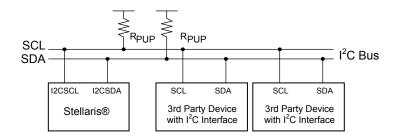
a. The TTL designation indicates the pin has TTL-compatible voltage levels.

13.3 Functional Description

The I²C module is comprised of both master and slave functions which are implemented as separate peripherals. For proper operation, the SDA and SCL pins must be connected to bi-directional open-drain pads. A typical I²C bus configuration is shown in Figure 13-2 on page 483.

See "Inter-Integrated Circuit (I²C) Interface" on page 656 for I²C timing diagrams.

Figure 13-2. I²C Bus Configuration



13.3.1 I²C Bus Functional Overview

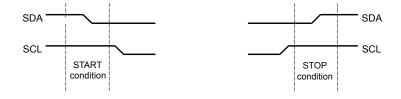
The I²C bus uses only two signals: SDA and SCL, named I2CSDA and I2CSCL on Stellaris microcontrollers. SDA is the bi-directional serial data line and SCL is the bi-directional serial clock line. The bus is considered idle when both lines are High.

Every transaction on the I²C bus is nine bits long, consisting of eight data bits and a single acknowledge bit. The number of bytes per transfer (defined as the time between a valid START and STOP condition, described in "START and STOP Conditions" on page 483) is unrestricted, but each byte has to be followed by an acknowledge bit, and data must be transferred MSB first. When a receiver cannot receive another complete byte, it can hold the clock line SCL Low and force the transmitter into a wait state. The data transfer continues when the receiver releases the clock SCL.

13.3.1.1 START and STOP Conditions

The protocol of the I²C bus defines two states to begin and end a transaction: START and STOP. A High-to-Low transition on the SDA line while the SCL is High is defined as a START condition, and a Low-to-High transition on the SDA line while SCL is High is defined as a STOP condition. The bus is considered busy after a START condition and free after a STOP condition. See Figure 13-3 on page 483.

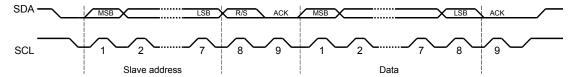
Figure 13-3. START and STOP Conditions



13.3.1.2 Data Format with 7-Bit Address

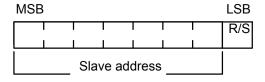
Data transfers follow the format shown in Figure 13-4 on page 484. After the START condition, a slave address is sent. This address is 7-bits long followed by an eighth bit, which is a data direction bit (\mathbb{R}/\mathbb{S} bit in the **I2CMSA** register). A zero indicates a transmit operation (send), and a one indicates a request for data (receive). A data transfer is always terminated by a STOP condition generated by the master, however, a master can initiate communications with another device on the bus by generating a repeated START condition and addressing another slave without first generating a STOP condition. Various combinations of receive/send formats are then possible within a single transfer.

Figure 13-4. Complete Data Transfer with a 7-Bit Address



The first seven bits of the first byte make up the slave address (see Figure 13-5 on page 484). The eighth bit determines the direction of the message. A zero in the R/S position of the first byte means that the master will write (send) data to the selected slave, and a one in this position means that the master will receive data from the slave.

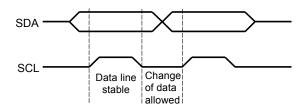
Figure 13-5. R/S Bit in First Byte



13.3.1.3 Data Validity

The data on the SDA line must be stable during the high period of the clock, and the data line can only change when SCL is Low (see Figure 13-6 on page 484).

Figure 13-6. Data Validity During Bit Transfer on the I²C Bus



13.3.1.4 Acknowledge

All bus transactions have a required acknowledge clock cycle that is generated by the master. During the acknowledge cycle, the transmitter (which can be the master or slave) releases the SDA line. To acknowledge the transaction, the receiver must pull down SDA during the acknowledge clock cycle. The data sent out by the receiver during the acknowledge cycle must comply with the data validity requirements described in "Data Validity" on page 484.

When a slave receiver does not acknowledge the slave address, SDA must be left High by the slave so that the master can generate a STOP condition and abort the current transfer. If the master device is acting as a receiver during a transfer, it is responsible for acknowledging each transfer made by the slave. Since the master controls the number of bytes in the transfer, it signals the end of data to the slave transmitter by not generating an acknowledge on the last data byte. The slave transmitter must then release SDA to allow the master to generate the STOP or a repeated START condition.

13.3.1.5 Arbitration

A master may start a transfer only if the bus is idle. It's possible for two or more masters to generate a START condition within minimum hold time of the START condition. In these situations, an

arbitration scheme takes place on the SDA line, while SCL is High. During arbitration, the first of the competing master devices to place a '1' (High) on SDA while another master transmits a '0' (Low) will switch off its data output stage and retire until the bus is idle again.

Arbitration can take place over several bits. Its first stage is a comparison of address bits, and if both masters are trying to address the same device, arbitration continues on to the comparison of data bits.

13.3.2 Available Speed Modes

The I²C clock rate is determined by the parameters: CLK_PRD, TIMER_PRD, SCL_LP, and SCL_HP.

where:

CLK_PRD is the system clock period

SCL_LP is the low phase of SCL (fixed at 6)

SCL_HP is the high phase of SCL (fixed at 4)

TIMER_PRD is the programmed value in the I²C Master Timer Period (I2CMTPR) register (see page 503).

The I²C clock period is calculated as follows:

```
SCL PERIOD = 2*(1 + TIMER PRD)*(SCL LP + SCL HP)*CLK PRD
```

For example:

```
CLK_PRD = 50 ns
TIMER_PRD = 2
SCL_LP=6
SCL_HP=4
```

yields a SCL frequency of:

```
1/T = 333 \text{ Khz}
```

Table 13-3 on page 485 gives examples of timer period, system clock, and speed mode (Standard or Fast).

Table 13-3. Examples of I²C Master Timer Period versus Speed Mode

System Clock	Timer Period	Standard Mode	Timer Period	Fast Mode
4 MHz	0x01	100 Kbps	-	-
6 MHz	0x02	100 Kbps	-	-
12.5 MHz	0x06	89 Kbps	0x01	312 Kbps
16.7 MHz	0x08	93 Kbps	0x02	278 Kbps
20 MHz	0x09	100 Kbps	0x02	333 Kbps
25 MHz	0x0C	96.2 Kbps	0x03	312 Kbps
33 MHz	0x10	97.1 Kbps	0x04	330 Kbps
40 MHz	0x13	100 Kbps	0x04	400 Kbps
50 MHz	0x18	100 Kbps	0x06	357 Kbps

13.3.3 Interrupts

The I²C can generate interrupts when the following conditions are observed:

- Master transaction completed
- Master arbitration lost
- Master transaction error
- Slave transaction received
- Slave transaction requested

There is a separate interrupt signal for the I²C master and I²C slave modules. While both modules can generate interrupts for multiple conditions, only a single interrupt signal is sent to the interrupt controller.

13.3.3.1 I²C Master Interrupts

The I^2C master module generates an interrupt when a transaction completes (either transmit or receive), when arbitration is lost, or when an error occurs during a transaction. To enable the I^2C master interrupt, software must set the IM bit in the I^2C Master Interrupt Mask (I2CMIMR) register. When an interrupt condition is met, software must check the ERROR and ARBLST bits in the I^2C Master Control/Status (I2CMCS) register to verify that an error didn't occur during the last transaction and to ensure that arbitration has not been lost. An error condition is asserted if the last transaction wasn't acknowledged by the slave. If an error is not detected and the master has not lost arbitration, the application can proceed with the transfer. The interrupt is cleared by writing a 1 to the IC bit in the I^2C Master Interrupt Clear (I2CMICR) register.

If the application doesn't require the use of interrupts, the raw interrupt status is always visible via the I^2C Master Raw Interrupt Status (I2CMRIS) register.

13.3.3.2 I²C Slave Interrupts

The slave module can generate an interrupt when data has been received or requested. This interrupt is enabled by writing a 1 to the DATAIM bit in the I^2C Slave Interrupt Mask (I2CSIMR) register. Software determines whether the module should write (transmit) or read (receive) data from the I^2C Slave Data (I2CSDR) register, by checking the RREQ and TREQ bits of the I^2C Slave Control/Status (I2CSCSR) register. If the slave module is in receive mode and the first byte of a transfer is received, the FBR bit is set along with the RREQ bit. The interrupt is cleared by writing a 1 to the DATAIC bit in the I^2C Slave Interrupt Clear (I2CSICR) register.

If the application doesn't require the use of interrupts, the raw interrupt status is always visible via the I²C Slave Raw Interrupt Status (I2CSRIS) register.

13.3.4 Loopback Operation

The I²C modules can be placed into an internal loopback mode for diagnostic or debug work. This is accomplished by setting the LPBK bit in the I²C Master Configuration (I2CMCR) register. In loopback mode, the SDA and SCL signals from the master and slave modules are tied together.

13.3.5 Command Sequence Flow Charts

This section details the steps required to perform the various I²C transfer types in both master and slave mode.

13.3.5.1 I²C Master Command Sequences

The figures that follow show the command sequences available for the I^2C master.

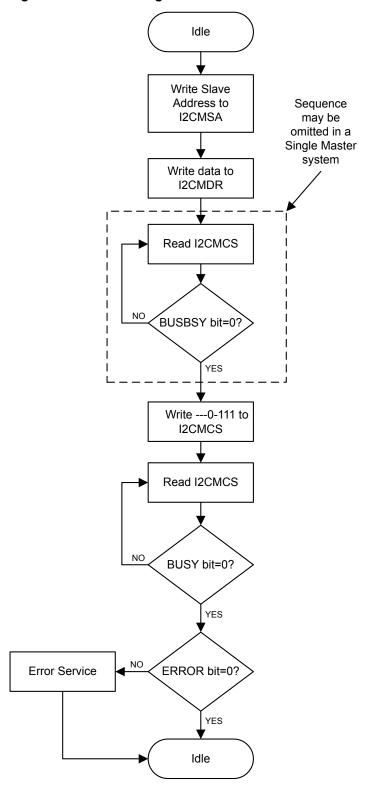


Figure 13-7. Master Single SEND

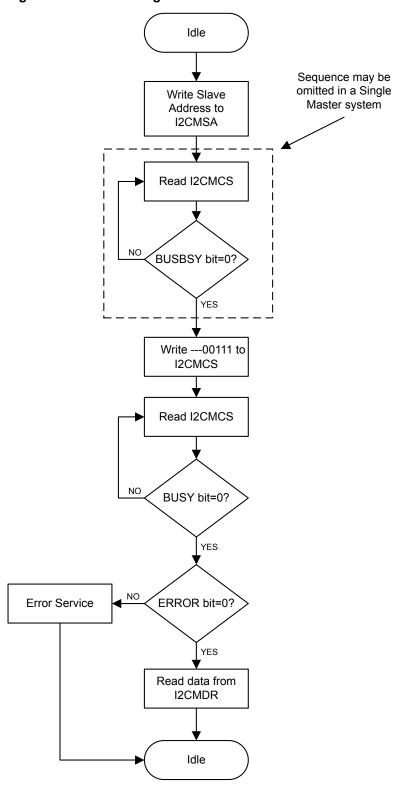


Figure 13-8. Master Single RECEIVE

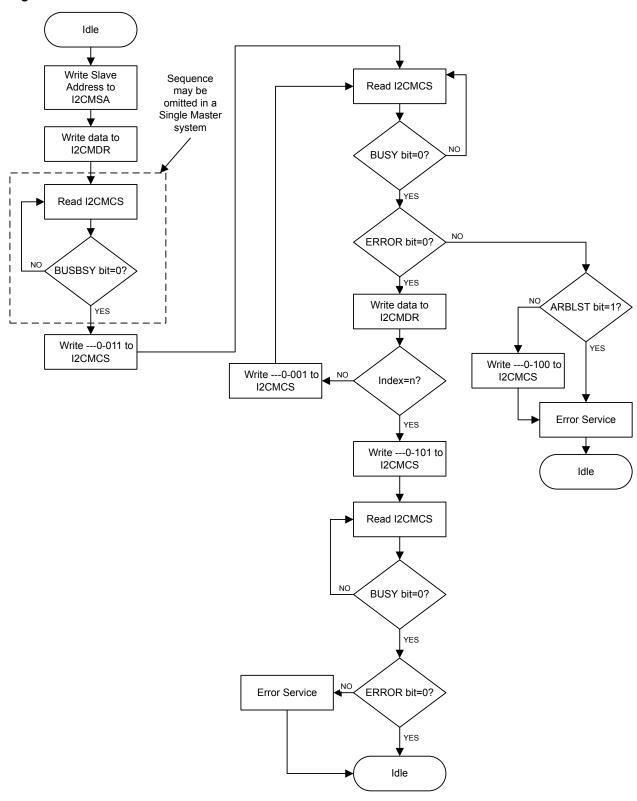


Figure 13-9. Master Burst SEND

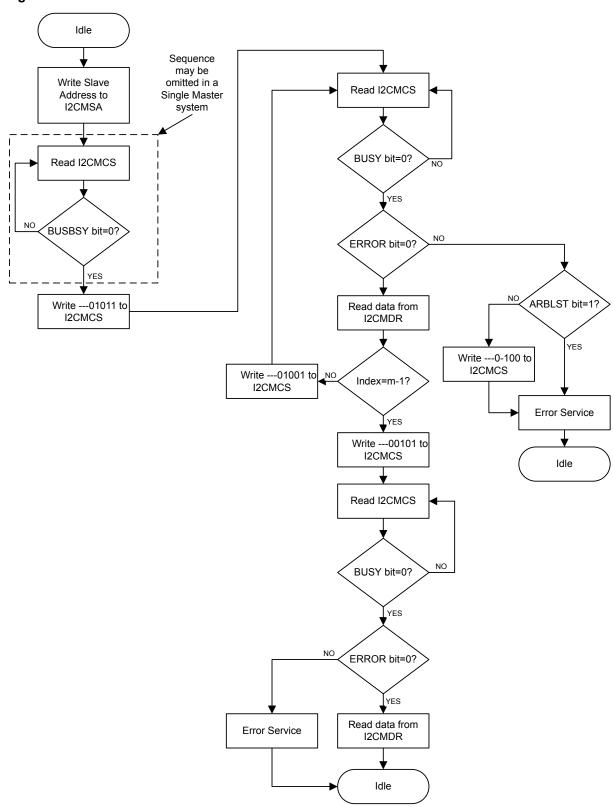


Figure 13-10. Master Burst RECEIVE

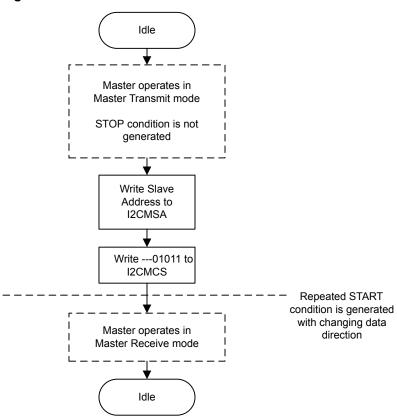


Figure 13-11. Master Burst RECEIVE after Burst SEND

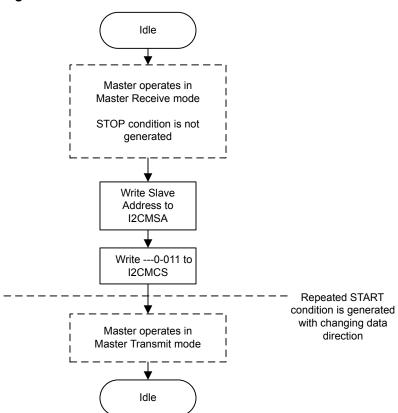


Figure 13-12. Master Burst SEND after Burst RECEIVE

13.3.5.2 I²C Slave Command Sequences

Figure 13-13 on page 494 presents the command sequence available for the I²C slave.

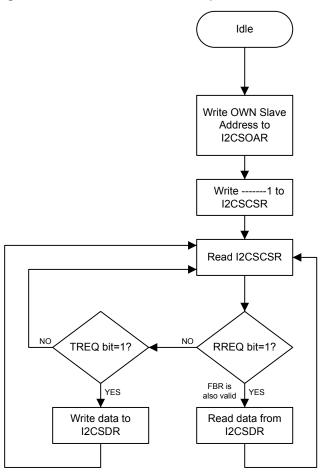


Figure 13-13. Slave Command Sequence

13.4 Initialization and Configuration

The following example shows how to configure the I²C module to send a single byte as a master. This assumes the system clock is 20 MHz.

- **1.** Enable the I²C clock by writing a value of 0x0000.1000 to the **RCGC1** register in the System Control module.
- Enable the clock to the appropriate GPIO module via the RCGC2 register in the System Control module.
- **3.** In the GPIO module, enable the appropriate pins for their alternate function using the **GPIOAFSEL** register. Also, be sure to enable the same pins for Open Drain operation.
- **4.** Initialize the I²C Master by writing the **I2CMCR** register with a value of 0x0000.0020.
- **5.** Set the desired SCL clock speed of 100 Kbps by writing the **I2CMTPR** register with the correct value. The value written to the **I2CMTPR** register represents the number of system clock periods in one SCL clock period. The TPR value is determined by the following equation:

```
TPR = (System Clock / (2 * (SCL_LP + SCL_HP) * SCL_CLK)) - 1;

TPR = (20MHz / (2 * (6 + 4) * 100000)) - 1;

TPR = 9
```

Write the **I2CMTPR** register with the value of 0x0000.0009.

- **6.** Specify the slave address of the master and that the next operation will be a Send by writing the **I2CMSA** register with a value of 0x0000.0076. This sets the slave address to 0x3B.
- Place data (byte) to be sent in the data register by writing the I2CMDR register with the desired data.
- **8.** Initiate a single byte send of the data from Master to Slave by writing the **I2CMCS** register with a value of 0x0000.0007 (STOP, START, RUN).
- **9.** Wait until the transmission completes by polling the **I2CMCS** register's BUSBSY bit until it has been cleared.

13.5 Register Map

Table 13-4 on page 495 lists the I²C registers. All addresses given are relative to the I²C base addresses for the master and slave:

■ I²C 0: 0x4002.0000

Note that the I²C module clock must be enabled before the registers can be programmed (see page 215). There must be a delay of 3 system clocks after the I²C module clock is enabled before any I²C module registers are accessed.

The hw_i2c.h file in the StellarisWare[®] Driver Library uses a base address of 0x800 for the I²C slave registers. Be aware when using registers with offsets between 0x800 and 0x818 that StellarisWare uses an offset between 0x000 and 0x018 with the slave base address.

Table 13-4. Inter-Integrated Circuit (I²C) Interface Register Map

Offset	Name	Туре	Reset	Description	See page
I ² C Maste	r	`			,
0x000	I2CMSA	R/W	0x0000.0000	I2C Master Slave Address	497
0x004	I2CMCS	R/W	0x0000.0000	I2C Master Control/Status	498
0x008	I2CMDR	R/W	0x0000.0000	I2C Master Data	502
0x00C	I2CMTPR	R/W	0x0000.0001	I2C Master Timer Period	503
0x010	I2CMIMR	R/W	0x0000.0000	I2C Master Interrupt Mask	504
0x014	I2CMRIS	RO	0x0000.0000	I2C Master Raw Interrupt Status	505
0x018	I2CMMIS	RO	0x0000.0000	I2C Master Masked Interrupt Status	506
0x01C	I2CMICR	WO	0x0000.0000	I2C Master Interrupt Clear	507
0x020	I2CMCR	R/W	0x0000.0000	I2C Master Configuration	508

Table 13-4. Inter-Integrated Circuit (I²C) Interface Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
I ² C Slave	,				·
0x800	I2CSOAR	R/W	0x0000.0000	I2C Slave Own Address	510
0x804	I2CSCSR	RO	0x0000.0000	I2C Slave Control/Status	511
0x808	I2CSDR	R/W	0x0000.0000	I2C Slave Data	513
0x80C	I2CSIMR	R/W	0x0000.0000	I2C Slave Interrupt Mask	514
0x810	I2CSRIS	RO	0x0000.0000	I2C Slave Raw Interrupt Status	515
0x814	I2CSMIS	RO	0x0000.0000	I2C Slave Masked Interrupt Status	516
0x818	I2CSICR	WO	0x0000.0000	I2C Slave Interrupt Clear	517

13.6 Register Descriptions (I²C Master)

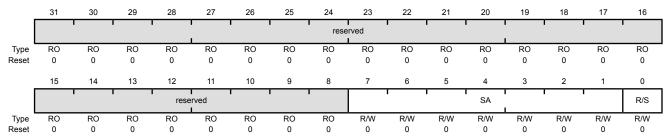
The remainder of this section lists and describes the I²C master registers, in numerical order by address offset. See also "Register Descriptions (I²C Slave)" on page 509.

Register 1: I²C Master Slave Address (I2CMSA), offset 0x000

This register consists of eight bits: seven address bits (A6-A0), and a Receive/Send bit, which determines if the next operation is a Receive (High), or Send (Low).

I2C Master Slave Address (I2CMSA)

I2C 0 base: 0x4002.0000 Offset 0x000 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:1	SA	R/W	0	I ² C Slave Address This field specifies bits A6 through A0 of the slave address.
0	R/S	R/W	0	Receive/Send The \mathbb{R}/\mathbb{S} bit specifies if the next operation is a Receive (High) or Send (Low).

Value Description

Send.

Receive.

Register 2: I²C Master Control/Status (I2CMCS), offset 0x004

This register accesses four control bits when written, and accesses seven status bits when read.

The status register consists of seven bits, which when read determine the state of the I²C bus controller.

The control register consists of four bits: the RUN, START, STOP, and ACK bits. The START bit causes the generation of the START, or REPEATED START condition.

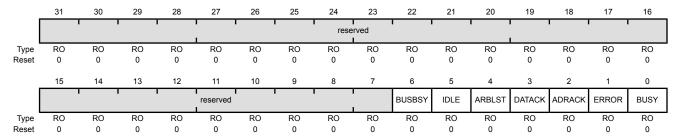
The STOP bit determines if the cycle stops at the end of the data cycle, or continues on to a burst. To generate a single send cycle, the I^2C Master Slave Address (I2CMSA) register is written with the desired address, the R/S bit is set to 0, and the Control register is written with ACK=X (0 or 1), STOP=1, START=1, and RUN=1 to perform the operation and stop. When the operation is completed (or aborted due an error), the interrupt pin becomes active and the data may be read from the I2CMDR register. When the I^2C module operates in Master receiver mode, the ACK bit must be set normally to logic 1. This causes the I^2C bus controller to send an acknowledge automatically after each byte. This bit must be reset when the I^2C bus controller requires no further data to be sent from the slave transmitter.

Reads

I2C Master Control/Status (I2CMCS)

I2C 0 base: 0x4002.0000 Offset 0x004

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	BUSBSY	RO	0	Bus Busy This bit specifies the state of the I ² C bus. If set, the bus is busy; otherwise, the bus is idle. The bit changes based on the START and STOP conditions.
5	IDLE	RO	0	$\rm I^2C$ Idle This bit specifies the $\rm I^2C$ controller state. If set, the controller is idle; otherwise the controller is not idle.
4	ARBLST	RO	0	Arbitration Lost This bit specifies the result of bus arbitration. If set, the controller lost arbitration; otherwise, the controller won arbitration.

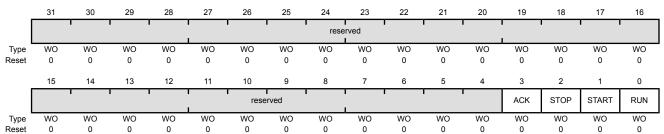
Bit/Field	Name	Туре	Reset	Description
3	DATACK	RO	0	Acknowledge Data This bit specifies the result of the last data operation. If set, the transmitted data was not acknowledged; otherwise, the data was acknowledged.
2	ADRACK	RO	0	Acknowledge Address This bit specifies the result of the last address operation. If set, the transmitted address was not acknowledged; otherwise, the address was acknowledged.
1	ERROR	RO	0	Error This bit specifies the result of the last bus operation. If set, an error occurred on the last operation; otherwise, no error was detected. The error can be from the slave address not being acknowledged or the transmit data not being acknowledged.
0	BUSY	RO	0	I ² C Busy This bit specifies the state of the controller. If set, the controller is busy; otherwise, the controller is idle. When the BUSY bit is set, the other status

bits are not valid.

Writes

I2C Master Control/Status (I2CMCS)

I2C 0 base: 0x4002.0000 Offset 0x004 Type WO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	WO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	ACK	WO	0	Data Acknowledge Enable
				When set, causes received data byte to be acknowledged automatically by the master. See field decoding in Table 13-5 on page 500.
2	STOP	WO	0	Generate STOP
				When set, causes the generation of the STOP condition. See field decoding in Table 13-5 on page 500.
1	START	WO	0	Generate START
				When set, causes the generation of a START or repeated START condition. See field decoding in Table 13-5 on page 500.

Bit/Field	Name	Type	Reset	Description
0	RUN	WO	0	I ² C Master Enable

When set, allows the master to send or receive data. See field decoding in Table 13-5 on page 500.

Table 13-5. Write Field Decoding for I2CMCS[3:0] Field (Sheet 1 of 3)

Current	I2CMSA[0]	2CMSA[0] I2CMCS[3:0]			Description	
State	R/S	ACK	STOP	START	RUN	Description
	0	X ^a	0	1	1	START condition followed by SEND (master goes to the Master Transmit state).
	0	Х	1	1	1	START condition followed by a SEND and STOP condition (master remains in Idle state).
	1	0	0	1	1	START condition followed by RECEIVE operation with negative ACK (master goes to the Master Receive state).
Idle	1	0	1	1	1	START condition followed by RECEIVE and STOP condition (master remains in Idle state).
	1	1	0	1	1	START condition followed by RECEIVE (master goes to the Master Receive state).
	1	1	1	1	1	Illegal.
	All other co	mbination	s not listed	are non-op	erations.	NOP.
	Х	Х	0	0	1	SEND operation (master remains in Master Transmit state).
	Х	Х	1	0	0	STOP condition (master goes to Idle state).
	Х	Х	1	0	1	SEND followed by STOP condition (master goes to Idle state).
	0	Х	0	1	1	Repeated START condition followed by a SEND (master remains in Master Transmit state).
Master	0	Х	1	1	1	Repeated START condition followed by SEND and STOP condition (master goes to Idle state).
Transmit	1	0	0	1	1	Repeated START condition followed by a RECEIVE operation with a negative ACK (master goes to Master Receive state).
	1	0	1	1	1	Repeated START condition followed by a SEND and STOP condition (master goes to Idle state).
	1	1	0	1	1	Repeated START condition followed by RECEIVE (master goes to Master Receive state).
	1	1	1	1	1	Illegal.
	All other co	mbination	s not listed	are non-or	perations.	NOP.

Table 13-5. Write Field Decoding for I2CMCS[3:0] Field (Sheet 1 of 3) (continued)

Current	I2CMSA[0]		I2CMC	S[3:0]		- Description
State	R/S ACK STOP START RUN		RUN	Description		
	Х	0	0	0	1	RECEIVE operation with negative ACK (master remains in Master Receive state).
	Х	Х	1	0	0	STOP condition (master goes to Idle state). ^b
	Х	0	1	0	1	RECEIVE followed by STOP condition (master goes to Idle state).
	Х	1	0	0	1	RECEIVE operation (master remains in Master Receive state).
	Х	1	1	0	1	Illegal.
Master Receive	1	0	0	1	1	Repeated START condition followed by RECEIVE operation with a negative ACK (master remains in Master Receive state).
	1	0	1	1	1	Repeated START condition followed by RECEIVE and STOP condition (master goes to Idle state).
	1	1	0	1	1	Repeated START condition followed by RECEIVE (master remains in Master Receive state).
	0	Х	0	1	1	Repeated START condition followed by SEND (master goes to Master Transmit state).
	0	Х	1	1	1	Repeated START condition followed by SEND and STOP condition (master goes to Idle state).
	All other co	mbinations	s not listed	are non-op	erations.	NOP.

a. An X in a table cell indicates the bit can be 0 or 1.

b. In Master Receive mode, a STOP condition should be generated only after a Data Negative Acknowledge executed by the master or an Address Negative Acknowledge executed by the slave.

Register 3: I²C Master Data (I2CMDR), offset 0x008

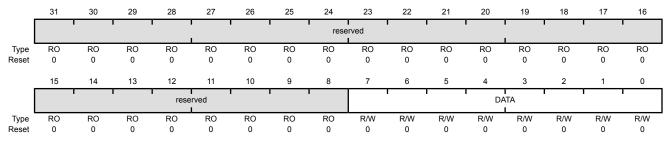
Important: This register is read-sensitive. See the register description for details.

This register contains the data to be transmitted when in the Master Transmit state, and the data received when in the Master Receive state.

I2C Master Data (I2CMDR)

I2C 0 base: 0x4002.0000 Offset 0x008

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DATA	R/W	0x00	Data Transferred

Data transferred during transaction.

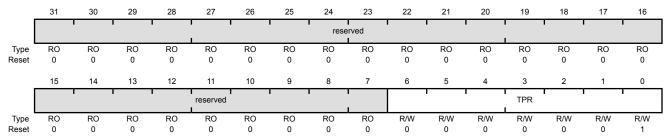
Register 4: I²C Master Timer Period (I2CMTPR), offset 0x00C

This register specifies the period of the SCL clock.

Caution – Take care not to set bit 7 when accessing this register as unpredictable behavior can occur.

I2C Master Timer Period (I2CMTPR)

I2C 0 base: 0x4002.0000 Offset 0x00C Type R/W, reset 0x0000.0001



Bit/Field	Name	Type	Reset	Description
31:7	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6:0	TPR	R/W	0x1	SCL Clock Period

This field specifies the period of the SCL clock.

SCL_PRD = 2*(1 + TPR)*(SCL_LP + SCL_HP)*CLK_PRD

where:

 SCL_PRD is the SCL line period (I²C clock).

TPR is the Timer Period register value (range of 1 to 127).

 SCL_LP is the SCL Low period (fixed at 6).

SCL_HP is the SCL High period (fixed at 4).

Register 5: I²C Master Interrupt Mask (I2CMIMR), offset 0x010

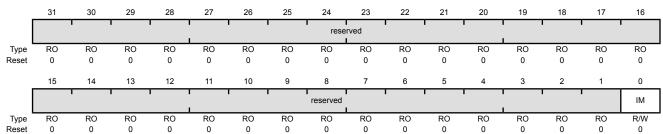
This register controls whether a raw interrupt is promoted to a controller interrupt.

I2C Master Interrupt Mask (I2CMIMR)

I2C 0 base: 0x4002.0000

Offset 0x010

Type R/W, reset 0x0000.0000



Bit/Field	Name	туре	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	IM	R/W	0	Interrupt Mask

This bit controls whether a raw interrupt is promoted to a controller interrupt. If set, the interrupt is not masked and the interrupt is promoted; otherwise, the interrupt is masked.

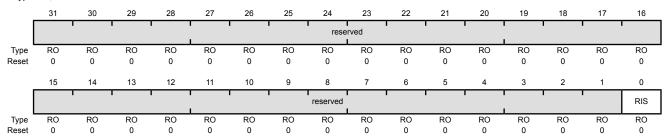
Register 6: I²C Master Raw Interrupt Status (I2CMRIS), offset 0x014

This register specifies whether an interrupt is pending.

I2C Master Raw Interrupt Status (I2CMRIS)

I2C 0 base: 0x4002.0000 Offset 0x014

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	RIS	RO	0	Raw Interrupt Status

This bit specifies the raw interrupt state (prior to masking) of the I²C master block. If set, an interrupt is pending; otherwise, an interrupt is not pending.

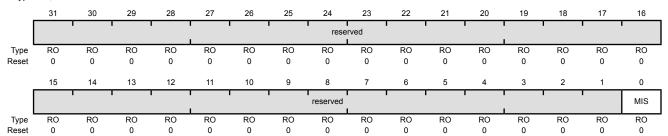
Register 7: I²C Master Masked Interrupt Status (I2CMMIS), offset 0x018

This register specifies whether an interrupt was signaled.

I2C Master Masked Interrupt Status (I2CMMIS)

I2C 0 base: 0x4002.0000 Offset 0x018

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	MIS	RO	0	Masked Interrupt Status

This bit specifies the raw interrupt state (after masking) of the I²C master block. If set, an interrupt was signaled; otherwise, an interrupt has not been generated since the bit was last cleared.

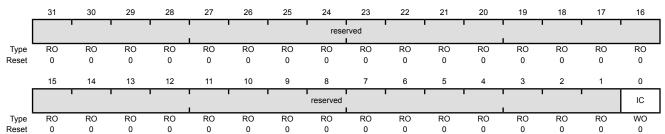
Register 8: I²C Master Interrupt Clear (I2CMICR), offset 0x01C

This register clears the raw interrupt.

I2C Master Interrupt Clear (I2CMICR)

I2C 0 base: 0x4002.0000 Offset 0x01C

Type WO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	IC	WO	0	Interrupt Clear

This bit controls the clearing of the raw interrupt. A write of 1 clears the interrupt; otherwise, a write of 0 has no affect on the interrupt state. A read of this register returns no meaningful data.

Register 9: I²C Master Configuration (I2CMCR), offset 0x020

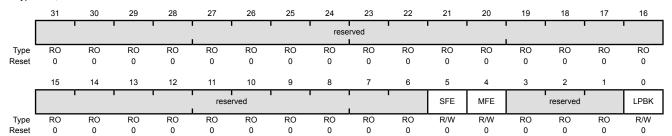
This register configures the mode (Master or Slave) and sets the interface for test mode loopback.

I2C Master Configuration (I2CMCR)

I2C 0 base: 0x4002.0000

Offset 0x020

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	SFE	R/W	0	I ² C Slave Function Enable
				This bit specifies whether the interface may operate in Slave mode. If set, Slave mode is enabled; otherwise, Slave mode is disabled.
4	MFE	R/W	0	I ² C Master Function Enable
				This bit specifies whether the interface may operate in Master mode. If set, Master mode is enabled; otherwise, Master mode is disabled and the interface clock is disabled.
3:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	LPBK	R/W	0	I ² C Loopback

This bit specifies whether the interface is operating normally or in Loopback mode. If set, the device is put in a test mode loopback configuration; otherwise, the device operates normally.

13.7 Register Descriptions (I²C Slave)

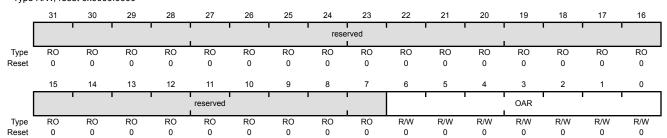
The remainder of this section lists and describes the I^2C slave registers, in numerical order by address offset. See also "Register Descriptions (I^2C Master)" on page 496.

Register 10: I²C Slave Own Address (I2CSOAR), offset 0x800

This register consists of seven address bits that identify the Stellaris I²C device on the I²C bus.

I2C Slave Own Address (I2CSOAR)

I2C 0 base: 0x4002.0000 Offset 0x800 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:7	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6:0	OAR	R/W	0x00	I ² C Slave Own Address

This field specifies bits A6 through A0 of the slave address.

Register 11: I²C Slave Control/Status (I2CSCSR), offset 0x804

This register accesses one control bit when written, and three status bits when read.

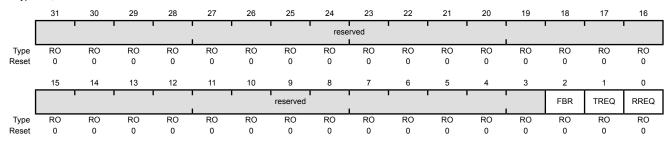
The read-only Status register consists of three bits: the FBR, RREQ, and TREQ bits. The First Byte Received (FBR) bit is set only after the Stellaris device detects its own slave address and receives the first data byte from the I^2C master. The Receive Request (RREQ) bit indicates that the Stellaris I^2C device has received a data byte from an I^2C master. Read one data byte from the I^2C Slave Data (I2CSDR) register to clear the RREQ bit. The Transmit Request (TREQ) bit indicates that the Stellaris I^2C device is addressed as a Slave Transmitter. Write one data byte into the I^2C Slave Data (I2CSDR) register to clear the TREQ bit.

The write-only Control register consists of one bit: the DA bit. The DA bit enables and disables the Stellaris I^2C slave operation.

Reads

I2C Slave Control/Status (I2CSCSR)

I2C 0 base: 0x4002.0000 Offset 0x804 Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:3	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	FBR	RO	0	First Byte Received
				Indicates that the first byte following the slave's own address is received. This bit is only valid when the \mathtt{RREQ} bit is set, and is automatically cleared when data has been read from the <code>I2CSDR</code> register.
				Note: This bit is not used for slave transmit operations.
1	TREQ	RO	0	Transmit Request
				This bit specifies the state of the I^2C slave with regards to outstanding transmit requests. If set, the I^2C unit has been addressed as a slave transmitter and uses clock stretching to delay the master until data has been written to the $I2CSDR$ register. Otherwise, there is no outstanding transmit request.
0	RREQ	RO	0	Receive Request This bit specifies the status of the I^2C slave with regards to outstanding

data is outstanding.

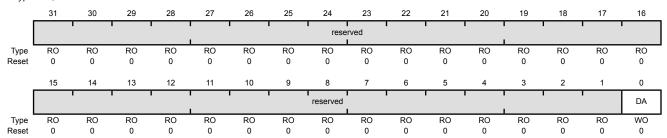
receive requests. If set, the I^2C unit has outstanding receive data from the I^2C master and uses clock stretching to delay the master until the data has been read from the I^2CSDR register. Otherwise, no receive

Writes

I2C Slave Control/Status (I2CSCSR)

I2C 0 base: 0x4002.0000 Offset 0x804

Type WO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	DA	WO	0	Device Active

Value Description

- Disables the I²C slave operation.
- Enables the I²C slave operation.

Once this bit has been set, it should not be set again unless it has been cleared by writing a 0 or by a reset, otherwise transfer failures may occur.

Register 12: I²C Slave Data (I2CSDR), offset 0x808

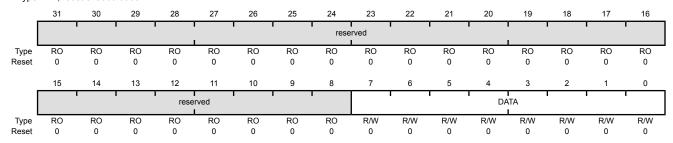
Important: This register is read-sensitive. See the register description for details.

This register contains the data to be transmitted when in the Slave Transmit state, and the data received when in the Slave Receive state.

I2C Slave Data (I2CSDR)

I2C 0 base: 0x4002.0000 Offset 0x808

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DATA	R/W	0x0	Data for Transfer

This field contains the data for transfer during a slave receive or transmit operation.

Register 13: I²C Slave Interrupt Mask (I2CSIMR), offset 0x80C

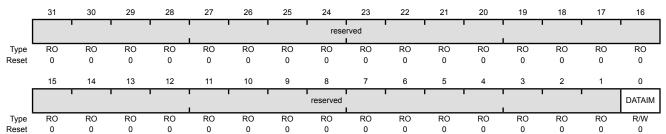
This register controls whether a raw interrupt is promoted to a controller interrupt.

I2C Slave Interrupt Mask (I2CSIMR)

I2C 0 base: 0x4002.0000

Offset 0x80C

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	DATAIM	R/W	0	Data Interrupt Mask

This bit controls whether the raw interrupt for data received and data

requested is promoted to a controller interrupt. If set, the interrupt is not masked and the interrupt is promoted; otherwise, the interrupt is masked.

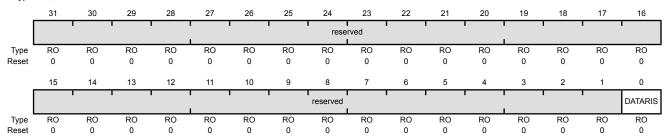
Register 14: I²C Slave Raw Interrupt Status (I2CSRIS), offset 0x810

This register specifies whether an interrupt is pending.

I2C Slave Raw Interrupt Status (I2CSRIS)

I2C 0 base: 0x4002.0000 Offset 0x810

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	DATARIS	RO	0	Data Raw Interrupt Status

This bit specifies the raw interrupt state for data received and data

requested (prior to masking) of the I²C slave block. If set, an interrupt is pending; otherwise, an interrupt is not pending.

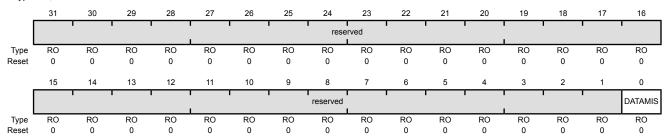
Register 15: I²C Slave Masked Interrupt Status (I2CSMIS), offset 0x814

This register specifies whether an interrupt was signaled.

I2C Slave Masked Interrupt Status (I2CSMIS)

I2C 0 base: 0x4002.0000 Offset 0x814

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	DATAMIS	RO	0	Data Masked Interrupt Status

Data Masked Interrupt Status

This bit specifies the interrupt state for data received and data requested (after masking) of the I²C slave block. If set, an interrupt was signaled; otherwise, an interrupt has not been generated since the bit was last cleared.

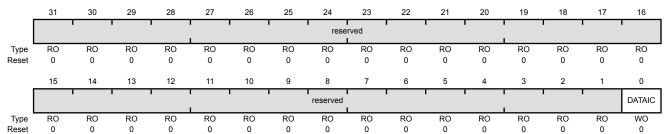
Register 16: I²C Slave Interrupt Clear (I2CSICR), offset 0x818

This register clears the raw interrupt. A read of this register returns no meaningful data.

I2C Slave Interrupt Clear (I2CSICR)

I2C 0 base: 0x4002.0000 Offset 0x818

Type WO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	DATAIC	WO	0	Data Interrupt Clear

This bit controls the clearing of the raw interrupt for data received and data requested. When set, it clears the DATARIS interrupt bit; otherwise, it has no effect on the DATARIS bit value.

14 Ethernet Controller

The Stellaris[®] Ethernet Controller consists of a fully integrated media access controller (MAC) and network physical (PHY) interface. The Ethernet Controller conforms to *IEEE 802.3* specifications and fully supports 10BASE-T and 100BASE-TX standards.

The Stellaris Ethernet Controller module has the following features:

- Conforms to the IEEE 802.3-2002 specification
 - 10BASE-T/100BASE-TX IEEE-802.3 compliant. Requires only a dual 1:1 isolation transformer interface to the line
 - 10BASE-T/100BASE-TX ENDEC, 100BASE-TX scrambler/descrambler
 - Full-featured auto-negotiation
- Multiple operational modes
 - Full- and half-duplex 100 Mbps
 - Full- and half-duplex 10 Mbps
 - Power-saving and power-down modes
- Highly configurable
 - Programmable MAC address
 - LED activity selection
 - Promiscuous mode support
 - CRC error-rejection control
 - User-configurable interrupts
- Physical media manipulation
 - Automatic MDI/MDI-X cross-over correction
 - Register-programmable transmit amplitude
 - Automatic polarity correction and 10BASE-T signal reception

14.1 Block Diagram

As shown in Figure 14-1 on page 519, the Ethernet Controller is functionally divided into two layers: the Media Access Controller (MAC) layer and the Network Physical (PHY) layer. These layers correspond to the OSI model layers 2 and 1. The CPU accesses the Ethernet Controller via the MAC layer. The MAC layer provides transmit and receive processing for Ethernet frames. The MAC layer also provides the interface to the PHY layer via an internal Media Independent Interface (MII). The PHY layer communicates with the Ethernet bus.

Figure 14-1. Ethernet Controller

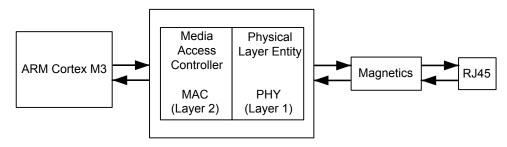
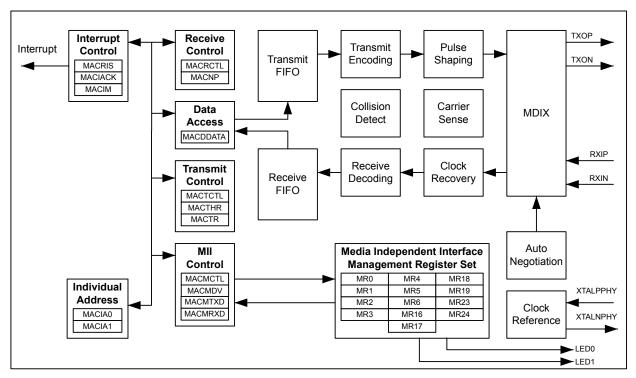


Figure 14-2 on page 519 shows more detail of the internal structure of the Ethernet Controller and how the register set relates to various functions.

Figure 14-2. Ethernet Controller Block Diagram



14.2 Signal Description

Table 14-1 on page 520 and Table 14-2 on page 520 list the external signals of the Ethernet Controller and describe the function of each. The Ethernet LED signals are alternate functions for GPIO signals and default to be GPIO signals at reset. The column in the table below titled "Pin Assignment" lists the GPIO pin placement for the LED signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the LED function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258. The remaining signals (with the word "fixed" in the Pin Mux/Pin Assignment column) have a fixed pin assignment and function.

Table 14-1. Ethernet Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ERBIAS	41	I	Analog	12.4-k Ω resistor (1% precision) used internally for Ethernet PHY.
GNDPHY	42 85 86	-	Power	GND of the Ethernet PHY.
LED0	59	0	TTL	Ethernet LED 0.
LED1	60	0	TTL	Ethernet LED 1.
MDIO	58	I/O	TTL	MDIO of the Ethernet PHY.
RXIN	37	I	Analog	RXIN of the Ethernet PHY.
RXIP	40	ļ	Analog	RXIP of the Ethernet PHY.
TXON	46	0	Analog	TXON of the Ethernet PHY.
TXOP	43	0	Analog	TXOP of the Ethernet PHY.
VCCPHY	36 83 84	-	Power	VCC of the Ethernet PHY.
XTALNPHY	17	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
XTALPPHY	16	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 14-2. Ethernet Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ERBIAS	К3	I	Analog	12.4-k Ω resistor (1% precision) used internally for Ethernet PHY.
GNDPHY	C8 C9 K4	-	Power	GND of the Ethernet PHY.
LED0	J12	0	TTL	Ethernet LED 0.
LED1	J11	0	TTL	Ethernet LED 1.
MDIO	L9	I/O	TTL	MDIO of the Ethernet PHY.
RXIN	L7	1	Analog	RXIN of the Ethernet PHY.
RXIP	M7	I	Analog	RXIP of the Ethernet PHY.
TXON	L8	0	Analog	TXON of the Ethernet PHY.
TXOP	M8	0	Analog	TXOP of the Ethernet PHY.
VCCPHY	C10 D10 D11	-	Power	VCC of the Ethernet PHY.
XTALNPHY	J1	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
XTALPPHY	J2	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

14.3 Functional Description

Note: A 12.4-k Ω resistor should be connected between the ERBIAS and ground. The 12.4-k Ω resistor should have a 1% tolerance and should be located in close proximity to the ERBIAS pin. Power dissipation in the resistor is low, so a chip resistor of any geometry may be used.

The functional description of the Ethernet Controller is discussed in the following sections.

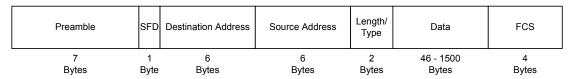
14.3.1 MAC Operation

The following sections decribe the operation of the MAC unit, including an overview of the Ethernet frame format, the MAC layer FIFOs, Ethernet transmission and reception options, and LED indicators.

14.3.1.1 Ethernet Frame Format

Ethernet data is carried by Ethernet frames. The basic frame format is shown in Figure 14-3 on page 521.

Figure 14-3. Ethernet Frame



The seven fields of the frame are transmitted from left to right. The bits within the frame are transmitted from least to most significant bit.

■ Preamble

The Preamble field is used to synchronize with the received frame's timing. The preamble is 7 octets long.

Start Frame Delimiter (SFD)

The SFD field follows the preamble pattern and indicates the start of the frame. Its value is 1010.1011.

■ Destination Address (DA)

This field specifies destination addresses for which the frame is intended. The LSB (bit 16 of DA oct 1 in the frame, see Table 14-3 on page 523) of the DA determines whether the address is an individual (0), or group/multicast (1) address.

Source Address (SA)

The source address field identifies the station from which the frame was initiated.

■ Length/Type Field

The meaning of this field depends on its numeric value. This field can be interpreted as length or type code. The maximum length of the data field is 1500 octets. If the value of the Length/Type field is less than or equal to 1500 decimal, it indicates the number of MAC client data octets. If the value of this field is greater than or equal to 1536 decimal, then it is type interpretation. The meaning of the Length/Type field when the value is between 1500 and 1536 decimal is unspecified by the IEEE 802.3 standard. However, the Ethernet Controller assumes type interpretation if the

value of the Length/Type field is greater than 1500 decimal. The definition of the Type field is specified in the IEEE 802.3 standard. The first of the two octets in this field is most significant.

Data

The data field is a sequence of octets that is at least 46 in length, up to 1500 in length. Full data transparency is provided so any values can appear in this field. A minimum frame size of 46 octets is required to meet the IEEE standard. If the frame size is too small, the Ethernet Controller automatically appends extra bits (a pad), thus the pad can have a size of 0 to 46 octets. Data padding can be disabled by clearing the PADEN bit in the **Ethernet MAC Transmit Control** (MACTCTL) register.

For the Ethernet Controller, data sent/received can be larger than 1500 bytes without causing a Frame Too Long error. Instead, a FIFO overrun error is reported using the FOV bit in the **Ethernet MAC Raw Interrupt Status(MACRIS)** register when the frame received is too large to fit into the Ethernet Controller's 2K RAM.

Frame Check Sequence (FCS)

The frame check sequence carries the cyclic redundancy check (CRC) value. The CRC is computed over the destination address, source address, length/type, and data (including pad) fields using the CRC-32 algorithm. The Ethernet Controller computes the FCS value one nibble at a time. For transmitted frames, this field is automatically inserted by the MAC layer, unless disabled by clearing the CRC bit in the **MACTCTL** register. For received frames, this field is automatically checked. If the FCS does not pass, the frame is not placed in the RX FIFO, unless the FCS check is disabled by clearing the BADCRC bit in the **MACRCTL** register.

14.3.1.2 MAC Layer FIFOs

The Ethernet Controller is capable of simultaneous transmission and reception. This feature is enabled by setting the DUPLEX bit in the **MACTCTL** register.

For Ethernet frame transmission, a 2 KB transmit FIFO is provided that can be used to store a single frame. While the *IEEE 802.3 specification* limits the size of an Ethernet frame's payload section to 1500 Bytes, the Ethernet Controller places no such limit. The full buffer can be used, for a payload of up to 2032 bytes (as the first 16 bytes in the FIFO are reserved for destination address, source address and length/type information).

For Ethernet frame reception, a 2-KB receive FIFO is provided that can be used to store multiple frames, up to a maximum of 31 frames. If a frame is received, and there is insufficient space in the RX FIFO, an overflow error is indicated using the FOV bit in the **MACRIS** register.

For details regarding the TX and RX FIFO layout, refer to Table 14-3 on page 523. Please note the following difference between TX and RX FIFO layout. For the TX FIFO, the Data Length field in the first FIFO word refers to the Ethernet frame data payload, as shown in the 5th to nth FIFO positions. For the RX FIFO, the Frame Length field is the total length of the received Ethernet frame, including the Length/Type bytes and the FCS bits.

If FCS generation is disabled by clearing the CRC bit in the **MACTCTL** register, the last word in the TX FIFO must contain the FCS bytes for the frame that has been written to the FIFO.

Also note that if the length of the data payload section is not a multiple of 4, the FCS field is not be aligned on a word boundary in the FIFO. However, for the RX FIFO the beginning of the next frame is always on a word boundary.

Table 14-3. TX & RX FIFO Organization

FIFO Word Read/Write Sequence	Word Bit Fields	TX FIFO (Write)	RX FIFO (Read)				
1st	7:0	Data Length Least Significant Byte	Frame Length Least Significant Byte				
	15:8	Data Length Most Significant Byte	Frame Length Most Significant Byte				
	23:16	DA	oct 1				
	31:24	DA	oct 2				
2nd	7:0	DA	oct 3				
	15:8	DA	oct 4				
	23:16	DA	oct 5				
	31:24	DA	oct 6				
3rd	7:0	SA	oct 1				
	15:8	SA	oct 2				
	23:16	SA	oct 3				
	31:24	SA	SA oct 4				
4th	7:0	SA	SA oct 5				
	15:8	SA	SA oct 6				
	23:16	Len/Type Most	Len/Type Most Significant Byte				
	31:24	Len/Type Least	Significant Byte				
5th to nth	7:0	data	oct n				
	15:8	data d	data oct n+1				
	23:16	data d	data oct n+2				
	31:24	data d	data oct n+3				
last	7:0	FC	S 1				
	15:8	FC	S 2				
	23:16	FC	S 3				
	31:24	FC	S 4				

Note: If the CRC bit in the **MACTCTL** register is clear, the FCS bytes must be written with the correct CRC. If the CRC bit is set, the Ethernet Controller automatically writes the FCS bytes.

14.3.1.3 Ethernet Transmission Options

At the MAC layer, the transmitter can be configured for both full-duplex and half-duplex operation by using the <code>DUPLEX</code> bit in the **MACTCTL** register.

The Ethernet Controller automatically generates and inserts the Frame Check Sequence (FCS) at the end of the transmit frame when the CRC bit in the **MACTCTL** register is set. However, for test purposes, this feature can be disabled in order to generate a frame with an invalid CRC by clearing the CRC bit.

The *IEEE 802.3 specification* requires that the Ethernet frame payload section be a minimum of 46 bytes. The Ethernet Controller automatically pads the data section if the payload data section loaded into the FIFO is less than the minimum 46 bytes when the PADEN bit in the **MACTCTL** register is set. This feature can be disabled by clearing the PADEN bit.

The transmitter must be enabled by setting the TXEN bit in the TCTL register.

14.3.1.4 Ethernet Reception Options

The Ethernet Controller RX FIFO should be cleared during software initialization. The receiver should first be disabled by clearing the RXEN bit in the **Ethernet MAC Receive Control (MACRCTL)** register, then the FIFO can be cleared by setting the RSTFIFO bit in the **MACRCTL** register.

The receiver automatically rejects frames that contain bad CRC values in the FCS field. In this case, a Receive Error interrupt is generated and the receive data is lost. To accept all frames, clear the BADCRC bit in the **MACRCTL** register.

In normal operating mode, the receiver accepts only those frames that have a destination address that matches the address programmed into the **Ethernet MAC Individual Address 0 (MACIA0)** and **Ethernet MAC Individual Address 1 (MACIA1)** registers. However, the Ethernet receiver can also be configured for Promiscuous and Multicast modes by setting the PRMS and AMUL bits in the **MACRCTL** register.

14.3.2 Internal MII Operation

For the MII management interface to function properly, the MDIO signal must be connected through a 10k Ω pull-up resistor to the +3.3 V supply. Failure to connect this pull-up resistor prevents management transactions on this internal MII to function. Note that it is possible for data transmission across the MII to still function since the PHY layer auto-negotiates the link parameters by default.

For the MII management interface to function properly, the internal clock must be divided down from the system clock to a frequency no greater than 2.5 MHz. The **Ethernet MAC Management Divider (MACMDV)** register contains the divider used for scaling down the system clock. See page 543 for more details about the use of this register.

14.3.3 PHY Operation

The Physical Layer (PHY) in the Ethernet Controller includes integrated ENDECs, scrambler/descrambler, dual-speed clock recovery, and full-featured auto-negotiation functions. The transmitter includes an on-chip pulse shaper and a low-power line driver. The receiver has an adaptive equalizer and a baseline restoration circuit required for accurate clock and data recovery. The transceiver interfaces to Category-5 unshielded twisted pair (Cat-5 UTP) cabling for 100BASE-TX applications, and Category-3 unshielded twisted pair (Cat-3 UTP) for 10BASE-T applications. The Ethernet Controller is connected to the line media via dual 1:1 isolation transformers. No external filter is required.

14.3.3.1 Clock Selection

The Ethernet Controller has an on-chip crystal oscillator which can also be driven by an external oscillator. In this mode of operation, a 25-MHz crystal should be connected between the XTALPPHY and XTALNPHY pins. Alternatively, an external 25-MHz clock input can be connected to the XTALPPHY pin. In this mode of operation, a crystal is not required and the XTALNPHY pin must be tied to ground.

14.3.3.2 Auto-Negotiation

The Ethernet Controller supports the auto-negotiation functions of Clause 28 of the *IEEE 802.3* standard for 10/100 Mbps operation over copper wiring. This function is controlled via register settings. The auto-negotiation function is turned on by default, and the ANEGEN bit in the **Ethernet PHY Management Register 0 - Control (MR0)** is set after reset. Software can disable the auto-negotiation function by clearing the ANEGEN bit. The contents of the **Ethernet PHY Management Register - Auto-Negotiation Advertisement (MR4)** are reflected to the Ethernet Controller's link partner during auto-negotiation via fast-link pulse coding.

Once auto-negotiation is complete, the DPLX and RATE bits in the **Ethernet PHY Management Register 18 - Diagnostic (MR18)** register reflect the actual speed and duplex condition. If auto-negotiation fails to establish a link for any reason, the ANEGF bit in the **MR18** register reflects this and auto-negotiation restarts from the beginning. Setting the RANEG bit in the **MR0** register also causes auto-negotiation to restart.

14.3.3.3 Polarity Correction

The Ethernet Controller is capable of either automatic or manual polarity reversal for 10BASE-T and auto-negotiation functions. Bits 4 and 5 (RVSPOL and APOL) in the **Ethernet PHY Management Register 16 - Vendor-Specific (MR16)** control this feature. The default is automatic mode, where APOL is clear and RVSPOL indicates if the detection circuitry has inverted the input signal. To enter manual mode, APOL should be set. In manual mode RVSPOL controls the signal polarity.

14.3.3.4 MDI/MDI-X Configuration

The Ethernet Controller supports the MDI/MDI-X configuration as defined in *IEEE 802.3-2002* specification. The MDI/MDI-X configuration eliminates the need for cross-over cables when connecting to another device, such as a hub. The algorithm is controlled via settings in the **Ethernet PHY Management Register 24 - MDI/MIDIX Control (MR24)**. Refer to page 565 for additional details about these settings.

14.3.3.5 Power Management

The PHY has two power-saving modes:

- Power-Down
- Receive Power Management

Power-down mode is activated by setting the PWRDN bit in the **MR0** register. When the PHY is in power-down mode, it consumes minimum power. While in the power-down state, the Ethernet Controller still responds to management transactions.

Receive power management (RXCC mode) is activated by setting the RXCC bit in the **MR16** register. In this mode of operation, the adaptive equalizer, the clock recovery phase lock loop (PLL), and all other receive circuitry are powered down. As soon as a valid signal is detected, all circuits are automatically powered up to resume normal operation. Note that the RXCC mode is not supported during 10BASE-T operation.

14.3.3.6 LED Indicators

The Ethernet Controller supports two LED signals that can be used to indicate various states of operation. These signals are mapped to the LED0 and LED1 pins. By default, these pins are configured as GPIO signals (PF3 and PF2). For the PHY layer to drive these signals, they must be reconfigured to their alternate function. See "General-Purpose Input/Outputs (GPIOs)" on page 258 for additional details. The function of these pins is programmable via the PHY layer **Ethernet PHY Management Register 23 - LED Configuration (MR23)**. Refer to page 564 for additional details on how to program these LED functions.

14.3.4 Interrupts

The Ethernet Controller can generate an interrupt for one or more of the following conditions:

A frame has been received into an empty RX FIFO

- A frame transmission error has occurred
- A frame has been transmitted successfully
- A frame has been received with inadequate room in the RX FIFO (overrun)
- A frame has been received with one or more error conditions (for example, FCS failed)
- An MII management transaction between the MAC and PHY layers has completed
- One or more of the following PHY layer conditions occurs:
 - Auto-Negotiate Complete
 - Remote Fault
 - Link Status Change
 - Link Partner Acknowledge
 - Parallel Detect Fault
 - Page Received
 - Receive Error
 - Jabber Event Detected

14.4 Initialization and Configuration

The following sections describe the hardware and software configuration required to set up the Ethernet Controller.

14.4.1 Hardware Configuration

Figure 14-4 on page 527 shows the proper method for interfacing the Ethernet Controller to a 10/100BASE-T Ethernet jack.

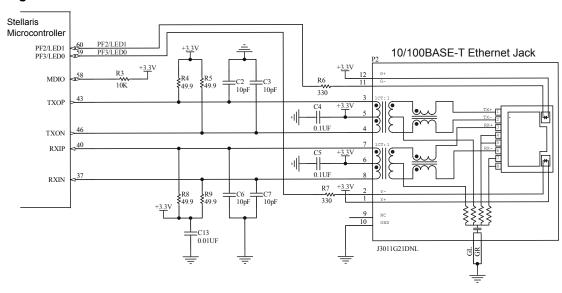


Figure 14-4. Interface to an Ethernet Jack

The following isolation transformers have been tested and are known to successfully interface to the Ethernet PHY layer.

- Isolation Transformers
 - TDK TLA-6T103
 - Bel-Fuse S558-5999-46
 - Halo TG22-3506ND
 - Pulse PE-68515
 - Valor ST6118
 - YCL 20PMT04
- Isolation transformers in low profile packages (0.100 in/2.5 mm or less)
 - TDK TLA-6T118
 - Halo TG110-S050
 - PCA EPF8023G
- Isolation transformers with integrated RJ45 connector
 - TDK TLA-6T704
 - Delta RJS-1A08T089A
- Isolation transformers with integrated RJ45 connector, LEDs and termination resistors
 - Pulse J0011D21B/E
 - Pulse J3011G21DNL

14.4.2 Software Configuration

To use the Ethernet Controller, it must be enabled by setting the EPHY0 and EMACO bits in the **RCGC2** register (see page 221). The following steps can then be used to configure the Ethernet Controller for basic operation.

- 1. Program the **MACDIV** register to obtain a 2.5 MHz clock (or less) on the internal MII. Assuming a 20-MHz system clock, the **MACDIV** value should be 0x03 or greater.
- 2. Program the MACIA0 and MACIA1 register for address filtering.
- **3.** Program the **MACTCTL** register for Auto CRC generation, padding, and full-duplex operation using a value of 0x16.

- **4.** Program the **MACRCTL** register to flush the receive FIFO and reject frames with bad FCS using a value of 0x18.
- **5.** Enable both the Transmitter and Receive by setting the LSB in both the **MACTCTL** and **MACRCTL** registers.
- 6. To transmit a frame, write the frame into the TX FIFO using the **Ethernet MAC Data (MACDATA)** register. Then set the NEWTX bit in the **Ethernet Mac Transmission Request (MACTR)** register to initiate the transmit process. When the NEWTX bit has been cleared, the TX FIFO is available for the next transmit frame.
- 7. To receive a frame, wait for the NPR field in the Ethernet MAC Number of Packets (MACNP) register to be non-zero. Then begin reading the frame from the RX FIFO by using the MACDATA register. To ensure that the entire packet is received, either use the DriverLib EthernetPacketGet() API or compare the number of bytes received to the Length field from the frame to determine when the packet has been completely read.

14.5 Ethernet Register Map

Table 14-4 on page 528 lists the Ethernet MAC registers. All addresses given are relative to the Ethernet MAC base address of 0x4004.8000. Note that the Ethernet module clock must be enabled before the registers can be programmed (see page 221). There must be a delay of 3 system clocks after the Ethernet module clock is enabled before any Ethernet module registers are accessed.

The IEEE 802.3 standard specifies a register set for controlling and gathering status from the PHY layer. The registers are collectively known as the MII Management registers and are detailed in Section 22.2.4 of the IEEE 802.3 specification. Table 14-4 on page 528 also lists these MII Management registers. All addresses given are absolute and are written directly to the REGADR field of the Ethernet MAC Management Control (MACMCTL) register. The format of registers 0 to 15 are defined by the IEEE specification and are common to all PHY layer implementations. The only variance allowed is for features that may or may not be supported by a specific PHY implementation. Registers 16 to 31 are vendor-specific registers, used to support features that are specific to a vendor's PHY implementation. Vendor-specific registers not listed are reserved.

Table 14-4. Ethernet Register Map

Offset	Name	Туре	Reset	Description	See page						
Ethernet	Ethernet MAC										
0x000	MACRIS/MACIACK	R/W1C	0x0000.0000	Ethernet MAC Raw Interrupt Status/Acknowledge	530						
0x004	MACIM	R/W	0x0000.007F	Ethernet MAC Interrupt Mask	533						
0x008	MACRCTL	R/W	0x0000.0008	Ethernet MAC Receive Control	534						
0x00C	MACTCTL	R/W	0x0000.0000	Ethernet MAC Transmit Control	535						
0x010	MACDATA	R/W	0x0000.0000	Ethernet MAC Data	536						
0x014	MACIA0	R/W	0x0000.0000	Ethernet MAC Individual Address 0	538						
0x018	MACIA1	R/W	0x0000.0000	Ethernet MAC Individual Address 1	539						
0x01C	MACTHR	R/W	0x0000.003F	Ethernet MAC Threshold	540						
0x020	MACMCTL	R/W	0x0000.0000	Ethernet MAC Management Control	542						

Table 14-4. Ethernet Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0x024	MACMDV	R/W	0x0000.0080	Ethernet MAC Management Divider	543
0x02C	MACMTXD	R/W	0x0000.0000	Ethernet MAC Management Transmit Data	544
0x030	MACMRXD	R/W	0x0000.0000	Ethernet MAC Management Receive Data	545
0x034	MACNP	RO	0x0000.0000	Ethernet MAC Number of Packets	546
0x038	MACTR	R/W	0x0000.0000	Ethernet MAC Transmission Request	547
MII Mana	gement				
-	MR0	R/W	0x3100	Ethernet PHY Management Register 0 – Control	548
-	MR1	RO	0x7849	Ethernet PHY Management Register 1 – Status	550
-	MR2	RO	0x000E	Ethernet PHY Management Register 2 – PHY Identifier 1	552
-	MR3	RO	0x7237	Ethernet PHY Management Register 3 – PHY Identifier 2	553
-	MR4	R/W	0x01E1	Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement	554
-	MR5	RO	0x0000	Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability	556
-	MR6	RO	0x0000	Ethernet PHY Management Register 6 – Auto-Negotiation Expansion	557
-	MR16	R/W	0x0140	Ethernet PHY Management Register 16 – Vendor-Specific	558
-	MR17	R/W	0x0000	Ethernet PHY Management Register 17 – Interrupt Control/Status	560
-	MR18	RO	0x0000	Ethernet PHY Management Register 18 – Diagnostic	562
-	MR19	R/W	0x4000	Ethernet PHY Management Register 19 – Transceiver Control	563
-	MR23	R/W	0x0010	Ethernet PHY Management Register 23 – LED Configuration	564
-	MR24	R/W	0x00C0	Ethernet PHY Management Register 24 –MDI/MDIX Control	565

14.6 Ethernet MAC Register Descriptions

The remainder of this section lists and describes the Ethernet MAC registers, in numerical order by address offset. Also see "MII Management Register Descriptions" on page 547.

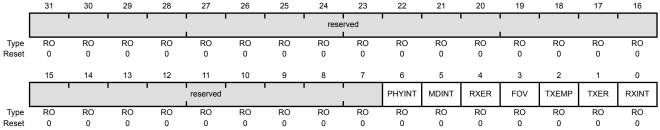
Register 1: Ethernet MAC Raw Interrupt Status/Acknowledge (MACRIS/MACIACK), offset 0x000

The MACRIS/MACIACK register is the interrupt status and acknowledge register. On a read, this register gives the current status value of the corresponding interrupt prior to masking. On a write, setting any bit clears the corresponding interrupt status bit.

Reads

Ethernet MAC Raw Interrupt Status/Acknowledge (MACRIS/MACIACK)

Base 0x4004.8000 Offset 0x000 Type RO, reset 0x0000.0000



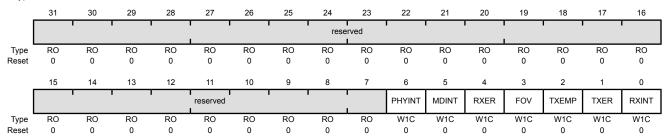
set	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
В	it/Field		Nam	ne	Ту	ре	Reset	Desc	ription							
	31:7		reserv	ved	R	0	0x0000.00	comp	atibility	with futu	ure prod	he value ucts, the dify-write	value of	a reserv		
	6		PHYII	NT	R	0	0	Wher	rred. MF	dicates t	e PHY m	nabled in nust be re rupt.			•	
	5		MDIN	NT	R	0	0	Wher	n set, ind	ion Com dicates thed succe	hat a trar	nsaction	(read or v	write) on	the MII i	nterface
	4		RXE	:R	R	0	0	This I		ates that		r was en e this int				r. The
									receivenly).	e error o	ccurs du	iring the	receptio	n of a fra	ame (100) Mb/s
									he fram		an integ	er numbe	er of byte	es (dribb	le bits) d	ue to an
								■ T	he CR0	C of the f	frame do	es not p	ass the I	FCS che	ck.	
										ith/type f ed as a l		consiste eld.	nt with th	ne frame	data siz	e when
	3		FO\	V	R	0	0	FIFO	Overru	n						
								Wher FIFO	,	dicates t	that an c	verrun w	as enco	untered	on the re	eceive

Bit/Field	Name	Туре	Reset	Description
2	TXEMP	RO	0	Transmit FIFO Empty When set, indicates that the packet was transmitted and that the TX FIFO is empty.
1	TXER	RO	0	Transmit Error When set, indicates that an error was encountered on the transmitter. The possible errors that can cause this interrupt bit to be set are: The data length field stored in the TX FIFO exceeds 2032 decimal (buffer length - 16 bytes of header data). The frame is not sent when this error occurs. The retransmission attempts during the backoff process have exceeded the maximum limit of 16 decimal.
0	RXINT	RO	0	Packet Received When set, indicates that at least one packet has been received and is stored in the receiver FIFO.

Writes

Ethernet MAC Raw Interrupt Status/Acknowledge (MACRIS/MACIACK)

Base 0x4004.8000 Offset 0x000 Type WO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PHYINT	W1C	0	Clear PHY Interrupt Setting this bit clears the PHYINT interrupt in the MACRIS register.
5	MDINT	W1C	0	Clear MII Transaction Complete Setting this bit clears the MDINT interrupt in the MACRIS register.
4	RXER	W1C	0	Clear Receive Error Setting this bit clears the RXER interrupt in the MACRIS register.
3	FOV	W1C	0	Clear FIFO Overrun Setting this bit clears the FOV interrupt in the MACRIS register.

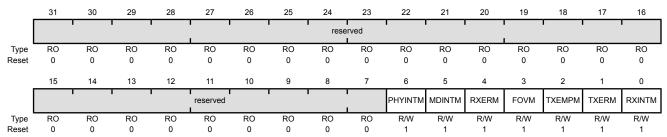
Bit/Field	Name	Type	Reset	Description
2	TXEMP	W1C	0	Clear Transmit FIFO Empty Setting this bit clears the TXEMP interrupt in the MACRIS register.
1	TXER	W1C	0	Clear Transmit Error Setting this bit clears the TXER interrupt in the MACRIS register and resets the TX FIFO write pointer.
0	RXINT	W1C	0	Clear Packet Received Setting this bit clears the RXINT interrupt in the MACRIS register.

Register 2: Ethernet MAC Interrupt Mask (MACIM), offset 0x004

This register allows software to enable/disable Ethernet MAC interrupts. Clearing a bit disables the interrupt, while setting the bit enables it.

Ethernet MAC Interrupt Mask (MACIM)

Base 0x4004.8000 Offset 0x004 Type R/W, reset 0x0000.007F



Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PHYINTM	R/W	1	Mask PHY Interrupt
				Clearing this bit masks the ${\tt PHYINT}$ bit in the MACRIS register from being set.
5	MDINTM	R/W	1	Mask MII Transaction Complete
				Clearing this bit masks the ${\tt MDINT}$ bit in the \textbf{MACRIS} register from being set.
4	RXERM	R/W	1	Mask Receive Error
				Clearing this bit masks the ${\tt RXER}$ bit in the \textbf{MACRIS} register from being set.
3	FOVM	R/W	1	Mask FIFO Overrun
				Clearing this bit masks the ${\tt FOV}$ bit in the \textbf{MACRIS} register from being set.
2	TXEMPM	R/W	1	Mask Transmit FIFO Empty
				Clearing this bit masks the \mathtt{TXEMP} bit in the \textbf{MACRIS} register from being set.
1	TXERM	R/W	1	Mask Transmit Error
				Clearing this bit masks the \mathtt{TXER} bit in the \textbf{MACRIS} register from being set.
0	RXINTM	R/W	1	Mask Packet Received
				Clearing this bit masks the ${\tt RXINT}$ bit in the \textbf{MACRIS} register from being set.

Register 3: Ethernet MAC Receive Control (MACRCTL), offset 0x008

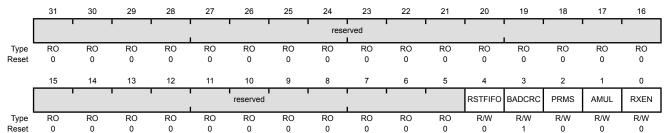
This register configures the receiver and controls the types of frames that are received.

It is important to note that when the receiver is enabled, all valid frames with a broadcast address of FF-FF-FF-FF-FF in the Destination Address field are received and stored in the RX FIFO, even if the AMUL bit is not set.

Ethernet MAC Receive Control (MACRCTL)

Base 0x4004.8000 Offset 0x008

Type R/W, reset 0x0000.0008



Bit/Field	Name	Type	Reset	Description
31:5	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	RSTFIFO	R/W	0	Clear Receive FIFO
				When set, this bit clears the receive FIFO. This should be done when software initialization is performed.
				It is recommended that the receiver be disabled (RXEN = 0), before a reset is initiated (RSTFIFO = 1). This sequence flushes and resets the RX FIFO.
				This bit is automatically cleared when read.
3	BADCRC	R/W	1	Enable Reject Bad CRC
				When set, the BADCRC bit enables the rejection of frames with an incorrectly calculated CRC. If a bad CRC is encountered, the RXER bit in the MACRIS register is set and the receiver FIFO is reset.
2	PRMS	R/W	0	Enable Promiscuous Mode
				When set, the ${\tt PRMS}$ bit enables Promiscuous mode, which accepts all valid frames, regardless of the specified Destination Address.
1	AMUL	R/W	0	Enable Multicast Frames
				When set, the ${\tt AMUL}$ bit enables the reception of multicast frames.
0	RXEN	R/W	0	Enable Receiver
				When set the RXEN bit enables the Ethernet receiver. When this bit is clear, the receiver is disabled and all frames are ignored.

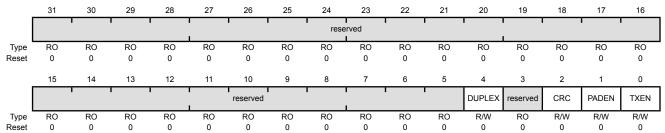
Register 4: Ethernet MAC Transmit Control (MACTCTL), offset 0x00C

This register configures the transmitter and controls the frames that are transmitted.

Ethernet MAC Transmit Control (MACTCTL)

Base 0x4004.8000

Offset 0x00C Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:5	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	DUPLEX	R/W	0	Enable Duplex Mode
				When set, this bit enables Duplex mode, allowing simultaneous transmission and reception.
3	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	CRC	R/W	0	Enable CRC Generation
				When set this bit enables the automatic generation of the CRC and its placement at the end of the packet. If this bit is clear, the frames placed in the TX FIFO are sent exactly as they are written into the FIFO.
				Note that this bit should generally be set.
1	PADEN	R/W	0	Enable Packet Padding
				When set, this bit enables the automatic padding of packets that do not meet the minimum frame size.
				Note that this bit should generally be set.
0	TXEN	R/W	0	Enable Transmitter
				When set, this bit enables the transmitter. When this bit is clear, the transmitter is disabled.

Register 5: Ethernet MAC Data (MACDATA), offset 0x010

Important: This register is read-sensitive. See the register description for details.

This register enables software to access the TX and RX FIFOs.

Reads from this register return the data stored in the RX FIFO from the location indicated by the read pointer. The read pointer is then auto incremented to the next RX FIFO location. Reading from the RX FIFO when a frame has not been received or is in the process of being received will return indeterminate data and not increment the read pointer.

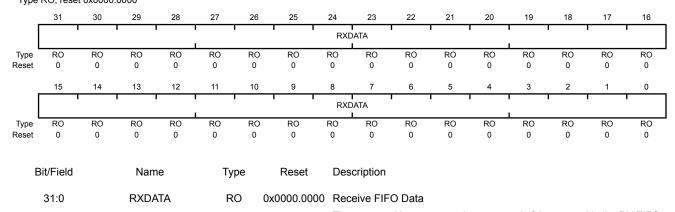
Writes to this register store the data in the TX FIFO at the location indicated by the write pointer. The write pointer is the auto incremented to the next TX FIFO location. Writing more data into the TX FIFO than indicated in the length field will result in the data being lost. Writing less data into the TX FIFO than indicated in the length field will result in indeterminate data being appended to the end of the frame to achieve the indicated length. Attempting to write the next frame into the TX FIFO before transmission of the first has completed will result in the data being lost.

There is no mechanism for randomly accessing bytes in either the RX or TX FIFOs. Data must be read from the RX FIFO sequentially and stored in a buffer for further processing. Once a read has been performed, the data in the FIFO cannot be re-read. Data must be written to the TX FIFO sequentially. If an error is made in placing the frame into the TX FIFO, the write pointer can be reset to the start of the TX FIFO by writing the TXER bit of the MACIACK register and then the data re-written.

Reads

Ethernet MAC Data (MACDATA)

Base 0x4004.8000 Offset 0x010 Type RO, reset 0x0000.0000

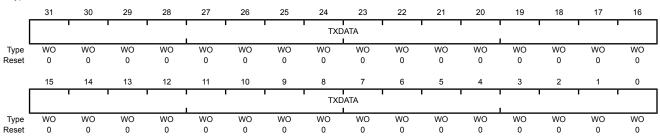


The RXDATA bits represent the next word of data stored in the RX FIFO.

Writes

Ethernet MAC Data (MACDATA)

Base 0x4004.8000 Offset 0x010 Type WO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:0	TXDATA	WO	0x0000.0000	Transmit FIFO Data

The $\ensuremath{\mathtt{TXDATA}}$ bits represent the next word of data to place in the TX FIFO for transmission.

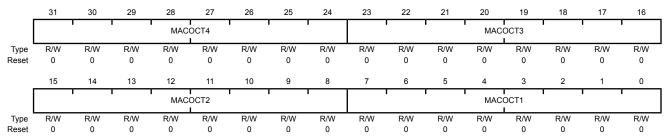
Register 6: Ethernet MAC Individual Address 0 (MACIA0), offset 0x014

This register enables software to program the first four bytes of the hardware MAC address of the Network Interface Card (NIC). (The last two bytes are in MACIA1). The 6-byte Individual Address is compared against the incoming Destination Address fields to determine whether the frame should be received.

Ethernet MAC Individual Address 0 (MACIA0)

Base 0x4004.8000

Offset 0x014
Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:24	MACOCT4	R/W	0x00	MAC Address Octet 4 The MACOCT4 bits represent the fourth octet of the MAC address used to uniquely identify the Ethernet Controller.
23:16	MACOCT3	R/W	0x00	MAC Address Octet 3 The MACOCT3 bits represent the third octet of the MAC address used to uniquely identify the Ethernet Controller.
15:8	MACOCT2	R/W	0x00	MAC Address Octet 2 The MACOCT2 bits represent the second octet of the MAC address used to uniquely identify the Ethernet Controller.
7:0	MACOCT1	R/W	0x00	MAC Address Octet 1

The MACOCT1 bits represent the first octet of the MAC address used to uniquely identify the Ethernet Controller.

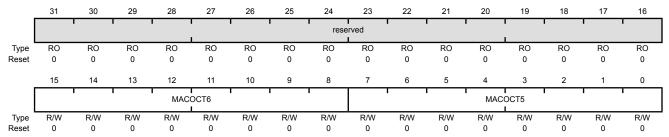
Register 7: Ethernet MAC Individual Address 1 (MACIA1), offset 0x018

This register enables software to program the last two bytes of the hardware MAC address of the Network Interface Card (NIC). (The first four bytes are in MACIAO). The 6-byte IAR is compared against the incoming Destination Address fields to determine whether the frame should be received.

Ethernet MAC Individual Address 1 (MACIA1)

Base 0x4004.8000

Offset 0x018
Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:8	MACOCT6	R/W	0x00	MAC Address Octet 6
				The MACOCT6 bits represent the sixth octet of the MAC address used to uniquely identify each Ethernet Controller.
7:0	MACOCT5	R/W	0x00	MAC Address Octet 5

The MACOCT5 bits represent the fifth octet of the MAC address used to uniquely identify the Ethernet Controller.

Register 8: Ethernet MAC Threshold (MACTHR), offset 0x01C

In order to increase the transmission rate, it is possible to program the Ethernet Controller to begin transmission of the next frame prior to the completion of the transmission of the current frame. Note: Extreme care must be used when implementing this function. Software must be able to guarantee that the complete frame is able to be stored in the transmission FIFO prior to the completion of the transmission frame.

This register enables software to set the threshold level at which the transmission of the frame begins. If the THRESH bits are set to 0x3F, which is the reset value, the early transmission feature is disabled, and transmission does not start until the NEWTX bit is set in the **MACTR** register.

Writing the THRESH bits to any value besides 0x3F enables the early transmission feature. Once the byte count of data in the TX FIFO reaches the value derived from the THRESH bits as shown below, transmission of the frame begins. When THRESH is set to all 0s, transmission of the frame begins after 4 bytes (a single write) are stored in the TX FIFO. Each increment of the THRESH bit field waits for an additional 32 bytes of data (eight writes) to be stored in the TX FIFO. Therefore, a value of 0x01 causes the transmitter to wait for 36 bytes of data to be written while a value of 0x02 makes the wait equal to 68 bytes of written data. In general, early transmission starts when:

```
Number of Bytes >= 4 (THRESH x 8 + 1)
```

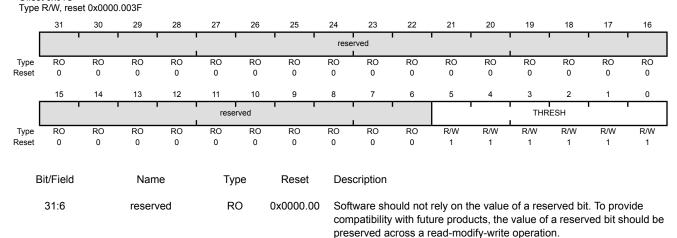
Reaching the threshold level has the same effect as setting the NEWTX bit in the **MACTR** register. Transmission of the frame begins and then the number of bytes indicated by the Data Length field is transmitted. Because under-run checking is not performed, if any event, such as an interrupt, delays the filling of the FIFO, the tail pointer may reach and pass the write pointer in the TX FIFO. In this event, indeterminate values are transmitted rather than the end of the frame. Therefore, sufficient bus bandwidth for writing to the TX FIFO must be guaranteed by the software.

If a frame smaller than the threshold level must be sent, the NEWTX bit in the **MACTR** register must be set with an explicit write. This initiates the transmission of the frame even though the threshold limit has not been reached.

If the threshold level is set too small, it is possible for the transmitter to underrun. If this occurs, the transmit frame is aborted, and a transmit error occurs. Note that in this case, the TXER bit in the MACRIS is not set meaning that the CPU receives no indication that a transmit error happened.

Ethernet MAC Threshold (MACTHR)

Base 0x4004.8000 Offset 0x01C



Bit/Field	Name	Туре	Reset	Description
5:0	THRESH	R/W	0x3F	Threshold Value The THRESH bits represent the early transmit threshold. Once the amount of data in the TX FIFO exceeds the value represented by the above equation, transmission of the packet begins.

Register 9: Ethernet MAC Management Control (MACMCTL), offset 0x020

This register enables software to control the transfer of data to and from the MII Management registers in the Ethernet PHY layer. The address, name, type, reset configuration, and functional description of each of these registers can be found in Table 14-4 on page 528 and in "MII Management Register Descriptions" on page 547.

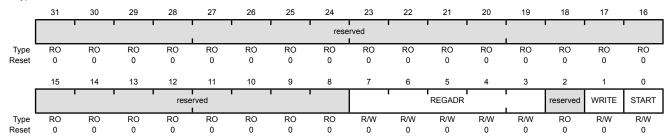
In order to initiate a read transaction from the MII Management registers, the WRITE bit must be cleared during the same cycle that the START bit is set.

In order to initiate a write transaction to the MII Management registers, the WRITE bit must be set during the same cycle that the START bit is set.

Ethernet MAC Management Control (MACMCTL)

Base 0x4004.8000

Offset 0x020 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:3	REGADR	R/W	0x0	MII Register Address
				The REGADR bit field represents the MII Management register address for the next MII management interface transaction. Refer to Table 14-4 on page 528 for the PHY register offsets.
				Note that any address that is not valid in the register map should not be written to and any data read should be ignored.
2	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	WRITE	R/W	0	MII Register Transaction Type
				The WRITE bit represents the operation of the next MII management interface transaction. If WRITE is set, the next operation is a write; if WRITE is clear, the next transaction is a read.
0	START	R/W	0	MII Register Transaction Enable
				The START bit represents the initiation of the next MII management interface transaction. When this bit is set, the MII register located at

REGADR is read (WRITE=0) or written (WRITE=1).

Register 10: Ethernet MAC Management Divider (MACMDV), offset 0x024

This register enables software to set the clock divider for the Management Data Clock (MDC). This clock is used to synchronize read and write transactions between the system and the MII Management registers. The frequency of the MDC clock can be calculated from the following formula:

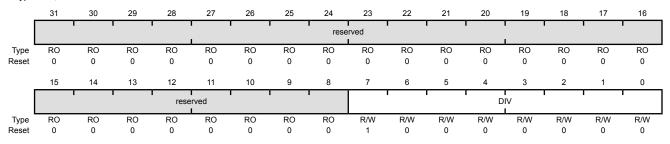
$$F_{mdc} = \frac{F_{ipclk}}{2 \times (MACDVR + 1)}$$

The clock divider must be written with a value that ensures that the MDC clock does not exceed a frequency of 2.5 MHz.

Ethernet MAC Management Divider (MACMDV)

Base 0x4004.8000 Offset 0x024

Type R/W, reset 0x0000.0080



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DIV	R/W	0x80	Clock Divider

The t DIV bits are used to set the clock divider for the MDC clock used to transmit data between the MAC and PHY layers over the serial MII interface.

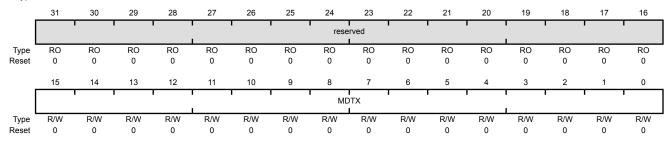
Register 11: Ethernet MAC Management Transmit Data (MACMTXD), offset 0x02C

This register holds the next value to be written to the MII Management registers.

Ethernet MAC Management Transmit Data (MACMTXD)

Base 0x4004.8000

Offset 0x02C Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	MDTX	R/W	0x0000	MII Register Transmit Data

The \mathtt{MDTX} bits represent the data that will be written in the next \mathbf{MII} management transaction.

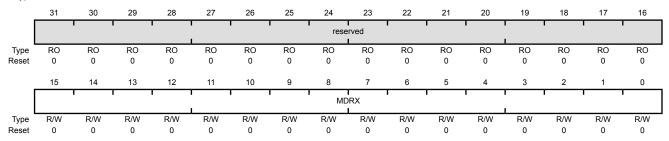
Register 12: Ethernet MAC Management Receive Data (MACMRXD), offset 0x030

This register holds the last value read from the MII Management registers.

Ethernet MAC Management Receive Data (MACMRXD)

Base 0x4004.8000

Offset 0x030 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:16	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	MDRX	R/W	0x0000	MII Register Receive Data

The ${\tt MDRX}$ bits represent the data that was read in the previous ${\tt MII}$ management transaction.

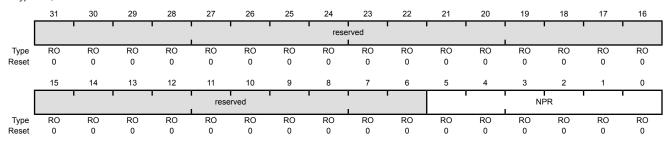
Register 13: Ethernet MAC Number of Packets (MACNP), offset 0x034

This register holds the number of frames that are currently in the RX FIFO. When NPR is 0, there are no frames in the RX FIFO, and the RXINT bit is clear. When NPR is any other value, at least one frame is in the RX FIFO, and the RXINT bit in the **MACRIS** register is set.

Note: The FCS bytes are not included in the NPR value. As a result, the NPR value could be zero before the FCS bytes are read from the FIFO. In addition, a new packet could be received before the NPR value reaches zero. To ensure that the entire packet is received, either use the DriverLib EthernetPacketGet() API or compare the number of bytes received to the Length field from the frame to determine when the packet has been completely read.

Ethernet MAC Number of Packets (MACNP)

Base 0x4004.8000 Offset 0x034 Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:6	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5:0	NPR	RO	0x00	Number of Packets in Receive FIFO

The NPR bits represent the number of packets stored in the RX FIFO. While the NPR field is greater than 0, the RXINT interrupt in the **MACRIS** register is set.

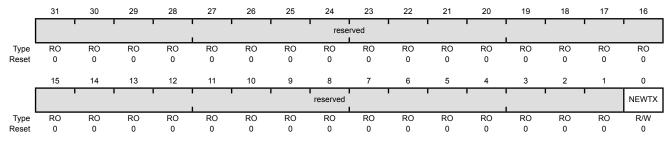
Register 14: Ethernet MAC Transmission Request (MACTR), offset 0x038

This register enables software to initiate the transmission of the frame currently located in the TX FIFO. Once the frame has been transmitted from the TX FIFO or a transmission error has been encountered, the NEWTX bit is automatically cleared.

Ethernet MAC Transmission Request (MACTR)

Base 0x4004.8000 Offset 0x038

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x0000.000	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	NEWTX	R/W	0	New Transmission

When set, the NEWTX bit initiates an Ethernet transmission once the packet has been placed in the TX FIFO. This bit is cleared once the transmission has been completed. If early transmission is being used (see the MACTHR register), this bit does not need to be set.

14.7 MII Management Register Descriptions

The *IEEE 802.3 standard* specifies a register set for controlling and gathering status from the PHY layer. The registers are collectively known as the MII Management registers. All addresses given are absolute. Addresses not listed are reserved; these addresses should not be written to and any data read should be ignored. Also see "Ethernet MAC Register Descriptions" on page 529.

Register 15: Ethernet PHY Management Register 0 – Control (MR0), address 0x00

This register enables software to configure the operation of the PHY layer. The default settings of these registers are designed to initialize the Ethernet Controller to a normal operational mode without configuration.

Ethernet PHY Management Register 0 – Control (MR0)

Base 0x4004.8000 Address 0x00 Type R/W, reset 0x3100

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESET	LOOPBK	SPEEDSL	ANEGEN	PWRDN	ISO	RANEG	DUPLEX	COLT	1			reserved		ı	
Type Reset	R/W 0	R/W 0	R/W 1	R/W 1	R/W 0	R/W 0	R/W 0	R/W 1	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0
E	Bit/Field		Nan	ne	Тур	oe	Reset	Desc	cription							
	15		RES	ET	R/\	W	0	Rese	et Regis	ters						
								and	reinitiali		nal state	machin	er register es. Once ware.			
	14		LOOF	PBK	R/\	W	0	Loop	back M	ode						
								igno					k mode o			
	13		SPEE	DSL	R/	W	1	Spee	ed Seled	ct						
								Valu	ie Desc	cription						
								1	Enat	oles the 1	00 Mb/s	mode o	of operation	n (100	BASE-TX	۲).
								0	Enat	oles the 1	0 Mb/s r	node of	operation	(10BA	SE-T).	
	12		ANEG	SEN	R/\	W	1	Auto	-Negotia	ation Ena	able					
								Whe	n set, th	nis bit ena	ables the	auto-ne	egotiation	proces	ss.	
	11		PWR	DN	R/\	V	0	Pow	er Dowr	1						
										nis bit pla a on the			er into a le nored.	ow-pov	er consu	uming
	10		ISC)	R/	W	0	Isola	ite							
									-				t and reco		ta paths	and
	9		RANI	EG	R/	V	0	Rest	art Auto	-Negotia	tion					
										is bit rest , this bit i		-	otiation p dware.	rocess.	Once th	e restart

Bit/Field	Name	Туре	Reset	Description			
8	DUPLEX	R/W	1	Set Duplex Mode			
				Value Description			
				Enables the Full-Duplex mode of operation. This bit can be set by software in a manual configuration process or by the auto-negotiation process.			
				0 Enables the Half-Duplex mode of operation.			
7	COLT	R/W	0	Collision Test			
				When set, this bit enables the Collision Test mode of operation. The ${\tt COLT}$ bit is set after the initiation of a transmission and is cleared once the transmission is halted.			
6:0	reserved	R/W	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. These bits should always be written as zero.			

Register 16: Ethernet PHY Management Register 1 – Status (MR1), address 0x01

This register enables software to determine the capabilities of the PHY layer and perform its initialization and operation appropriately.

Ethernet PHY Management Register 1 – Status (MR1)

10

13

Base 0x4004.8000 Address 0x01 Type RO, reset 0x7849

15

	13	14	13	12		10	3	0							'		
	reserved	100X_F	100X_H	10T_F	10T_H		reser	ved		MFPS	ANEGC	RFAULT	ANEGA	LINK	JAB	EXTD	
Type Reset	RO 0	RO 1	RO 1	RO 1	RO 1	RO 0	RO 0	RO 0	RO 0	RO 1	RO 0	RC 0	RO 1	RO 0	RC 0	RO 1	
					_		_	_									
E	Bit/Field		Nam	ne	Тур	oe	Reset	Des	cription								
	15		reserv	/ed	RO		0	con	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.								
	14		100X	_F	R	O	1	100	BASE-T	X Full-Dι	uplex Mo	ode					
									en set, th porting 1					Controlle	r is capa	ible of	
	13		100X	_H	R	O	1	100	BASE-T	X Half-D	uplex Mo	ode					
									en set, th porting 1					Controlle	r is capa	ible of	
	12		10T_	_F	R)	1	10B	ASE-T F	ull-Dupl	ex Mode						
									en set, th SASE-T F				thernet C	Controlle	r is capa	ible of	
	11		10T_	<u>_</u> H	R)	1	10B	ASE-T H	lalf-Dupl	ex Mode)					
									en set, th porting 1					Controlle	r is capa	ible of	
	10:7		reser	/ed	R)	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.									
	6		MFP	S	R)	1	Mar	nagemen	t Frame	s with Pr	eamble	Suppres	sed			
									en set, the				-			•	
	5		ANEC	3C	R)	0	Auto	o-Negotia	ation Co	mplete						
								com	en set, th opleted a o-negotia	nd that t	he exten	ided regi				as been	
	4		RFAU	JLT	R	С	0	Ren	note Fau	It							
								dete	en set, thected. The	is bit rer							
	3		ANE	GΑ	R)	1	Auto	o-Negotia	ation							
								Whe	en set, th	is bit inc	licates th	nat the E	thernet C	Controlle	r has the	ability	

to perform auto-negotiation.

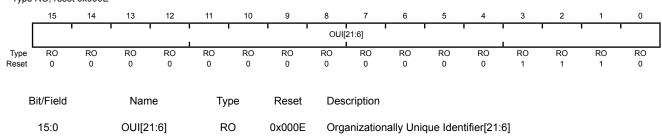
Bit/Field	Name	Туре	Reset	Description
2	LINK	RO	0	Link Made When set, this bit indicates that a valid link has been established by the Ethernet Controller.
1	JAB	RC	0	Jabber Condition When set, this bit indicates that a jabber condition has been detected by the Ethernet Controller. This bit remains set until it is read, even if the jabber condition no longer exists.
0	EXTD	RO	1	Extended Capabilities When set, this bit indicates that the Ethernet Controller provides an extended set of capabilities that can be accessed through the extended register set.

Register 17: Ethernet PHY Management Register 2 – PHY Identifier 1 (MR2), address 0x02

This register, along with **MR3**, provides a 32-bit value indicating the manufacturer, model, and revision information.

Ethernet PHY Management Register 2 – PHY Identifier 1 (MR2)

Base 0x4004.8000 Address 0x02 Type RO, reset 0x000E



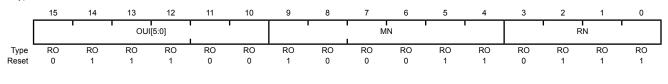
This field, along with the \mathtt{OUI} [5:0] field in **MR3**, makes up the Organizationally Unique Identifier indicating the PHY manufacturer.

Register 18: Ethernet PHY Management Register 3 – PHY Identifier 2 (MR3), address 0x03

This register, along with **MR2**, provides a 32-bit value indicating the manufacturer, model, and revision information.

Ethernet PHY Management Register 3 – PHY Identifier 2 (MR3)

Base 0x4004.8000 Address 0x03 Type RO, reset 0x7237



Bit/Field	Name	Туре	Reset	Description
15:10	OUI[5:0]	RO	0x1C	Organizationally Unique Identifier[5:0] This field, along with the OUI[21:6] field in MR2, makes up the Organizationally Unique Identifier indicating the PHY manufacturer.
9:4	MN	RO	0x23	Model Number The MN field represents the Model Number of the PHY.
3:0	RN	RO	0x7	Revision Number

The ${\tt RN}$ field represents the Revision Number of the PHY implementation.

Register 19: Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4), address 0x04

This register provides the advertised abilities of the Ethernet Controller used during auto-negotiation. Bits 8:5 represent the Technology Ability Field bits. This field can be overwritten by software to auto-negotiate to an alternate common technology. Writing to this register has no effect until auto-negotiation is re-initiated by setting the RANEG bit in the **MR0** register.

Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4)

Base 0x4004.8000 Address 0x04 Type R/W, reset 0x01E1

_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NP	reserved	RF		reser	ved		A3	A2	A1	A0		I I	S		<u>'</u>
Type Reset	RO 0	RO 0	R/W 0	RO 0	RO 0	RO 0	RO 0	R/W 1	R/W 1	R/W 1	R/W 1	RO 0	RO 0	RO 0	RO 0	RO 1
В	it/Field		Nan	ne	Туре		Reset	Des	Description							
	15		NF)	RO)	0	Nex	t Page							
							Pag					net Conti iled inforr		•		
	14	reserved		ved	RO)	0	Software should not rely on the value of a reserved bit. To compatibility with future products, the value of a reserved preserved across a read-modify-write operation.								
	13		RF		RΛ	N	0	Ren	note Fau	It						
								When set, this bit inc condition has been e					partner	that a Re	emote Fa	ault
	12:9		reserved		RO)	0x0	com	Software should not rely on the value of a compatibility with future products, the value preserved across a read-modify-write oper				value of	a reserv		
	8		A3	3	RΛ	N	1	Tech	nnology	Ability Fi	eld[3]					
								100 that	When set, this bit indicates that the Ethernet Controller supports the 100Base-TX full-duplex signaling protocol. If software wants to ensur that this mode is not used, this bit can be cleared and auto-negotiatio re-initiated with the RANEG bit in the MR0 register.							ensure
	7		A2	2	RΛ	N	1	Tech	nnology	Ability Fi	eld[2]					
								100 that	When set, this bit 100Base-TX half- that this mode is r re-initiated with the		olex sign used, th	aling pro	otocol. If s n be clea	software red and	wants to	ensure
	6		A1		RΛ	N	1	Tech	nnology	Ability Fi	eld[1]					
								10B that	ASE-T for	ull-duple: de is not	x signali used, th	ng proto is bit car	thernet (col. If soft be clea	ftware w red and	ants to e	ensure

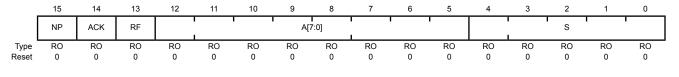
Bit/Field	Name	Туре	Reset	Description
5	A0	R/W	1	Technology Ability Field[0] When set, this bit indicates that the Ethernet Controller supports the 10BASE-T half-duplex signaling protocol. If software wants to ensure that this mode is not used, this bit can be cleared and auto-negotiation re-initiated with the RANEG bit in the MR0 register
4:0	S	RO	0x1	Selector Field The S field encodes 32 possible messages for communicating between Ethernet Controllers. This field is hard-coded to 0x01, indicating that the Stellaris Ethernet Controller is <i>IEEE 802.3</i> compliant.

Register 20: Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability (MR5), address 0x05

This register provides the advertised abilities of the link partner's Ethernet Controller that are received and stored during auto-negotiation.

Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability (MR5)

Base 0x4004.8000 Address 0x05 Type RO, reset 0x0000



Bit/Field	Name	Туре	Reset	Description
15	NP	RO	0	Next Page When set, this bit indicates that the link partner's Ethernet Controller is capable of Next page exchanges to provide more detailed information on the Ethernet Controller's capabilities.
14	ACK	RO	0	Acknowledge When set, this bit indicates that the Ethernet Controller has successfully received the link partner's advertised abilities during auto-negotiation.
13	RF	RO	0	Remote Fault Used as a standard transport mechanism for transmitting simple fault information from the link partner.
12:5	A[7:0]	RO	0x00	Technology Ability Field The A[7:0] field encodes individual technologies that are supported by the Ethernet Controller. See the MR4 register for definitions. Note that bits 12:9 describe functions that are not implemented on the Stellaris Ethernet Controller. Refer to the IEEE 802.3 standard for definitions.
4:0	S	RO	0x00	Selector Field

The $\ensuremath{\mathtt{S}}$ field encodes possible messages for communicating between Ethernet Controllers.

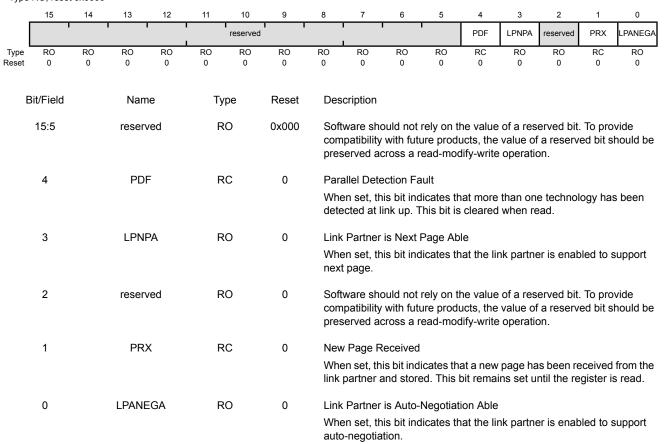
Value	Description
0x00	Reserved
0x01	IEEE Std 802.3
0x02	IEEE Std 802.9 ISLAN-16T
0x03	IEEE Std 802.5
0x04	IEEE Std 1394
0x05-0x1F	Reserved

Register 21: Ethernet PHY Management Register 6 – Auto-Negotiation Expansion (MR6), address 0x06

This register enables software to determine the auto-negotiation and next page capabilities of the Ethernet Controller and the link partner after auto-negotiation.

Ethernet PHY Management Register 6 - Auto-Negotiation Expansion (MR6)

Base 0x4004.8000 Address 0x06 Type RO, reset 0x0000



Register 22: Ethernet PHY Management Register 16 – Vendor-Specific (MR16), address 0x10

This register enables software to configure the operation of vendor-specific modes of the Ethernet Controller.

Ethernet PHY Management Register 16 – Vendor-Specific (MR16)

Base 0x4004.8000 Address 0x10 Type R/W, reset 0x0140

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RPTR	INPOL	reserved	TXHIM	SQEI	NL10		reser	ved		APOL	RVSPOL	rese	rved	PCSBP	RXCC
Type Reset	R/W 0	R/W0 0	RO 0	R/W 0	R/W 0	R/W 0	RO 0	RO 1	RO 0	RO 1	R/W 0	R/W 0	RO 0	RO 0	R/W 0	R/W 0
E	Bit/Field		Nam	ie	Ту	pe	Reset	Desc	ription							
	15		RPT	R	R/	W	0	Repe	eater Mo	ode						
								full-d		not allow		repeater I the Carr		•		
	14		INPO	DL	R/\	N0	0	Interi	rupt Pol	arity						
								Valu	e Desc	ription						
								1	Sets	the pola	rity of th	e PHY in	terrupt to	be acti	ve High.	
								0	Sets	the pola	rity of th	e PHY in	terrupt to	o active	Low.	
								lmp	ortan	Low ii	nterrupts	Media Ac s from the 0 to ensu	PHY, th	nis bit m	ust alway	
	13		reserv	/ed	R	0	0	comp	atibility	with futu	ire prod	he value ucts, the dify-write	value of	a reserv		
	12		TXHI	M	R/	W	0	Trans	smit Hig	h Imped	ance Mo	ode				
								mode	e, the TX	KOP and T	rxon tra	transmitt nsmitter p ins remai	ins are p	out into a	high imp	
	11		SQE	ΞI	R/	W	0	SQE	Inhibit '	Testing						
								Whe	n set, th	is bit pro	hibits 10	DBASE-T	SQE te	sting.		
												s perform ne transm				on pulse
	10		NL1	0	R/	W	0	Natu	ral Loop	back Mo	ode					
								this r	node, thed	ne transn	nission o	e 10BASI data recei data pat	ved by t	he Ethe	rnet Con	troller is
	9:6		reserv	/ed	R	0	0x5	comp	oatibility	with futu	ıre prod	he value ucts, the dify-write	value of	a reserv	•	

Bit/Field	Name	Туре	Reset	Description
5	APOL	R/W	0	Auto-Polarity Disable When set, this bit disables the Ethernet Controller's auto-polarity function. If this bit is clear, the Ethernet Controller automatically inverts the received signal due to a wrong polarity connection during auto-negotiation when in 10BASE-T mode.
4	RVSPOL	R/W	0	Receive Data Polarity This bit indicates whether the receive data pulses are being inverted. If the APOL bit is 0, then the RVSPOL bit is read-only and indicates whether the auto-polarity circuitry is reversing the polarity. In this case, if RVSPOL is set, it indicates that the receive data is inverted; if RVSPOL is clear, it indicates that the receive data is not inverted. If the APOL bit is 1, then the RVSPOL bit is writable and software can force the receive data to be inverted. Setting RVSPOL to 1 forces the receive data to be inverted; clearing RVSPOL does not invert the receive data.
3:2	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PCSBP	R/W	0	PCS Bypass When set, this bit enables the bypass of the PCS and scrambling/descrambling functions in 100BASE-TX mode. This mode is only valid when auto-negotiation is disabled and 100BASE-TX mode is enabled.
0	RXCC	R/W	0	Receive Clock Control When set, this bit enables the Receive Clock Control power saving mode if the Ethernet Controller is configured in 100BASE-TX mode. This mode shuts down the receive clock when no data is being received to save power. This mode should not be used when PCSBP is enabled and is automatically disabled when the LOOPBK bit in the MR0 register is set.

Register 23: Ethernet PHY Management Register 17 – Interrupt Control/Status (MR17), address 0x11

This register provides the means for controlling and observing the events which trigger a PHY layer interrupt in the **MACRIS** register. This register can also be used in a polling mode via the Media Independent Interface as a means to observe key events within the PHY layer via one register address. Bits 0 through 7 are status bits which are each set based on an event. These bits are cleared after the register is read. Bits 8 through 15 of this register, when set, enable the corresponding bit in the lower byte to signal a PHY layer interrupt in the **MACRIS** register.

Ethernet PHY Management Register 17 – Interrupt Control/Status (MR17)

Base 0x4004.8000 Address 0x11 Type R/W, reset 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	JABBER_IE	RXER_IE	PRX_IE	PDF_IE	LPACK_IE	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT	RXER_INT	PRX_INT	PDF_INT	LPACK_INT	LSCHG_INT	RFAULT_INT	ANEGCOMP_INT
Туре	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RC	RC	RC	RC	RC	RC	RC	RC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
15	JABBER_IE	R/W	0	Jabber Interrupt Enable When set, this bit enables system interrupts when a Jabber condition is detected by the Ethernet Controller.
14	RXER_IE	R/W	0	Receive Error Interrupt Enable When set, this bit enables system interrupts when a receive error is detected by the Ethernet Controller.
13	PRX_IE	R/W	0	Page Received Interrupt Enable When set, this bit enables system interrupts when a new page is received by the Ethernet Controller.
12	PDF_IE	R/W	0	Parallel Detection Fault Interrupt Enable When set, this bit enables system interrupts when a Parallel Detection Fault is detected by the Ethernet Controller.
11	LPACK_IE	R/W	0	LP Acknowledge Interrupt Enable When set, this bit enables system interrupts when FLP bursts are received with the ACK bit in the MR5 register during auto-negotiation.
10	LSCHG_IE	R/W	0	Link Status Change Interrupt Enable When set, this bit enables system interrupts when the link status changes from OK to FAIL.
9	RFAULT_IE	R/W	0	Remote Fault Interrupt Enable When set, this bit enables system interrupts when a remote fault condition is signaled by the link partner.
8	ANEGCOMP_IE	R/W	0	Auto-Negotiation Complete Interrupt Enable When set, this bit enables system interrupts when the auto-negotiation sequence has completed successfully.

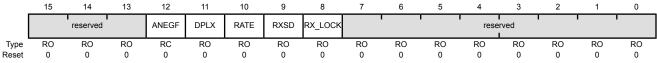
Bit/Field	Name	Туре	Reset	Description
7	JABBER_INT	RC	0	Jabber Event Interrupt When set, this bit indicates that a Jabber event has been detected by the 10BASE-T circuitry.
6	RXER_INT	RC	0	Receive Error Interrupt When set, this bit indicates that a receive error has been detected by the Ethernet Controller.
5	PRX_INT	RC	0	Page Receive Interrupt When set, this bit indicates that a new page has been received from the link partner during auto-negotiation.
4	PDF_INT	RC	0	Parallel Detection Fault Interrupt When set, this bit indicates that a parallel detection fault has been detected by the Ethernet Controller during the auto-negotiation process.
3	LPACK_INT	RC	0	LP Acknowledge Interrupt When set, this bit indicates that an FLP burst has been received with the ACK bit set in the MR5 register during auto-negotiation.
2	LSCHG_INT	RC	0	Link Status Change Interrupt When set, this bit indicates that the link status has changed from OK to FAIL.
1	RFAULT_INT	RC	0	Remote Fault Interrupt When set, this bit indicates that a remote fault condition has been signaled by the link partner.
0	ANEGCOMP_INT	RC	0	Auto-Negotiation Complete Interrupt When set, this bit indicates that the auto-negotiation sequence has completed successfully.

Register 24: Ethernet PHY Management Register 18 – Diagnostic (MR18), address 0x12

This register enables software to diagnose the results of the previous auto-negotiation.

Ethernet PHY Management Register 18 – Diagnostic (MR18)

Base 0x4004.8000 Address 0x12 Type RO, reset 0x0000



Bit/Field	Name	Туре	Reset	Description
15:13	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12	ANEGF	RC	0	Auto-Negotiation Failure When set, this bit indicates that no common technology was found during auto-negotiation and auto-negotiation has failed. This bit remains set until read.
11	DPLX	RO	0	Duplex Mode When set, this bit indicates that Full-Duplex was the highest common denominator found during the auto-negotiation process. Otherwise, Half-Duplex was the highest common denominator found.
10	RATE	RO	0	Rate When set, this bit indicates that 100BASE-TX was the highest common denominator found during the auto-negotiation process. Otherwise, 10BASE-T was the highest common denominator found.
9	RXSD	RO	0	Receive Detection When set, this bit indicates that receive signal detection has occurred (in 100BASE-TX mode) or that Manchester-encoded data has been detected (in 10BASE-T mode).
8	RX_LOCK	RO	0	Receive PLL Lock When set, this bit indicates that the Receive PLL has locked onto the receive signal for the selected speed of operation (10BASE-T or 100BASE-TX).
7:0	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 25: Ethernet PHY Management Register 19 – Transceiver Control (MR19), address 0x13

This register enables software to set the gain of the transmit output to compensate for transformer loss.

Ethernet PHY Management Register 19 - Transceiver Control (MR19)

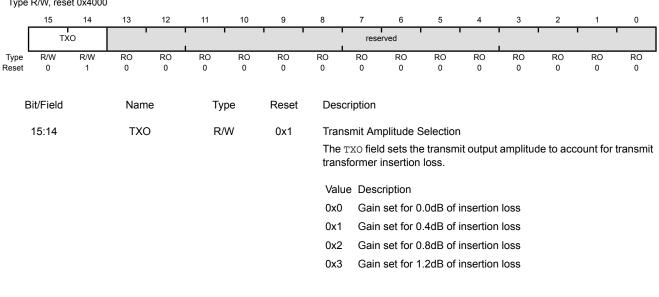
RO

reserved

0x000

Base 0x4004.8000 Address 0x13 Type R/W, reset 0x4000

13:0



Software should not rely on the value of a reserved bit. To provide

preserved across a read-modify-write operation.

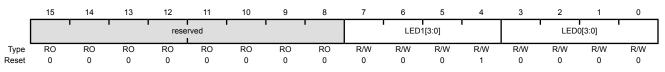
compatibility with future products, the value of a reserved bit should be

Register 26: Ethernet PHY Management Register 23 – LED Configuration (MR23), address 0x17

This register enables software to select the source that causes the LED1 and LED0 signals to toggle.

Ethernet PHY Management Register 23 – LED Configuration (MR23)

Base 0x4004.8000 Address 0x17 Type R/W, reset 0x0010



Bit/Field	Name	Туре	Reset	Description
15:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:4	LED1[3:0]	R/W	0x1	LED1 Source The LED1 field selects the source that toggles the LED1 signal. Value Description 0x0 Link OK

0x1 RX or TX Activity (Default LED1)
0x2 Reserved
0x3 Reserved
0x4 Reserved
0x5 100BASE-TX mode
0x6 10BASE-T mode

0x7 Full-Duplex 0x8 Link OK & Blink=RX or TX Activity

3:0 LED0[3:0] R/W 0x0 LED0 Source

The LED0 field selects the source that toggles the LED0 signal.

Value Description

0x0 Link OK (Default LED0)

0x1 RX or TX Activity

0x2 Reserved

0x3 Reserved

0x4 Reserved

0x5 100BASE-TX mode

0x6 10BASE-T mode

0x7 Full-Duplex

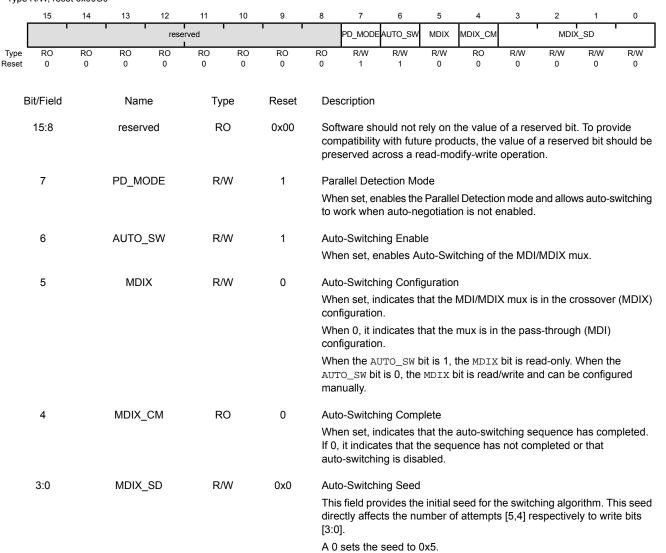
0x8 Link OK & Blink=RX or TX Activity

Register 27: Ethernet PHY Management Register 24 – MDI/MDIX Control (MR24), address 0x18

This register enables software to control the behavior of the MDI/MDIX mux and its switching capabilities.

Ethernet PHY Management Register 24 - MDI/MDIX Control (MR24)

Base 0x4004.8000 Address 0x18 Type R/W, reset 0x00C0



15 Analog Comparators

An analog comparator is a peripheral that compares two analog voltages, and provides a logical output that signals the comparison result.

Note: Not all comparators have the option to drive an output pin.

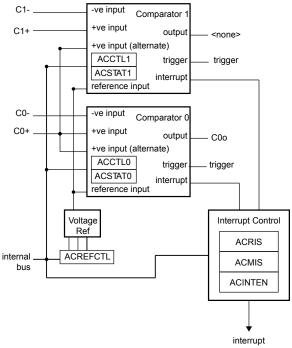
The comparator can provide its output to a device pin, acting as a replacement for an analog comparator on the board, or it can be used to signal the application via interrupts or triggers to the ADC to cause it to start capturing a sample sequence. The interrupt generation and ADC triggering logic is separate. This means, for example, that an interrupt can be generated on a rising edge and the ADC triggered on a falling edge.

The Stellaris[®] Analog Comparators module has the following features:

- Two independent integrated analog comparators
- Configurable for output to drive an output pin, generate an interrupt, or initiate an ADC sample sequence
- Compare external pin input to external pin input or to internal programmable voltage reference
- Compare a test voltage against any one of these voltages
 - An individual external reference voltage
 - A shared single external reference voltage
 - A shared internal reference voltage

15.1 Block Diagram

Figure 15-1. Analog Comparator Module Block Diagram



15.2 Signal Description

Table 15-1 on page 567 and Table 15-2 on page 567 list the external signals of the Analog Comparators and describe the function of each. The Analog Comparator output signals are alternate functions for some GPIO signals and default to be GPIO signals at reset. The column in the table below titled "Pin Assignment" lists the possible GPIO pin placements for the Analog Comparator signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the Analog Comparator function. The positive and negative input signals are configured by clearing the DEN bit in the **GPIO Digital Enable (GPIODEN)** register. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOS)" on page 258.

Table 15-1. Analog Comparators Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
C0+	90	I	Analog	Analog comparator 0 positive input.
C0-	92	I	Analog	Analog comparator 0 negative input.
C0o	100	0	TTL	Analog comparator 0 output.
C1+	24	I	Analog	Analog comparator 1 positive input.
C1-	91	I	Analog	Analog comparator 1 negative input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 15-2. Analog Comparators Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
C0+	A7	I	Analog	Analog comparator 0 positive input.

Table 15-2. Analog Comparators Signals (108BGA) (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
C0-	A6	1	Analog	Analog comparator 0 negative input.
COo	F1	0	TTL	Analog comparator 0 output.
C1+	M1	I	Analog	Analog comparator 1 positive input.
C1-	В7	1	Analog	Analog comparator 1 negative input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

15.3 Functional Description

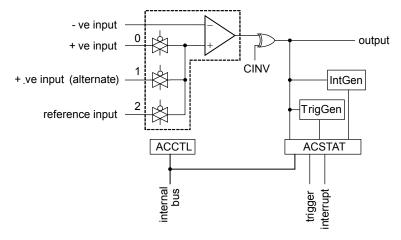
Important: It is recommended that the Digital-Input enable (the GPIODEN bit in the GPIO module) for the analog input pin be disabled to prevent excessive current draw from the I/O pads.

The comparator compares the VIN- and VIN+ inputs to produce an output, VOUT.

```
VIN- < VIN+, VOUT = 1
VIN- > VIN+, VOUT = 0
```

As shown in Figure 15-2 on page 568, the input source for VIN- is an external input. In addition to an external input, input sources for VIN+ can be the +ve input of comparator 0 or an internal reference.

Figure 15-2. Structure of Comparator Unit



A comparator is configured through two status/control registers (ACCTL and ACSTAT). The internal reference is configured through one control register (ACREFCTL). Interrupt status and control is configured through three registers (ACMIS, ACRIS, and ACINTEN).

Typically, the comparator output is used internally to generate controller interrupts. It may also be used to drive an external pin or generate an analog-to-digital converter (ADC) trigger.

Important: The ASRCP bits in the **ACCTLn** register must be set before using the analog comparators.

15.3.1 Internal Reference Programming

The structure of the internal reference is shown in Figure 15-3 on page 569. This is controlled by a single configuration register (**ACREFCTL**). Table 15-3 on page 569 shows the programming options

to develop specific internal reference values, to compare an external voltage against a particular voltage generated internally.

Figure 15-3. Comparator Internal Reference Structure

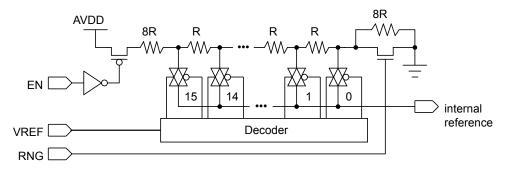


Table 15-3. Internal Reference Voltage and ACREFCTL Field Values

ACREFCTL Register		Output Reference Voltage Based on VREF Field Value		
EN Bit Value	RNG Bit Value	Output Reference voltage based on VREF Field Value		
EN=0	RNG=X	0 V (GND) for any value of VREF; however, it is recommended that RNG=1 and VREF=0 for the least noisy ground reference.		
	RNG=0	Total resistance in ladder is 31 R. $V_{REF} = AV_{DD} \times \frac{Rv_{REF}}{R_T}$ $V_{REF} = AV_{DD} \times \frac{(VREF + 8)}{31}$ $V_{REF} = 0.85 + 0.106 \times VREF$ The range of internal reference in this mode is 0.85-2.448 V.		
EN=1	RNG=1	Total resistance in ladder is 23 R. $V_{\it REF} = AV_{\it DD} \times \frac{Rv_{\it REF}}{Rr}$ $V_{\it REF} = AV_{\it DD} \times \frac{VREF}{23}$ $V_{\it REF} = 0.143 \times VREF$ The range of internal reference for this mode is 0-2.152 V.		

15.4 Initialization and Configuration

The following example shows how to configure an analog comparator to read back its output value from an internal register.

- **1.** Enable the analog comparator 0 clock by writing a value of 0x0010.0000 to the **RCGC1** register in the System Control module.
- 2. In the GPIO module, enable the GPIO port/pin associated with co- as a GPIO input.

- **3.** Configure the internal voltage reference to 1.65 V by writing the **ACREFCTL** register with the value 0x0000.030C.
- **4.** Configure comparator 0 to use the internal voltage reference and to *not* invert the output by writing the **ACCTL0** register with the value of 0x0000.040C.
- 5. Delay for some time.
- 6. Read the comparator output value by reading the ACSTAT0 register's OVAL value.

Change the level of the signal input on CO- to see the OVAL value change.

15.5 Register Map

Table 15-4 on page 570 lists the comparator registers. The offset listed is a hexadecimal increment to the register's address, relative to the Analog Comparator base address of 0x4003.C000.

Note that the analog comparator module clock must be enabled before the registers can be programmed (see page 215). There must be a delay of 3 system clocks after the ADC module clock is enabled before any ADC module registers are accessed.

Table 15-4. Analog Comparators Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	ACMIS	R/W1C	0x0000.0000	Analog Comparator Masked Interrupt Status	571
0x004	ACRIS	RO	0x0000.0000	Analog Comparator Raw Interrupt Status	572
0x008	ACINTEN	R/W	0x0000.0000	Analog Comparator Interrupt Enable	573
0x010	ACREFCTL	R/W	0x0000.0000	Analog Comparator Reference Voltage Control	574
0x020	ACSTAT0	RO	0x0000.0000	Analog Comparator Status 0	575
0x024	ACCTL0	R/W	0x0000.0000	Analog Comparator Control 0	576
0x040	ACSTAT1	RO	0x0000.0000	Analog Comparator Status 1	575
0x044	ACCTL1	R/W	0x0000.0000	Analog Comparator Control 1	576

15.6 Register Descriptions

The remainder of this section lists and describes the Analog Comparator registers, in numerical order by address offset.

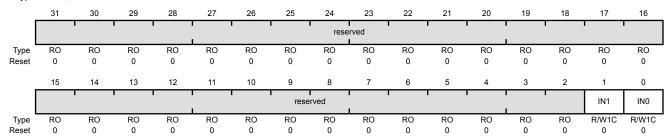
Register 1: Analog Comparator Masked Interrupt Status (ACMIS), offset 0x000

This register provides a summary of the interrupt status (masked) of the comparators.

Analog Comparator Masked Interrupt Status (ACMIS)

Base 0x4003.C000 Offset 0x000

Offset 0x000 Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	IN1	R/W1C	0	Comparator 1 Masked Interrupt Status
				Gives the masked interrupt state of this interrupt. Write 1 to this bit to clear the pending interrupt.
0	IN0	R/W1C	0	Comparator 0 Masked Interrupt Status
				Gives the masked interrupt state of this interrupt. Write 1 to this bit to clear the pending interrupt.

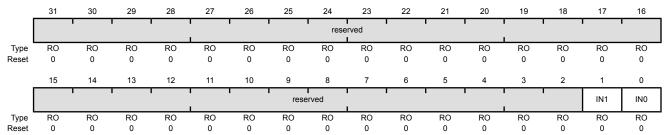
Register 2: Analog Comparator Raw Interrupt Status (ACRIS), offset 0x004

This register provides a summary of the interrupt status (raw) of the comparators.

Analog Comparator Raw Interrupt Status (ACRIS)

Base 0x4003.C000 Offset 0x004

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	IN1	RO	0	Comparator 1 Interrupt Status When set, indicates that an interrupt has been generated by comparator 1.
0	IN0	RO	0	Comparator 0 Interrupt Status When set, indicates that an interrupt has been generated by comparator 0.

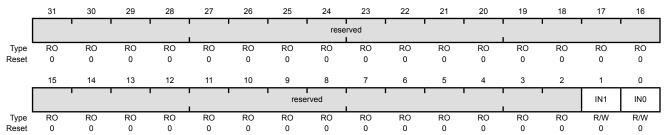
Register 3: Analog Comparator Interrupt Enable (ACINTEN), offset 0x008

This register provides the interrupt enable for the comparators.

Analog Comparator Interrupt Enable (ACINTEN)

Base 0x4003.C000

Offset 0x008 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	IN1	R/W	0	Comparator 1 Interrupt Enable When set, enables the controller interrupt from the comparator 1 output.
0	IN0	R/W	0	Comparator 0 Interrupt Enable When set, enables the controller interrupt from the comparator 0 output.

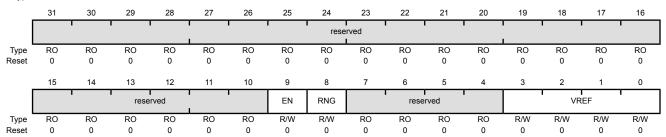
Register 4: Analog Comparator Reference Voltage Control (ACREFCTL), offset 0x010

This register specifies whether the resistor ladder is powered on as well as the range and tap.

Analog Comparator Reference Voltage Control (ACREFCTL)

Base 0x4003.C000

Offset 0x010 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:10	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
9	EN	R/W	0	Resistor Ladder Enable
				The $\tt EN$ bit specifies whether the resistor ladder is powered on. If 0, the resistor ladder is unpowered. If 1, the resistor ladder is connected to the analog $V_{\tt DD}.$
				This bit is reset to 0 so that the internal reference consumes the least amount of power if not used and programmed.
8	RNG	R/W	0	Resistor Ladder Range
				The RNG bit specifies the range of the resistor ladder. If 0, the resistor ladder has a total resistance of 31 R. If 1, the resistor ladder has a total resistance of 23 R.
7:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3:0	VREF	R/W	0x00	Resistor Ladder Voltage Ref

The VREF bit field specifies the resistor ladder tap that is passed through an analog multiplexer. The voltage corresponding to the tap position is the internal reference voltage available for comparison. See Table 15-3 on page 569 for some output reference voltage examples.

Register 5: Analog Comparator Status 0 (ACSTAT0), offset 0x020 Register 6: Analog Comparator Status 1 (ACSTAT1), offset 0x040

These registers specify the current output value of the comparator.

Analog Comparator Status 0 (ACSTAT0)

Base 0x4003.C000 Offset 0x020 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	OVAL	RO	0	Comparator Output Value The OVAL bit specifies the current output value of the comparator.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 7: Analog Comparator Control 0 (ACCTL0), offset 0x024 Register 8: Analog Comparator Control 1 (ACCTL1), offset 0x044

These registers configure the comparator's input and output.

Type

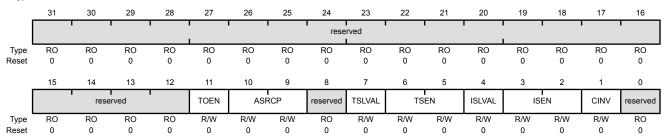
Reset

Analog Comparator Control 0 (ACCTL0)

Name

Base 0x4003.C000 Offset 0x024 Type R/W, reset 0x0000.0000

Bit/Field



Description

31:12	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11	TOEN	R/W	0	Trigger Output Enable The TOEN bit enables the ADC event transmission to the ADC. If 0, the event is suppressed and not sent to the ADC. If 1, the event is transmitted to the ADC.
10:9	ASRCP	R/W	0x00	Analog Source Positive The ASRCP field specifies the source of input voltage to the VIN+ terminal of the comparator. The encodings for this field are as follows: Value Function 0x0 Pin value 0x1 Pin value of C0+ 0x2 Internal voltage reference 0x3 Reserved
8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7	TSLVAL	R/W	0	Trigger Sense Level Value

The TSLVAL bit specifies the sense value of the input that generates an ADC event if in Level Sense mode. If 0, an ADC event is generated if the comparator output is Low. Otherwise, an ADC event is generated

if the comparator output is High.

Bit/Field	Name	Туре	Reset	Description
6:5	TSEN	R/W	0x0	Trigger Sense The TSEN field specifies the sense of the comparator output that generates an ADC event. The sense conditioning is as follows:
				Value Function
				0x0 Level sense, see TSLVAL
				0x1 Falling edge
				0x2 Rising edge
				0x3 Either edge
4	ISLVAL	R/W	0	Interrupt Sense Level Value
				The ISLVAL bit specifies the sense value of the input that generates an interrupt if in Level Sense mode. If 0, an interrupt is generated if the comparator output is Low. Otherwise, an interrupt is generated if the comparator output is High.
3:2	ISEN	R/W	0x0	Interrupt Sense
				The ISEN field specifies the sense of the comparator output that generates an interrupt. The sense conditioning is as follows:
				Value Function
				0x0 Level sense, see ISLVAL
				0x1 Falling edge
				0x2 Rising edge
				0x3 Either edge
1	CINV	R/W	0	Comparator Output Invert
				The ${\tt CINV}$ bit conditionally inverts the output of the comparator. If 0, the output of the comparator is unchanged. If 1, the output of the comparator is inverted prior to being processed by hardware.
0	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

16 Pulse Width Modulator (PWM)

Pulse width modulation (PWM) is a powerful technique for digitally encoding analog signal levels. High-resolution counters are used to generate a square wave, and the duty cycle of the square wave is modulated to encode an analog signal. Typical applications include switching power supplies and motor control.

The Stellaris[®] PWM module consists of one PWM generator block and a control block. The control block determines the polarity of the PWM signals, and which signals are passed through to the pins.

The PWM generator block produces two PWM signals that can either be independent signals (other than being based on the same timer and therefore having the same frequency) or a single pair of complementary signals with dead-band delays inserted. The output of the PWM generation block is managed by the output control block before being passed to the device pins.

The Stellaris PWM module provides a great deal of flexibility. It can generate simple PWM signals, such as those required by a simple charge pump. It can also generate paired PWM signals with dead-band delays, such as those required by a half-H bridge driver.

The Stellaris PWM module has the following features:

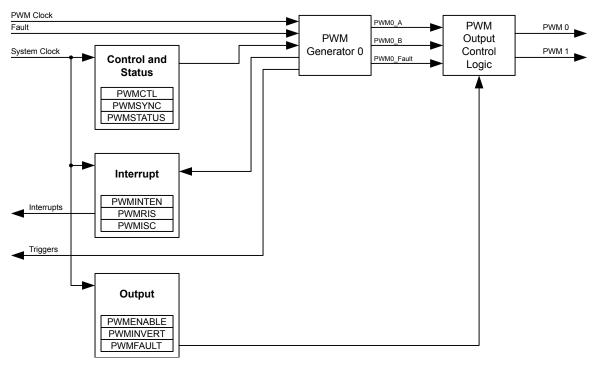
- One PWM generator blocks, each with one 16-bit counter, two PWM comparators, a PWM signal generator, a dead-band generator, and an interrupt/ADC-trigger selector
- One fault input in hardware to promote low-latency shutdown
- One 16-bit counter
 - Runs in Down or Up/Down mode
 - Output frequency controlled by a 16-bit load value
 - Load value updates can be synchronized
 - Produces output signals at zero and load value
- Two PWM comparators
 - Comparator value updates can be synchronized
 - Produces output signals on match
- PWM generator
 - Output PWM signal is constructed based on actions taken as a result of the counter and PWM comparator output signals
 - Produces two independent PWM signals
- Dead-band generator
 - Produces two PWM signals with programmable dead-band delays suitable for driving a half-H bridge
 - Can be bypassed, leaving input PWM signals unmodified

- Flexible output control block with PWM output enable of each PWM signal
 - PWM output enable of each PWM signal
 - Optional output inversion of each PWM signal (polarity control)
 - Optional fault handling for each PWM signal
 - Synchronization of timers in the PWM generator blocks
 - Interrupt status summary of the PWM generator blocks
- Can initiate an ADC sample sequence

16.1 Block Diagram

Figure 16-1 on page 579 provides the Stellaris PWM module unit diagram and Figure 16-2 on page 580 provides a more detailed diagram of a Stellaris PWM generator. The LM3S6432 controller contains one generator block (PWM0) and generates two independent PWM signals or one paired PWM signal with dead-band delays inserted.

Figure 16-1. PWM Unit Diagram



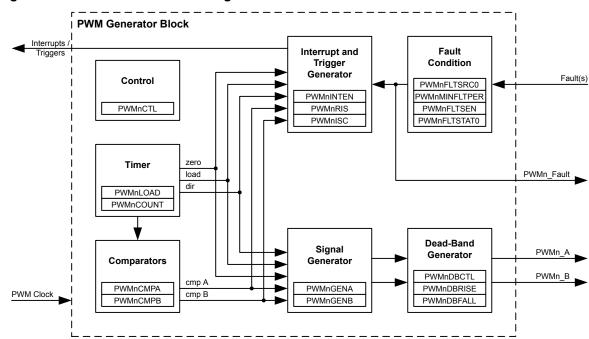


Figure 16-2. PWM Module Block Diagram

16.2 Signal Description

Table 16-1 on page 580 and Table 16-2 on page 580 list the external signals of the PWM module and describe the function of each. The PWM controller signals are alternate functions for some GPIO signals and default to be GPIO signals at reset. The column in the table below titled "Pin Assignment" lists the possible GPIO pin placements for these PWM signals. The AFSEL bit in the **GPIO Alternate Function Select (GPIOAFSEL)** register (page 280) should be set to choose the PWM function. For more information on configuring GPIOs, see "General-Purpose Input/Outputs (GPIOs)" on page 258.

Table 16-1. PWM Signals (100LQFP)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
Fault	99	I	TTL	PWM Fault.
PWM0	10	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
PWM1	11	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

Table 16-2. PWM Signals (108BGA)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
Fault	F2	1	TTL	PWM Fault.
PWM0	G1	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
PWM1	G2	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

16.3 Functional Description

16.3.1 **PWM Timer**

The timer runs in one of two modes: Count-Down mode or Count-Up/Down mode. In Count-Down mode, the timer counts from the load value to zero, goes back to the load value, and continues counting down. In Count-Up/Down mode, the timer counts from zero up to the load value, back down to zero, back up to the load value, and so on. Generally, Count-Down mode is used for generating left- or right-aligned PWM signals, while the Count-Up/Down mode is used for generating center-aligned PWM signals.

The timers output three signals that are used in the PWM generation process: the direction signal (this is always Low in Count-Down mode, but alternates between Low and High in Count-Up/Down mode), a single-clock-cycle-width High pulse when the counter is zero, and a single-clock-cycle-width High pulse when the counter is equal to the load value. Note that in Count-Down mode, the zero pulse is immediately followed by the load pulse.

16.3.2 PWM Comparators

There are two comparators in the PWM generator that monitor the value of the counter; when either match the counter, they output a single-clock-cycle-width High pulse. When in Count-Up/Down mode, these comparators match both when counting up and when counting down; they are therefore qualified by the counter direction signal. These qualified pulses are used in the PWM generation process. If either comparator match value is greater than the counter load value, then that comparator never outputs a High pulse.

Figure 16-3 on page 581 shows the behavior of the counter and the relationship of these pulses when the counter is in Count-Down mode. Figure 16-4 on page 582 shows the behavior of the counter and the relationship of these pulses when the counter is in Count-Up/Down mode.

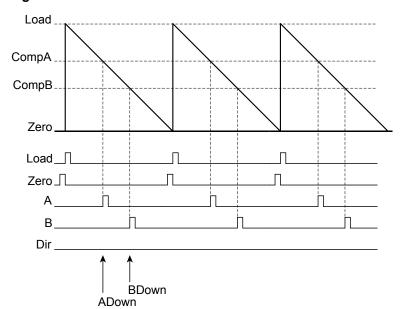


Figure 16-3. PWM Count-Down Mode

July 15, 2014 581

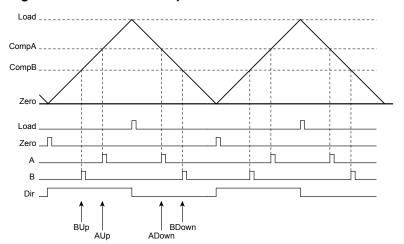


Figure 16-4. PWM Count-Up/Down Mode

16.3.3 PWM Signal Generator

The PWM generator takes these pulses (qualified by the direction signal), and generates two PWM signals. In Count-Down mode, there are four events that can affect the PWM signal: zero, load, match A down, and match B down. In Count-Up/Down mode, there are six events that can affect the PWM signal: zero, load, match A down, match A up, match B down, and match B up. The match A or match B events are ignored when they coincide with the zero or load events. If the match A and match B events coincide, the first signal, PWMA, is generated based only on the match A event, and the second signal, PWMB, is generated based only on the match B event.

For each event, the effect on each output PWM signal is programmable: it can be left alone (ignoring the event), it can be toggled, it can be driven Low, or it can be driven High. These actions can be used to generate a pair of PWM signals of various positions and duty cycles, which do or do not overlap. Figure 16-5 on page 582 shows the use of Count-Up/Down mode to generate a pair of center-aligned, overlapped PWM signals that have different duty cycles.

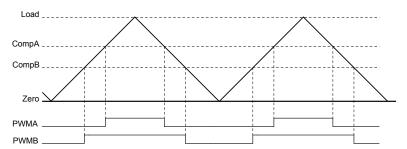


Figure 16-5. PWM Generation Example In Count-Up/Down Mode

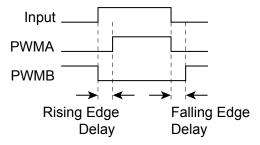
In this example, the first generator is set to drive High on match A up, drive Low on match A down, and ignore the other four events. The second generator is set to drive High on match B up, drive Low on match B down, and ignore the other four events. Changing the value of comparator A changes the duty cycle of the PWMA signal, and changing the value of comparator B changes the duty cycle of the PWMB signal.

16.3.4 Dead-Band Generator

The two PWM signals produced by the PWM generator are passed to the dead-band generator. If disabled, the PWM signals simply pass through unmodified. If enabled, the second PWM signal is lost and two PWM signals are generated based on the first PWM signal. The first output PWM signal is the input signal with the rising edge delayed by a programmable amount. The second output PWM signal is the inversion of the input signal with a programmable delay added between the falling edge of the input signal and the rising edge of this new signal.

This is therefore a pair of active High signals where one is always High, except for a programmable amount of time at transitions where both are Low. These signals are therefore suitable for driving a half-H bridge, with the dead-band delays preventing shoot-through current from damaging the power electronics. Figure 16-6 on page 583 shows the effect of the dead-band generator on an input PWM signal.

Figure 16-6. PWM Dead-Band Generator



16.3.5 Interrupt/ADC-Trigger Selector

The PWM generator also takes the same four (or six) counter events and uses them to generate an interrupt or an ADC trigger. Any of these events or a set of these events can be selected as a source for an interrupt; when any of the selected events occur, an interrupt is generated. Additionally, the same event, a different event, the same set of events, or a different set of events can be selected as a source for an ADC trigger; when any of these selected events occur, an ADC trigger pulse is generated. The selection of events allows the interrupt or ADC trigger to occur at a specific position within the PWM signal. Note that interrupts and ADC triggers are based on the raw events; delays in the PWM signal edges caused by the dead-band generator are not taken into account.

16.3.6 Synchronization Methods

There is a global reset capability that can reset the counter of the PWM generator.

The counter load values and comparator match values of the PWM generator can be updated in two ways. The first is immediate update mode, where a new value is used as soon as the counter reaches zero. By waiting for the counter to reach zero, a guaranteed behavior is defined, and overly short or overly long output PWM pulses are prevented.

The other update method is synchronous, where the new value is not used until a global synchronized update signal is asserted, at which point the new value is used as soon as the counter reaches zero. This second mode allows multiple items to be updated simultaneously without odd effects during the update; everything runs from the old values until a point at which they all run from the new values.

16.3.7 Fault Conditions

There are two external conditions that affect the PWM block; the signal input on the Fault pin and the stalling of the controller by a debugger. There are two mechanisms available to handle such

conditions: the output signals can be forced into an inactive state and/or the PWM timers can be stopped.

Each output signal has a fault bit. If set, a fault input signal causes the corresponding output signal to go into the inactive state. If the inactive state is a safe condition for the signal to be in for an extended period of time, this keeps the output signal from driving the outside world in a dangerous manner during the fault condition. A fault condition can also generate a controller interrupt.

Each PWM generator can also be configured to stop counting during a stall condition. The user can select for the counters to run until they reach zero then stop, or to continue counting and reloading. A stall condition does not generate a controller interrupt.

16.3.8 Output Control Block

With the PWM generator block producing two raw PWM signals, the output control block takes care of the final conditioning of the PWM signals before they go to the pins. Via a single register, the set of PWM signals that are actually enabled to the pins can be modified; this can be used, for example, to perform commutation of a brushless DC motor with a single register write (and without modifying the individual PWM generators, which are modified by the feedback control loop). Similarly, fault control can disable any of the PWM signals as well. A final inversion can be applied to any of the PWM signals, making them active Low instead of the default active High.

16.4 Initialization and Configuration

The following example shows how to initialize the PWM Generator 0 with a 25-KHz frequency, and with a 25% duty cycle on the PWM0 pin and a 75% duty cycle on the PWM1 pin. This example assumes the system clock is 20 MHz.

- **1.** Enable the PWM clock by writing a value of 0x0010.0000 to the **RCGC0** register in the System Control module.
- Enable the clock to the appropriate GPIO module via the RCGC2 register in the System Control module.
- 3. In the GPIO module, enable the appropriate pins for their alternate function using the GPIOAFSEL register.
- **4.** Configure the **Run-Mode Clock Configuration (RCC)** register in the System Control module to use the PWM divide (USEPWMDIV) and set the divider (PWMDIV) to divide by 2 (000).
- 5. Configure the PWM generator for countdown mode with immediate updates to the parameters.
 - Write the **PWM0CTL** register with a value of 0x0000.0000.
 - Write the **PWM0GENA** register with a value of 0x0000.008C.
 - Write the **PWM0GENB** register with a value of 0x0000.080C.
- **6.** Set the period. For a 25-KHz frequency, the period = 1/25,000, or 40 microseconds. The PWM clock source is 10 MHz; the system clock divided by 2. This translates to 400 clock ticks per period. Use this value to set the **PWM0LOAD** register. In Count-Down mode, set the Load field in the **PWM0LOAD** register to the requested period minus one.
 - Write the PWM0LOAD register with a value of 0x0000.018F.
- 7. Set the pulse width of the PWM0 pin for a 25% duty cycle.

- Write the **PWM0CMPA** register with a value of 0x0000.012B.
- 8. Set the pulse width of the PWM1 pin for a 75% duty cycle.
 - Write the **PWM0CMPB** register with a value of 0x0000.0063.
- 9. Start the timers in PWM generator 0.
 - Write the **PWM0CTL** register with a value of 0x0000.0001.
- 10. Enable PWM outputs.
 - Write the **PWMENABLE** register with a value of 0x0000.0003.

16.5 Register Map

Table 16-3 on page 585 lists the PWM registers. The offset listed is a hexadecimal increment to the register's address, relative to the PWM base address of 0x4002.8000. Note that the PWM module clock must be enabled before the registers can be programmed (see page 209). There must be a delay of 3 system clocks after the PWM module clock is enabled before any PWM module registers are accessed.

Table 16-3. PWM Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	PWMCTL	R/W	0x0000.0000	PWM Master Control	587
0x004	PWMSYNC	R/W	0x0000.0000	PWM Time Base Sync	588
800x0	PWMENABLE	R/W	0x0000.0000	PWM Output Enable	589
0x00C	PWMINVERT	R/W	0x0000.0000	PWM Output Inversion	590
0x010	PWMFAULT	R/W	0x0000.0000	PWM Output Fault	591
0x014	PWMINTEN	R/W	0x0000.0000	PWM Interrupt Enable	592
0x018	PWMRIS	RO	0x0000.0000	PWM Raw Interrupt Status	593
0x01C	PWMISC	R/W1C	0x0000.0000	PWM Interrupt Status and Clear	594
0x020	PWMSTATUS	RO	0x0000.0000	PWM Status	595
0x040	PWM0CTL	R/W	0x0000.0000	PWM0 Control	596
0x044	PWM0INTEN	R/W	0x0000.0000	PWM0 Interrupt and Trigger Enable	598
0x048	PWM0RIS	RO	0x0000.0000	PWM0 Raw Interrupt Status	601
0x04C	PWM0ISC	R/W1C	0x0000.0000	PWM0 Interrupt Status and Clear	602
0x050	PWM0LOAD	R/W	0x0000.0000	PWM0 Load	603
0x054	PWM0COUNT	RO	0x0000.0000	PWM0 Counter	604
0x058	PWM0CMPA	R/W	0x0000.0000	PWM0 Compare A	605
0x05C	PWM0CMPB	R/W	0x0000.0000	PWM0 Compare B	606
0x060	PWM0GENA	R/W	0x0000.0000	PWM0 Generator A Control	607

Table 16-3. PWM Register Map (continued)

Offset	Name	Туре	Reset	Description	See page
0x064	PWM0GENB	R/W	0x0000.0000	PWM0 Generator B Control	610
0x068	PWM0DBCTL	R/W	0x0000.0000	PWM0 Dead-Band Control	613
0x06C	PWM0DBRISE	R/W	0x0000.0000	PWM0 Dead-Band Rising-Edge Delay	614
0x070	PWM0DBFALL	R/W	0x0000.0000	PWM0 Dead-Band Falling-Edge-Delay	615

16.6 Register Descriptions

The remainder of this section lists and describes the PWM registers, in numerical order by address offset.

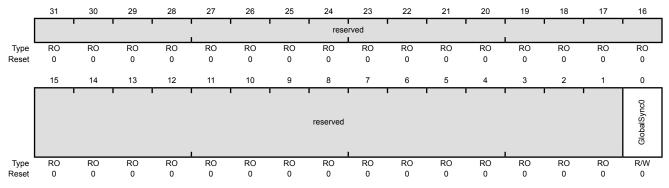
Register 1: PWM Master Control (PWMCTL), offset 0x000

This register provides master control over the PWM generation block.

PWM Master Control (PWMCTL)

Base 0x4002.8000 Offset 0x000

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	GlobalSync0	R/W	0	Update PWM Generator 0

Setting this bit causes any queued update to a load or comparator register in PWM generator 0 to be applied the next time the corresponding counter becomes zero. This bit automatically clears when the updates have completed; it cannot be cleared by software.

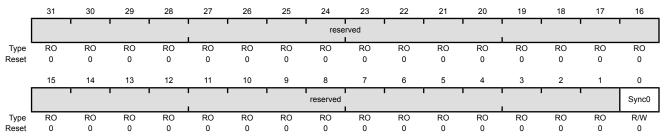
Register 2: PWM Time Base Sync (PWMSYNC), offset 0x004

This register provides a method to perform synchronization of the counters in the PWM generation blocks. Writing a bit in this register to 1 causes the specified counter to reset back to 0; writing multiple bits resets multiple counters simultaneously. The bits auto-clear after the reset has occurred; reading them back as zero indicates that the synchronization has completed.

PWM Time Base Sync (PWMSYNC)

Base 0x4002.8000

Offset 0x004 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	Sync0	R/W	0	Reset Generator 0 Counter

Performs a reset of the PWM generator 0 counter.

Register 3: PWM Output Enable (PWMENABLE), offset 0x008

This register provides a master control of which generated PWM signals are output to device pins. By disabling a PWM output, the generation process can continue (for example, when the time bases are synchronized) without driving PWM signals to the pins. When bits in this register are set, the corresponding PWM signal is passed through to the output stage, which is controlled by the **PWMINVERT** register. When bits are not set, the PWM signal is replaced by a zero value which is also passed to the output stage.

PWM Output Enable (PWMENABLE)

Base 0x4002.8000 Offset 0x008

Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1		1				rese	rved						1	
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		1		ı	 		rese	erved					 		PWM1En	PWM0En
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PWM1En	R/W	0	PWM1 Output Enable
				When set, allows the generated ${\tt PWM1}$ signal to be passed to the device pin.
0	PWM0En	R/W	0	PWM0 Output Enable
				When set, allows the generated ${\tt PWM0}$ signal to be passed to the device pin.

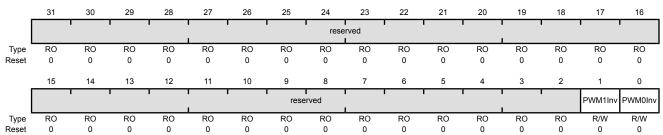
Register 4: PWM Output Inversion (PWMINVERT), offset 0x00C

This register provides a master control of the polarity of the PWM signals on the device pins. The PWM signals generated by the PWM generator are active High; they can optionally be made active Low via this register. Disabled PWM channels are also passed through the output inverter (if so configured) so that inactive channels maintain the correct polarity.

PWM Output Inversion (PWMINVERT)

Base 0x4002.8000

Offset 0x00C Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	PWM1Inv	R/W	0	Invert PWM1 Signal When set, the generated PWM1 signal is inverted.
0	PWM0Inv	R/W	0	Invert PWM0 Signal When set, the generated PWM0 signal is inverted.

Register 5: PWM Output Fault (PWMFAULT), offset 0x010

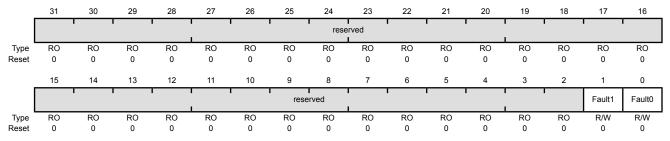
This register controls the behavior of the PWM outputs in the presence of fault conditions. Both the fault inputs and debug events are considered fault conditions. On a fault condition, each PWM signal can be passed through unmodified or driven Low. For outputs that are configured for pass-through, the debug event handling on the corresponding PWM generator also determines if the PWM signal continues to be generated.

Fault condition control occurs before the output inverter, so PWM signals driven Low on fault are inverted if the channel is configured for inversion (therefore, the pin is driven High on a fault condition).

PWM Output Fault (PWMFAULT)

Base 0x4002.8000

Offset 0x010 Type R/W, reset 0x0000.0000



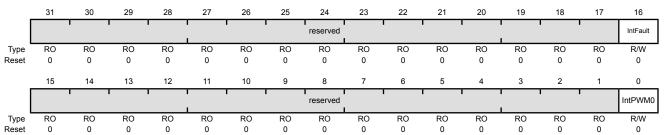
Bit/Field	Name	Type	Reset	Description
31:2	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
1	Fault1	R/W	0	PWM1 Fault When set, the PWM1 output signal is driven Low on a fault condition.
0	Fault0	R/W	0	PWM0 Fault When set, the PWM0 output signal is driven Low on a fault condition.

Register 6: PWM Interrupt Enable (PWMINTEN), offset 0x014

This register controls the global interrupt generation capabilities of the PWM module. The events that can cause an interrupt are the fault input and the individual interrupts from the PWM generator.

PWM Interrupt Enable (PWMINTEN)

Base 0x4002.8000 Offset 0x014 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:17	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16	IntFault	R/W	0	Fault Interrupt Enable When set, an interrupt occurs when the fault input is asserted.
15:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	IntPWM0	R/W	0	PWM0 Interrupt Enable When set, an interrupt occurs when the PWM generator 0 block asserts

an interrupt.

Register 7: PWM Raw Interrupt Status (PWMRIS), offset 0x018

This register provides the current set of interrupt sources that are asserted, regardless of whether they cause an interrupt to be asserted to the controller. The fault interrupt is latched on detection; it must be cleared through the **PWM Interrupt Status and Clear (PWMISC)** register (see page 594). The PWM generator interrupts simply reflect the status of the PWM generator; they are cleared via the interrupt status register in the PWM generator block. Bits set to 1 indicate the events that are active; zero bits indicate that the event in question is not active.

PWM Raw Interrupt Status (PWMRIS)

Base 0x4002.8000 Offset 0x018 Type RO, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	ı	1				reserved				1			1	IntFault
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		1	1	1				reserved							1	IntPWM0
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31:17	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16	IntFault	RO	0	Fault Interrupt Asserted Indicates that the fault input is asserting.
15:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	IntPWM0	RO	0	PWM0 Interrupt Asserted Indicates that the PWM generator 0 block is asserting its interrupt.

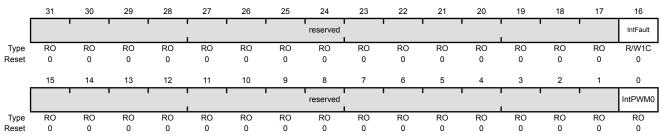
Register 8: PWM Interrupt Status and Clear (PWMISC), offset 0x01C

This register provides a summary of the interrupt status of the PWM generator block. A bit set to 1 indicates that the generator block is asserting an interrupt. The individual interrupt status registers must be consulted to determine the reason for the interrupt, and used to clear the interrupt. For the fault interrupt, a write of 1 to that bit position clears the latched interrupt status.

PWM Interrupt Status and Clear (PWMISC)

Base 0x4002.8000

Offset 0x01C Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:17	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
16	IntFault	R/W1C	0	Fault Interrupt Asserted Indicates that the fault input is asserting an interrupt.
15:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	IntPWM0	RO	0	PWM0 Interrupt Status Indicates if the PWM generator 0 block is asserting an interrupt.

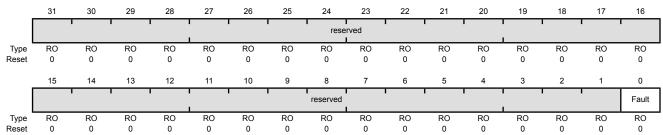
Register 9: PWM Status (PWMSTATUS), offset 0x020

This register provides the status of the ${\tt FAULT}$ input signal.

PWM Status (PWMSTATUS)

Base 0x4002.8000 Offset 0x020

Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	Fault	RO	0	Fault Interrupt Status When set, indicates the fault input is asserted.

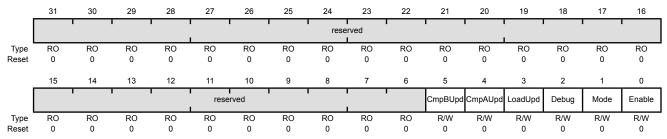
Register 10: PWM0 Control (PWM0CTL), offset 0x040

This register configures the PWM signal generation block. The Register Update mode, Debug mode, Counting mode, and Block Enable mode are all controlled via this register. The block produces the PWM signals, which can be either two independent PWM signals (from the same counter), or a paired set of PWM signals with dead-band delays added.

The PWM0 block produces the PWM0 and PWM1 outputs.

PWM0 Control (PWM0CTL)

Base 0x4002.8000 Offset 0x040 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	CmpBUpd	R/W	0	Comparator B Update Mode
				Same as ${\tt CmpAUpd}$ but for the comparator B register.
4	CmpAUpd	R/W	0	Comparator A Update Mode
				The Update mode for the comparator A register. When not set, updates to the register are reflected to the comparator the next time the counter is 0. When set, updates to the register are delayed until the next time the counter is 0 after a synchronous update has been requested through the PWM Master Control (PWMCTL) register (see page 587).
3	LoadUpd	R/W	0	Load Register Update Mode
				The Update mode for the load register. When not set, updates to the register are reflected to the counter the next time the counter is 0. When set, updates to the register are delayed until the next time the counter is 0 after a synchronous update has been requested through the PWM Master Control (PWMCTL) register.
2	Debug	R/W	0	Debug Mode
				The behavior of the counter in Debug mode. When not set, the counter stops running when it next reaches 0, and continues running again when no longer in Debug mode. When set, the counter always runs.
1	Mode	R/W	0	Counter Mode

The mode for the counter. When not set, the counter counts down from the load value to 0 and then wraps back to the load value (Count-Down mode). When set, the counter counts up from 0 to the load value, back

down to 0, and then repeats (Count-Up/Down mode).

E	Bit/Field	Name	Туре	Reset	Description
	0	Enable	R/W	0	PWM Block Enable Master enable for the PWM generation block. When not set, the entire block is disabled and not clocked. When set, the block is enabled and produces PWM signals.

Register 11: PWM0 Interrupt and Trigger Enable (PWM0INTEN), offset 0x044

This register controls the interrupt and ADC trigger generation capabilities of the PWM generator. The events that can cause an interrupt or an ADC trigger are:

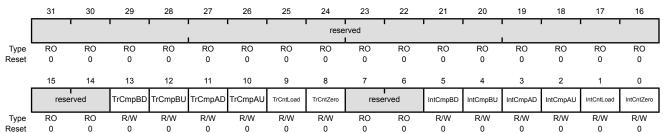
- The counter being equal to the load register
- The counter being equal to zero
- The counter being equal to the comparator A register while counting up
- The counter being equal to the comparator A register while counting down
- The counter being equal to the comparator B register while counting up
- The counter being equal to the comparator B register while counting down

Any combination of these events can generate either an interrupt, or an ADC trigger; though no determination can be made as to the actual event that caused an ADC trigger if more than one is specified.

PWM0 Interrupt and Trigger Enable (PWM0INTEN)

Base 0x4002.8000 Offset 0x044

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:14	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13	TrCmpBD	R/W	0	Trigger for Counter=Comparator B Down
				Value Description
				An ADC trigger pulse is output when the counter matches the value in the PWMnCMPB register value while counting down.
				0 No ADC trigger is output.
12	TrCmpBU	R/W	0	Trigger for Counter=Comparator B Up
				Value Description

- 1 An ADC trigger pulse is output when the counter matches the value in the **PWMnCMPB** register value while counting up.
- 0 No ADC trigger is output.

Bit/Field	Name	Туре	Reset	Description
11	TrCmpAD	R/W	0	Trigger for Counter=Comparator A Down
				Value Description
				An ADC trigger pulse is output when the counter matches the value in the PWMnCMPA register value while counting down.
				0 No ADC trigger is output.
10	TrCmpAU	R/W	0	Trigger for Counter=Comparator A Up
				Value Description
				An ADC trigger pulse is output when the counter matches the value in the PWMnCMPA register value while counting up.
				0 No ADC trigger is output.
9	TrCntLoad	R/W	0	Trigger for Counter=Load
				Value Description
				1 An ADC trigger pulse is output when the counter matches the PWMnLOAD register.
				0 No ADC trigger is output.
8	TrCntZero	R/W	0	Trigger for Counter=0
				Value Description
				1 An ADC trigger pulse is output when the counter is 0.
				0 No ADC trigger is output.
7:6	reserved	RO	0x0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	IntCmpBD	R/W	0	Interrupt for Counter=Comparator B Down
				Value Description
				A raw interrupt occurs when the counter matches the value in the PWMnCMPB register value while counting down.
				0 No interrupt.
4	IntCmpBU	R/W	0	Interrupt for Counter=Comparator B Up
				Value Description
				A raw interrupt occurs when the counter matches the value in the PWMnCMPB register value while counting up.
				0 No interrupt.

July 15, 2014 599

Bit/Field	Name	Туре	Reset	Description
3	IntCmpAD	R/W	0	Interrupt for Counter=Comparator A Down
				Value Description
				A raw interrupt occurs when the counter matches the value in the PWMnCMPA register value while counting down.
				0 No interrupt.
2	IntCmpAU	R/W	0	Interrupt for Counter=Comparator A Up
				Value Description
				A raw interrupt occurs when the counter matches the value in the PWMnCMPA register value while counting up.
				0 No interrupt.
1	IntCntLoad	R/W	0	Interrupt for Counter=Load
				Value Description
				A raw interrupt occurs when the counter matches the value in the PWMnLOAD register value.
				0 No interrupt.
0	IntCntZero	R/W	0	Interrupt for Counter=0
				Value Description
				1 A raw interrupt occurs when the counter is zero.
				0 No interrupt.

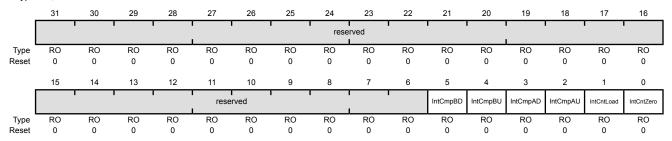
Register 12: PWM0 Raw Interrupt Status (PWM0RIS), offset 0x048

This register provides the current set of interrupt sources that are asserted, regardless of whether they cause an interrupt to be asserted to the controller. Bits set to 1 indicate the latched events that have occurred; bits set to 0 indicate that the event in question has not occurred.

PWM0 Raw Interrupt Status (PWM0RIS)

Base 0x4002.8000

Offset 0x048 Type RO, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	IntCmpBD	RO	0	Comparator B Down Interrupt Status Indicates that the counter has matched the comparator B value while counting down.
4	IntCmpBU	RO	0	Comparator B Up Interrupt Status Indicates that the counter has matched the comparator B value while counting up.
3	IntCmpAD	RO	0	Comparator A Down Interrupt Status Indicates that the counter has matched the comparator A value while counting down.
2	IntCmpAU	RO	0	Comparator A Up Interrupt Status Indicates that the counter has matched the comparator A value while counting up.
1	IntCntLoad	RO	0	Counter=Load Interrupt Status Indicates that the counter has matched the PWMnLOAD register.
0	IntCntZero	RO	0	Counter=0 Interrupt Status Indicates that the counter has matched 0.

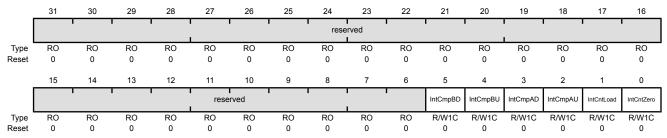
Register 13: PWM0 Interrupt Status and Clear (PWM0ISC), offset 0x04C

This register provides the current set of interrupt sources that are asserted to the controller. Bits set to 1 indicate the latched events that have occurred; bits set to 0 indicate that the event in question has not occurred. These are R/W1C registers; writing a 1 to a bit position clears the corresponding interrupt reason.

PWM0 Interrupt Status and Clear (PWM0ISC)

Base 0x4002.8000

Offset 0x04C Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:6	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	IntCmpBD	R/W1C	0	Comparator B Down Interrupt Indicates that the counter has matched the comparator B value while counting down.
4	IntCmpBU	R/W1C	0	Comparator B Up Interrupt Indicates that the counter has matched the comparator B value while counting up.
3	IntCmpAD	R/W1C	0	Comparator A Down Interrupt Indicates that the counter has matched the comparator A value while counting down.
2	IntCmpAU	R/W1C	0	Comparator A Up Interrupt Indicates that the counter has matched the comparator A value while counting up.
1	IntCntLoad	R/W1C	0	Counter=Load Interrupt Indicates that the counter has matched the PWMnLOAD register.
0	IntCntZero	R/W1C	0	Counter=0 Interrupt Indicates that the counter has matched 0.

Register 14: PWM0 Load (PWM0LOAD), offset 0x050

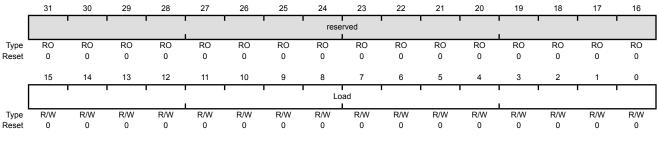
This register contains the load value for the PWM counter. Based on the counter mode, either this value is loaded into the counter after it reaches zero, or it is the limit of up-counting after which the counter decrements back to zero.

If the Load Value Update mode is immediate, this value is used the next time the counter reaches zero; if the mode is synchronous, it is used the next time the counter reaches zero after a synchronous update has been requested through the **PWM Master Control (PWMCTL)** register (see page 587). If this register is re-written before the actual update occurs, the previous value is never used and is lost.

PWM0 Load (PWM0LOAD)

Base 0x4002.8000 Offset 0x050

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	Load	R/W	0	Counter Load Value

The counter load value.

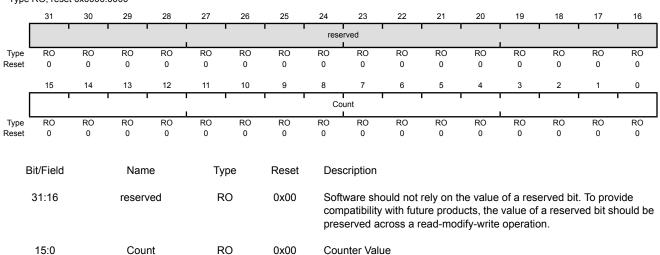
Register 15: PWM0 Counter (PWM0COUNT), offset 0x054

This register contains the current value of the PWM counter. When this value matches the load register, a pulse is output; this can drive the generation of a PWM signal (via the **PWMnGENA/PWMnGENB** registers, see page 607 and page 610) or drive an interrupt or ADC trigger (via the **PWMnINTEN** register, see page 598). A pulse with the same capabilities is generated when this value is zero.

PWM0 Counter (PWM0COUNT)

Base 0x4002.8000 Offset 0x054

Type RO, reset 0x0000.0000



The current value of the counter.

Register 16: PWM0 Compare A (PWM0CMPA), offset 0x058

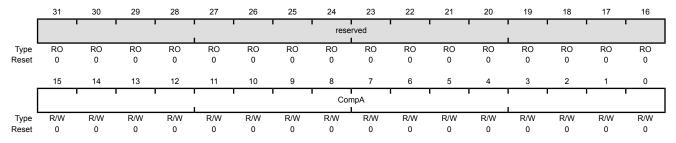
This register contains a value to be compared against the counter. When this value matches the counter, a pulse is output; this can drive the generation of a PWM signal (via the **PWMnGENA/PWMnGENB** registers) or drive an interrupt or ADC trigger (via the **PWMnINTEN** register). If the value of this register is greater than the **PWMnLOAD** register (see page 603), then no pulse is ever output.

If the comparator A update mode is immediate (based on the CmpAUpd bit in the **PWMnCTL** register), this 16-bit CompA value is used the next time the counter reaches zero. If the update mode is synchronous, it is used the next time the counter reaches zero after a synchronous update has been requested through the **PWM Master Control (PWMCTL)** register (see page 587). If this register is rewritten before the actual update occurs, the previous value is never used and is lost.

PWM0 Compare A (PWM0CMPA)

Base 0x4002.8000 Offset 0x058

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	CompA	R/W	0x00	Comparator A Value

The value to be compared against the counter.

Register 17: PWM0 Compare B (PWM0CMPB), offset 0x05C

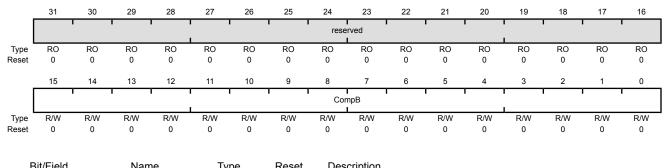
This register contains a value to be compared against the counter. When this value matches the counter, a pulse is output; this can drive the generation of a PWM signal (via the **PWMnGENA/PWMnGENB** registers) or drive an interrupt or ADC trigger (via the **PWMnINTEN** register). If the value of this register is greater than the **PWMnLOAD** register, no pulse is ever output.

If the comparator B update mode is immediate (based on the <code>CmpBUpd</code> bit in the <code>PWMnCTL</code> register), this 16-bit <code>CompB</code> value is used the next time the counter reaches zero. If the update mode is synchronous, it is used the next time the counter reaches zero after a synchronous update has been requested through the <code>PWM Master Control</code> (<code>PWMCTL</code>) register (see page 587). If this register is rewritten before the actual update occurs, the previous value is never used and is lost.

PWM0 Compare B (PWM0CMPB)

Base 0x4002.8000 Offset 0x05C

Type R/W, reset 0x0000.0000



DIVI ICIU	Name	Type	Neset	Description
31:16	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	CompB	R/W	0x00	Comparator B Value

The value to be compared against the counter.

Register 18: PWM0 Generator A Control (PWM0GENA), offset 0x060

This register controls the generation of the PWMnA signal based on the load and zero output pulses from the counter, as well as the compare A and compare B pulses from the comparators. When the counter is running in Count-Down mode, only four of these events occur; when running in Count-Up/Down mode, all six occur. These events provide great flexibility in the positioning and duty cycle of the PWM signal that is produced.

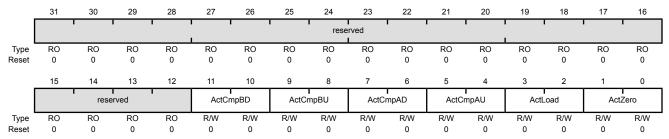
The **PWM0GENA** register controls generation of the PWM0A signal.

If a zero or load event coincides with a compare A or compare B event, the zero or load action is taken and the compare A or compare B action is ignored. If a compare A event coincides with a compare B event, the compare A action is taken and the compare B action is ignored.

PWM0 Generator A Control (PWM0GENA)

Base 0x4002.8000 Offset 0x060

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:12	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:10	ActCmpBD	R/W	0x0	Action for Comparator B Down

The action to be taken when the counter matches comparator B while counting down.

The table below defines the effect of the event on the output signal.

Value Description

0x0 Do nothing.

0x1 Invert the output signal.

0x2 Set the output signal to 0.

0x3 Set the output signal to 1.

Bit/Field	Name	Туре	Reset	Description
9:8	ActCmpBU	R/W	0x0	Action for Comparator B Up
	·			The action to be taken when the counter matches comparator B while counting up. Occurs only when the Mode bit in the PWMnCTL register (see page 596) is set to 1.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
7:6	ActCmpAD	R/W	0x0	Action for Comparator A Down
				The action to be taken when the counter matches comparator A while counting down.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
5:4	ActCmpAU	R/W	0x0	Action for Comparator A Up
				The action to be taken when the counter matches comparator A while counting up. Occurs only when the Mode bit in the PWMnCTL register is set to 1.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
3:2	ActLoad	R/W	0x0	Action for Counter=Load
				The action to be taken when the counter matches the load value.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.

Bit/Field	Name	Type	Reset	Description
1:0	ActZero	R/W	0x0	Action for Counter=0 The action to be taken when the counter is zero. The table below defines the effect of the event on the output signal. Value Description 0x0 Do nothing. 0x1 Invert the output signal. 0x2 Set the output signal to 0. 0x3 Set the output signal to 1.

Register 19: PWM0 Generator B Control (PWM0GENB), offset 0x064

This register controls the generation of the PWMnB signal based on the load and zero output pulses from the counter, as well as the compare A and compare B pulses from the comparators. When the counter is running in Down mode, only four of these events occur; when running in Up/Down mode, all six occur. These events provide great flexibility in the positioning and duty cycle of the PWM signal that is produced.

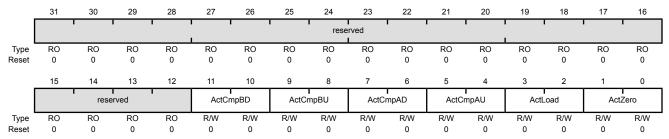
The **PWM0GENB** register controls generation of the PWM0B signal.

If a zero or load event coincides with a compare A or compare B event, the zero or load action is taken and the compare A or compare B action is ignored. If a compare A event coincides with a compare B event, the compare B action is taken and the compare A action is ignored.

PWM0 Generator B Control (PWM0GENB)

Base 0x4002.8000 Offset 0x064

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:12	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:10	ActCmpBD	R/W	0x0	Action for Comparator B Down

The action to be taken when the counter matches comparator B while counting down.

The table below defines the effect of the event on the output signal.

Value Description

0x0 Do nothing.

0x1 Invert the output signal.

0x2 Set the output signal to 0.

0x3 Set the output signal to 1.

Bit/Field	Name	Туре	Reset	Description
9:8	ActCmpBU	R/W	0x0	Action for Comparator B Up
	·			The action to be taken when the counter matches comparator B while counting up. Occurs only when the Mode bit in the PWMnCTL register is set to 1.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
7:6	ActCmpAD	R/W	0x0	Action for Comparator A Down
				The action to be taken when the counter matches comparator A while counting down.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
F. 4	A -+C A I I	DAM	0.40	Astion for Community A Ha
5:4	ActCmpAU	R/W	0x0	Action for Comparator A Up The action to be taken when the counter matches comparator A while
				counting up. Occurs only when the Mode bit in the PWMnCTL register is set to 1.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.
0.0	A -441	DAM	0-70	Advisor for Occuptor Local
3:2	ActLoad	R/W	0x0	Action for Counter=Load The action to be taken when the counter matches the load value.
				The table below defines the effect of the event on the output signal.
				Value Description
				0x0 Do nothing.
				0x1 Invert the output signal.
				0x2 Set the output signal to 0.
				0x3 Set the output signal to 1.

Bit/Field	Name	Type	Reset	Description
1:0	ActZero	R/W	0x0	Action for Counter=0 The action to be taken when the counter is 0. The table below defines the effect of the event on the output signal. Value Description 0x0 Do nothing. 0x1 Invert the output signal. 0x2 Set the output signal to 0. 0x3 Set the output signal to 1.
				one out the output digital to 1.

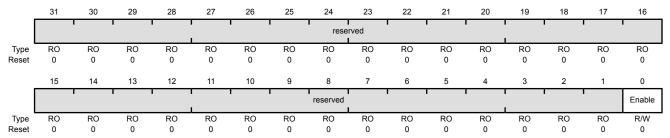
Register 20: PWM0 Dead-Band Control (PWM0DBCTL), offset 0x068

The **PWM0DBCTL** register controls the dead-band generator, which produces the PWM0 and PWM1 signals based on the PWM0A and PWM0B signals. When disabled, the PWM0A signal passes through to the PWM0 signal and the PWM0B signal passes through to the PWM1 signal. When enabled and inverting the resulting waveform, the PWM0B signal is ignored; the PWM0 signal is generated by delaying the rising edge(s) of the PWM0A signal by the value in the **PWM0DBRISE** register (see page 614), and the PWM1 signal is generated by delaying the falling edge(s) of the PWM0A signal by the value in the **PWM0DBFALL** register (see page 615).

PWM0 Dead-Band Control (PWM0DBCTL)

Base 0x4002.8000 Offset 0x068

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:1	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
0	Enable	R/W	0	Dead-Band Generator Enable

When set, the dead-band generator inserts dead bands into the output signals; when clear, it simply passes the PWM signals through.

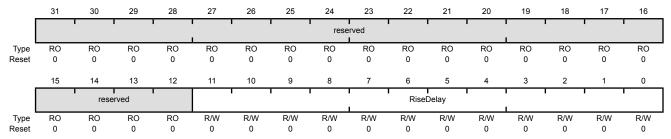
Register 21: PWM0 Dead-Band Rising-Edge Delay (PWM0DBRISE), offset 0x06C

The **PWM0DBRISE** register contains the number of clock ticks to delay the rising edge of the PWM0A signal when generating the PWM0 signal. If the dead-band generator is disabled through the **PWMnDBCTL** register, the **PWM0DBRISE** register is ignored. If the value of this register is larger than the width of a High pulse on the input PWM signal, the rising-edge delay consumes the entire High time of the signal, resulting in no High time on the output. Care must be taken to ensure that the input High time always exceeds the rising-edge delay.

PWM0 Dead-Band Rising-Edge Delay (PWM0DBRISE)

Base 0x4002.8000 Offset 0x06C

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	RiseDelay	R/W	0	Dead-Band Rise Delay

The number of clock ticks to delay the rising edge.

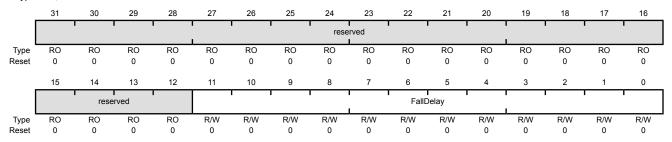
Register 22: PWM0 Dead-Band Falling-Edge-Delay (PWM0DBFALL), offset 0x070

The **PWM0DBFALL** register contains the number of clock ticks to delay the falling edge of the PWM0A signal when generating the PWM1 signal. If the dead-band generator is disabled, this register is ignored. If the value of this register is larger than the width of a Low pulse on the input PWM signal, the falling-edge delay consumes the entire Low time of the signal, resulting in no Low time on the output. Care must be taken to ensure that the input Low time always exceeds the falling-edge delay.

PWM0 Dead-Band Falling-Edge-Delay (PWM0DBFALL)

Base 0x4002.8000 Offset 0x070

Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:12	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
11:0	FallDelay	R/W	0x00	Dead-Band Fall Delay

The number of clock ticks to delay the falling edge.

17 Pin Diagram

The LM3S6432 microcontroller pin diagrams are shown below.

Figure 17-1. 100-Pin LQFP Package Pin Diagram

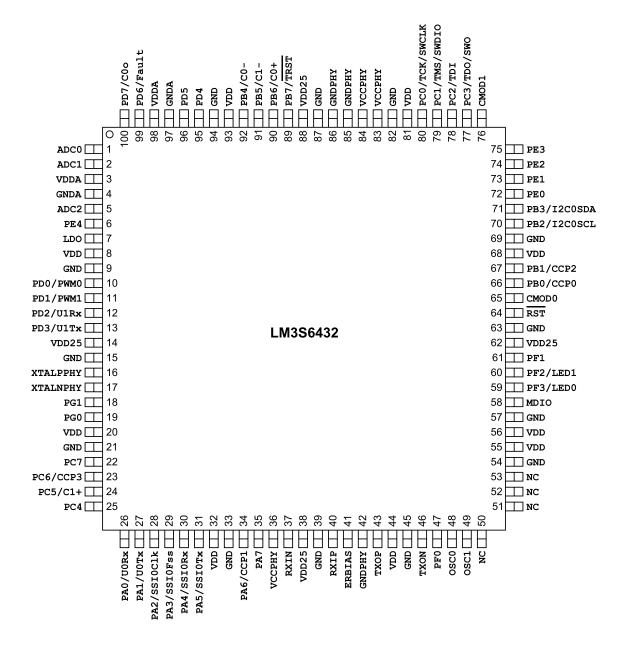


Figure 17-2. 108-Ball BGA Package Pin Diagram (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	
Α	ADC1	NC	NC NC	NC	GNDA	PB4 C0-	PB6 C0+	PB7 TRST	PC0 TCK SWCLK	PC3 TDO SWO	PEO	PE3	Α
В	ADC0	NC	ADC2	NC	GNDA	GND	PB5 C1-	PC2 TDI	PC1 TMS SWDIO	CMOD1	PE2	PE1	В
С	NC (NC	VDD25	GND	GND	VDDA	VDDA	GNDPHY	GNDPHY	VCCPHY	PB2 I2C0SCL	PB3 I2COSDA	С
D	PE4	NC	VDD25							VCCPHY	VCCPHY	PB1 CCP2	D
Е	PD4	PD5	TDO							VDD33	CMOD0	PB0 CCP0	E
F	PD7 C0o	PD6 Fault	VDD25							GND	GND	GND	F
G	PD0 PWM0	PD1 PWM1	VDD25			LM3	S6432			VDD33	VDD33	VDD33	G
Н	PD3 U1Tx	PD2 U1Rx	GND							VDD33	RST	PF1	Н
J	XTALNPHY	KTALPPHY	GND							GND	PF2 LED1	PF3 LED0	J
K	PG0	PG1	ERBIAS	GNDPHY	GND	GND	VDD33	VDD33	VDD33	GND	NC	NC	K
L	PC4	PC7	PA0 UORX	PA3 SSIOFss	PA4 SSIORX	PA6 CCP1	RXIN	TXON	MDIO	GND	OSC0	VDD	L
М	PC5 C1+	PC6 CCP3	PA1 UOTx	PA2 SSIOC1k	PA5 SSIOTX	PA7	RXIP	TXOP	PF0	NC	OSC1	NC	M
	1	2	3	4	5	6	7	8	9	10	11	12	

18 Signal Tables

Important: All multiplexed pins are GPIOs by default, with the exception of the five JTAG pins (PB7 and PC[3:0]) which default to the JTAG functionality.

The following tables list the signals available for each pin. Functionality is enabled by software with the **GPIOAFSEL** register. All digital inputs are Schmitt triggered.

- Signals by Pin Number
- Signals by Signal Name
- Signals by Function, Except for GPIO
- GPIO Pins and Alternate Functions
- Connections for Unused Signals

18.1 100-Pin LQFP Package Pin Tables

18.1.1 Signals by Pin Number

Table 18-1. Signals by Pin Number

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
1	ADC0	I	Analog	Analog-to-digital converter input 0.
2	ADC1	I	Analog	Analog-to-digital converter input 1.
3	VDDA	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
4	GNDA	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
5	ADC2	I	Analog	Analog-to-digital converter input 2.
6	PE4	I/O	TTL	GPIO port E bit 4.
7	LDO	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
8	VDD	-	Power	Positive supply for I/O and some logic.
9	GND	-	Power	Ground reference for logic and I/O pins.
10	PD0	I/O	TTL	GPIO port D bit 0.
10	PWM0	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
11	PD1	I/O	TTL	GPIO port D bit 1.
''	PWM1	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.
	PD2	I/O	TTL	GPIO port D bit 2.
12	U1Rx	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.

Table 18-1. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
	PD3	I/O	TTL	GPIO port D bit 3.
13	UlTx	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.
14	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
15	GND	-	Power	Ground reference for logic and I/O pins.
16	XTALPPHY	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.
17	XTALNPHY	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
18	PG1	I/O	TTL	GPIO port G bit 1.
19	PG0	I/O	TTL	GPIO port G bit 0.
20	VDD	-	Power	Positive supply for I/O and some logic.
21	GND	-	Power	Ground reference for logic and I/O pins.
22	PC7	I/O	TTL	GPIO port C bit 7.
	PC6	I/O	TTL	GPIO port C bit 6.
23	CCP3	I/O	TTL	Capture/Compare/PWM 3.
24	PC5	I/O	TTL	GPIO port C bit 5.
24	C1+	I	Analog	Analog comparator 1 positive input.
25	PC4	I/O	TTL	GPIO port C bit 4.
	PA0	I/O	TTL	GPIO port A bit 0.
26	U0Rx	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
	PA1	I/O	TTL	GPIO port A bit 1.
27	UOTx	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
20	PA2	I/O	TTL	GPIO port A bit 2.
28	SSI0Clk	I/O	TTL	SSI module 0 clock.
20	PA3	I/O	TTL	GPIO port A bit 3.
29	SSI0Fss	I/O	TTL	SSI module 0 frame signal.
20	PA4	I/O	TTL	GPIO port A bit 4.
30	SSI0Rx	I	TTL	SSI module 0 receive.
0.4	PA5	I/O	TTL	GPIO port A bit 5.
31	SSIOTx	0	TTL	SSI module 0 transmit.
32	VDD	-	Power	Positive supply for I/O and some logic.
33	GND	-	Power	Ground reference for logic and I/O pins.
0.4	PA6	I/O	TTL	GPIO port A bit 6.
34	CCP1	I/O	TTL	Capture/Compare/PWM 1.
35	PA7	I/O	TTL	GPIO port A bit 7.
36	VCCPHY	-	Power	VCC of the Ethernet PHY.
37	RXIN	1	Analog	RXIN of the Ethernet PHY.
38	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.

Table 18-1. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
39	GND	-	Power	Ground reference for logic and I/O pins.
40	RXIP	I	Analog	RXIP of the Ethernet PHY.
41	ERBIAS	ı	Analog	12.4-kΩ resistor (1% precision) used internally for Ethernet PHY.
42	GNDPHY	-	Power	GND of the Ethernet PHY.
43	TXOP	0	Analog	TXOP of the Ethernet PHY.
44	VDD	-	Power	Positive supply for I/O and some logic.
45	GND	-	Power	Ground reference for logic and I/O pins.
46	TXON	0	Analog	TXON of the Ethernet PHY.
47	PF0	I/O	TTL	GPIO port F bit 0.
48	OSC0	I	Analog	Main oscillator crystal input or an external clock reference input.
49	OSC1	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
50	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
51	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
52	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
53	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
54	GND	-	Power	Ground reference for logic and I/O pins.
55	VDD	-	Power	Positive supply for I/O and some logic.
56	VDD	-	Power	Positive supply for I/O and some logic.
57	GND	-	Power	Ground reference for logic and I/O pins.
58	MDIO	I/O	TTL	MDIO of the Ethernet PHY.
50	PF3	I/O	TTL	GPIO port F bit 3.
59	LED0	0	TTL	Ethernet LED 0.
60	PF2	I/O	TTL	GPIO port F bit 2.
60	LED1	0	TTL	Ethernet LED 1.
61	PF1	I/O	TTL	GPIO port F bit 1.
62	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
63	GND	-	Power	Ground reference for logic and I/O pins.
64	RST	I	TTL	System reset input.
65	CMOD0	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
66	PB0	I/O	TTL	GPIO port B bit 0.
	CCP0	I/O	TTL	Capture/Compare/PWM 0.
67	PB1	I/O	TTL	GPIO port B bit 1.
07	CCP2	I/O	TTL	Capture/Compare/PWM 2.
68	VDD	-	Power	Positive supply for I/O and some logic.
69	GND	-	Power	Ground reference for logic and I/O pins.
70	PB2	I/O	TTL	GPIO port B bit 2.
/0	I2C0SCL	I/O	OD	I ² C module 0 clock.
71 -	PB3	I/O	TTL	GPIO port B bit 3.
''	I2C0SDA	I/O	OD	I ² C module 0 data.

Table 18-1. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
72	PE0	I/O	TTL	GPIO port E bit 0.
73	PE1	I/O	TTL	GPIO port E bit 1.
74	PE2	I/O	TTL	GPIO port E bit 2.
75	PE3	I/O	TTL	GPIO port E bit 3.
76	CMOD1	1	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
	PC3	I/O	TTL	GPIO port C bit 3.
77	SWO	0	TTL	JTAG TDO and SWO.
	TDO	0	TTL	JTAG TDO and SWO.
78	PC2	I/O	TTL	GPIO port C bit 2.
/6	TDI	I	TTL	JTAG TDI.
	PC1	I/O	TTL	GPIO port C bit 1.
79	SWDIO	I/O	TTL	JTAG TMS and SWDIO.
	TMS	I/O	TTL	JTAG TMS and SWDIO.
	PC0	I/O	TTL	GPIO port C bit 0.
80	SWCLK	I	TTL	JTAG/SWD CLK.
	TCK	I	TTL	JTAG/SWD CLK.
81	VDD	-	Power	Positive supply for I/O and some logic.
82	GND	-	Power	Ground reference for logic and I/O pins.
83	VCCPHY	-	Power	VCC of the Ethernet PHY.
84	VCCPHY	-	Power	VCC of the Ethernet PHY.
85	GNDPHY	-	Power	GND of the Ethernet PHY.
86	GNDPHY	-	Power	GND of the Ethernet PHY.
87	GND	-	Power	Ground reference for logic and I/O pins.
88	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
89	PB7	I/O	TTL	GPIO port B bit 7.
09 –	TRST	I	TTL	JTAG TRST.
90	PB6	I/O	TTL	GPIO port B bit 6.
90 –	C0+	I	Analog	Analog comparator 0 positive input.
91	PB5	I/O	TTL	GPIO port B bit 5.
91 –	C1-	I	Analog	Analog comparator 1 negative input.
02	PB4	I/O	TTL	GPIO port B bit 4.
92	C0-	I	Analog	Analog comparator 0 negative input.
93	VDD	-	Power	Positive supply for I/O and some logic.
94	GND	-	Power	Ground reference for logic and I/O pins.
95	PD4	I/O	TTL	GPIO port D bit 4.
96	PD5	I/O	TTL	GPIO port D bit 5.
97	GNDA	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.

Table 18-1. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
98	VDDA	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
99	PD6	I/O	TTL	GPIO port D bit 6.
99	Fault	1	TTL	PWM Fault.
100	PD7	I/O	TTL	GPIO port D bit 7.
100	C0o	0	TTL	Analog comparator 0 output.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.1.2 Signals by Signal Name

Table 18-2. Signals by Signal Name

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ADC0	1	Į	Analog	Analog-to-digital converter input 0.
ADC1	2	Į	Analog	Analog-to-digital converter input 1.
ADC2	5	I	Analog	Analog-to-digital converter input 2.
C0+	90	Į	Analog	Analog comparator 0 positive input.
C0-	92	Į	Analog	Analog comparator 0 negative input.
C0o	100	0	TTL	Analog comparator 0 output.
C1+	24	Į	Analog	Analog comparator 1 positive input.
C1-	91	I	Analog	Analog comparator 1 negative input.
CCP0	66	I/O	TTL	Capture/Compare/PWM 0.
CCP1	34	I/O	TTL	Capture/Compare/PWM 1.
CCP2	67	I/O	TTL	Capture/Compare/PWM 2.
CCP3	23	I/O	TTL	Capture/Compare/PWM 3.
CMOD0	65	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	76	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
ERBIAS	41	I	Analog	12.4-k Ω resistor (1% precision) used internally for Ethernet PHY.
Fault	99	I	TTL	PWM Fault.
GND	9 15 21 33 39 45 54 57 63 69 82 87 94	-	Power	Ground reference for logic and I/O pins.

Table 18-2. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
GNDA	4 97	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
GNDPHY	42 85 86	-	Power	GND of the Ethernet PHY.
I2C0SCL	70	I/O	OD	I ² C module 0 clock.
I2C0SDA	71	I/O	OD	I ² C module 0 data.
LDO	7	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
LED0	59	0	TTL	Ethernet LED 0.
LED1	60	0	TTL	Ethernet LED 1.
MDIO	58	I/O	TTL	MDIO of the Ethernet PHY.
NC	50 51 52 53	-	-	No connect. Leave the pin electrically unconnected/isolated.
OSC0	48	I	Analog	Main oscillator crystal input or an external clock reference input.
OSC1	49	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
PA0	26	I/O	TTL	GPIO port A bit 0.
PA1	27	I/O	TTL	GPIO port A bit 1.
PA2	28	I/O	TTL	GPIO port A bit 2.
PA3	29	I/O	TTL	GPIO port A bit 3.
PA4	30	I/O	TTL	GPIO port A bit 4.
PA5	31	I/O	TTL	GPIO port A bit 5.
PA6	34	I/O	TTL	GPIO port A bit 6.
PA7	35	I/O	TTL	GPIO port A bit 7.
PB0	66	I/O	TTL	GPIO port B bit 0.
PB1	67	I/O	TTL	GPIO port B bit 1.
PB2	70	I/O	TTL	GPIO port B bit 2.
PB3	71	I/O	TTL	GPIO port B bit 3.
PB4	92	I/O	TTL	GPIO port B bit 4.
PB5	91	I/O	TTL	GPIO port B bit 5.
РВ6	90	I/O	TTL	GPIO port B bit 6.
PB7	89	I/O	TTL	GPIO port B bit 7.
PC0	80	I/O	TTL	GPIO port C bit 0.
PC1	79	I/O	TTL	GPIO port C bit 1.
PC2	78	I/O	TTL	GPIO port C bit 2.
PC3	77	I/O	TTL	GPIO port C bit 3.

Table 18-2. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
PC4	25	I/O	TTL	GPIO port C bit 4.
PC5	24	I/O	TTL	GPIO port C bit 5.
PC6	23	I/O	TTL	GPIO port C bit 6.
PC7	22	I/O	TTL	GPIO port C bit 7.
PD0	10	I/O	TTL	GPIO port D bit 0.
PD1	11	I/O	TTL	GPIO port D bit 1.
PD2	12	I/O	TTL	GPIO port D bit 2.
PD3	13	I/O	TTL	GPIO port D bit 3.
PD4	95	I/O	TTL	GPIO port D bit 4.
PD5	96	I/O	TTL	GPIO port D bit 5.
PD6	99	I/O	TTL	GPIO port D bit 6.
PD7	100	I/O	TTL	GPIO port D bit 7.
PE0	72	I/O	TTL	GPIO port E bit 0.
PE1	73	I/O	TTL	GPIO port E bit 1.
PE2	74	I/O	TTL	GPIO port E bit 2.
PE3	75	I/O	TTL	GPIO port E bit 3.
PE4	6	I/O	TTL	GPIO port E bit 4.
PF0	47	I/O	TTL	GPIO port F bit 0.
PF1	61	I/O	TTL	GPIO port F bit 1.
PF2	60	I/O	TTL	GPIO port F bit 2.
PF3	59	I/O	TTL	GPIO port F bit 3.
PG0	19	I/O	TTL	GPIO port G bit 0.
PG1	18	I/O	TTL	GPIO port G bit 1.
PWM0	10	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
PWM1	11	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.
RST	64	Į	TTL	System reset input.
RXIN	37	I	Analog	RXIN of the Ethernet PHY.
RXIP	40	I	Analog	RXIP of the Ethernet PHY.
SSIOClk	28	I/O	TTL	SSI module 0 clock.
SSI0Fss	29	I/O	TTL	SSI module 0 frame signal.
SSI0Rx	30	I	TTL	SSI module 0 receive.
SSIOTx	31	0	TTL	SSI module 0 transmit.
SWCLK	80	I	TTL	JTAG/SWD CLK.
SWDIO	79	I/O	TTL	JTAG TMS and SWDIO.
SWO	77	0	TTL	JTAG TDO and SWO.
TCK	80	I	TTL	JTAG/SWD CLK.
TDI	78	I	TTL	JTAG TDI.
TDO	77	0	TTL	JTAG TDO and SWO.
TMS	79	I/O	TTL	JTAG TMS and SWDIO.
TRST	89	I	TTL	JTAG TRST.
TXON	46	0	Analog	TXON of the Ethernet PHY.

Table 18-2. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
TXOP	43	0	Analog	TXOP of the Ethernet PHY.
U0Rx	26	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
UOTx	27	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
Ulrx	12	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
UlTx	13	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.
VCCPHY	36 83 84	-	Power	VCC of the Ethernet PHY.
VDD	8 20 32 44 55 56 68 81 93	-	Power	Positive supply for I/O and some logic.
VDD25	14 38 62 88	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDDA	3 98	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
XTALNPHY	17	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
XTALPPHY	16	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.1.3 Signals by Function, Except for GPIO

Table 18-3. Signals by Function, Except for GPIO

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	ADC0	1	1	Analog	Analog-to-digital converter input 0.
ADC	ADC1	2	I	Analog	Analog-to-digital converter input 1.
	ADC2	5	I	Analog	Analog-to-digital converter input 2.

Table 18-3. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	C0+	90	ı	Analog	Analog comparator 0 positive input.
	C0-	92	I	Analog	Analog comparator 0 negative input.
Analog Comparators	C0o	100	0	TTL	Analog comparator 0 output.
	C1+	24	I	Analog	Analog comparator 1 positive input.
	C1-	91	I	Analog	Analog comparator 1 negative input.
	ERBIAS	41	I	Analog	12.4- $k\Omega$ resistor (1% precision) used internally for Ethernet PHY.
	GNDPHY	42 85 86	-	Power	GND of the Ethernet PHY.
	LED0	59	0	TTL	Ethernet LED 0.
	LED1	60	0	TTL	Ethernet LED 1.
	MDIO	58	I/O	TTL	MDIO of the Ethernet PHY.
	RXIN	37	I	Analog	RXIN of the Ethernet PHY.
	RXIP	40	I	Analog	RXIP of the Ethernet PHY.
Ethernet	TXON	46	0	Analog	TXON of the Ethernet PHY.
	TXOP	43	0	Analog	TXOP of the Ethernet PHY.
	VCCPHY	36 83 84	-	Power	VCC of the Ethernet PHY.
	XTALNPHY	17	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
	XTALPPHY	16	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.
	CCP0	66	I/O	TTL	Capture/Compare/PWM 0.
General-Purpose	CCP1	34	I/O	TTL	Capture/Compare/PWM 1.
Timers	CCP2	67	I/O	TTL	Capture/Compare/PWM 2.
	CCP3	23	I/O	TTL	Capture/Compare/PWM 3.
12C	I2C0SCL	70	I/O	OD	I ² C module 0 clock.
120	I2C0SDA	71	I/O	OD	I ² C module 0 data.
	SWCLK	80	I	TTL	JTAG/SWD CLK.
	SWDIO	79	I/O	TTL	JTAG TMS and SWDIO.
	SWO	77	0	TTL	JTAG TDO and SWO.
ITA C/C/A/D/C/A/O	TCK	80	I	TTL	JTAG/SWD CLK.
JTAG/SWD/SWO	TDI	78	I	TTL	JTAG TDI.
	TDO	77	0	TTL	JTAG TDO and SWO.
	TMS	79	I/O	TTL	JTAG TMS and SWDIO.
	TRST	89	I	TTL	JTAG TRST.
	Fault	99	I	TTL	PWM Fault.
PWM	PWM0	10	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
	PWM1	11	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.

Table 18-3. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	GND	9 15 21 33 39 45 54 57 63 69 82 87 94	-	Power	Ground reference for logic and I/O pins.
	GNDA	4 97	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
Power	LDO	7	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
	VDD	8 20 32 44 55 56 68 81 93	-	Power	Positive supply for I/O and some logic.
	VDD25	14 38 62 88	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDDA	3 98	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
	SSI0Clk	28	I/O	TTL	SSI module 0 clock.
SSI	SSI0Fss	29	I/O	TTL	SSI module 0 frame signal.
331	SSI0Rx	30	I	TTL	SSI module 0 receive.
	SSI0Tx	31	0	TTL	SSI module 0 transmit.

Table 18-3. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	CMOD0	65	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
	CMOD1	76	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
System Control & Clocks	osc0	48	I	Analog	Main oscillator crystal input or an external clock reference input.
	osc1	49	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
	RST	64	I	TTL	System reset input.
	UORx	26	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
UART	UOTx	27	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
UART	U1Rx	12	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
	UlTx	13	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.1.4 GPIO Pins and Alternate Functions

Table 18-4. GPIO Pins and Alternate Functions

10	Pin Number	Multiplexed Function	Multiplexed Function
PA0	26	U0Rx	
PA1	27	U0Tx	
PA2	28	SSI0Clk	
PA3	29	SSI0Fss	
PA4	30	SSI0Rx	
PA5	31	SSIOTX	
PA6	34	CCP1	
PA7	35		
PB0	66	CCP0	
PB1	67	CCP2	
PB2	70	I2C0SCL	
PB3	71	I2C0SDA	
PB4	92	C0-	
PB5	91	C1-	
PB6	90	C0+	
PB7	89	TRST	
PC0	80	TCK	SWCLK
PC1	79	TMS	SWDIO
PC2	78	TDI	
PC3	77	TDO	SWO
PC4	25		

Table 18-4. GPIO Pins and Alternate Functions (continued)

10	Pin Number	Multiplexed Function	Multiplexed Function
PC5	24	C1+	
PC6	23	CCP3	
PC7	22		
PD0	10	PWM0	
PD1	11	PWM1	
PD2	12	U1Rx	
PD3	13	UlTx	
PD4	95		
PD5	96		
PD6	99	Fault	
PD7	100	C0o	
PE0	72		
PE1	73		
PE2	74		
PE3	75		
PE4	6		
PF0	47		
PF1	61		
PF2	60	LED1	
PF3	59	LED0	
PG0	19		
PG1	18		

18.2 108-Ball BGA Package Pin Tables

18.2.1 Signals by Pin Number

Table 18-5. Signals by Pin Number

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
A1	ADC1	I	Analog	Analog-to-digital converter input 1.
A2	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
A3	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
A4	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
A5	GNDA	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
A6	PB4	I/O	TTL	GPIO port B bit 4.
AU	C0-	I	Analog	Analog comparator 0 negative input.
A7	PB6	I/O	TTL	GPIO port B bit 6.
A	C0+	I	Analog	Analog comparator 0 positive input.
A8	PB7	I/O	TTL	GPIO port B bit 7.
Ao	TRST	I	TTL	JTAG TRST.

Table 18-5. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
	PC0	I/O	TTL	GPIO port C bit 0.
A9	SWCLK	I	TTL	JTAG/SWD CLK.
	TCK	I	TTL	JTAG/SWD CLK.
	PC3	I/O	TTL	GPIO port C bit 3.
A10	SWO	0	TTL	JTAG TDO and SWO.
	TDO	0	TTL	JTAG TDO and SWO.
A11	PE0	I/O	TTL	GPIO port E bit 0.
A12	PE3	I/O	TTL	GPIO port E bit 3.
B1	ADC0	I I	Analog	Analog-to-digital converter input 0.
B2	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
В3	ADC2	I.	Analog	Analog-to-digital converter input 2.
B4	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
B5	GNDA	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
В6	GND	-	Power	Ground reference for logic and I/O pins.
D7	PB5	I/O	TTL	GPIO port B bit 5.
B7 —	C1-	I I	Analog	Analog comparator 1 negative input.
Do	PC2	I/O	TTL	GPIO port C bit 2.
B8 —	TDI	I I	TTL	JTAG TDI.
	PC1	I/O	TTL	GPIO port C bit 1.
В9	SWDIO	I/O	TTL	JTAG TMS and SWDIO.
	TMS	I/O	TTL	JTAG TMS and SWDIO.
B10	CMOD1	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
B11	PE2	I/O	TTL	GPIO port E bit 2.
B12	PE1	I/O	TTL	GPIO port E bit 1.
C1	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
C2	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
C3	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
C4	GND	-	Power	Ground reference for logic and I/O pins.
C5	GND	-	Power	Ground reference for logic and I/O pins.
C6	VDDA	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
C7	VDDA	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.

Table 18-5. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
C8	GNDPHY	-	Power	GND of the Ethernet PHY.
C9	GNDPHY	-	Power	GND of the Ethernet PHY.
C10	VCCPHY	-	Power	VCC of the Ethernet PHY.
044	PB2	I/O	TTL	GPIO port B bit 2.
C11	I2C0SCL	I/O	OD	I ² C module 0 clock.
C42	PB3	I/O	TTL	GPIO port B bit 3.
C12 –	I2C0SDA	I/O	OD	I ² C module 0 data.
D1	PE4	I/O	TTL	GPIO port E bit 4.
D2	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
D3	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
D10	VCCPHY	-	Power	VCC of the Ethernet PHY.
D11	VCCPHY	-	Power	VCC of the Ethernet PHY.
D12 -	PB1	I/O	TTL	GPIO port B bit 1.
D12 =	CCP2	I/O	TTL	Capture/Compare/PWM 2.
E1	PD4	I/O	TTL	GPIO port D bit 4.
E2	PD5	I/O	TTL	GPIO port D bit 5.
E3	LDO	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
E10	VDD33	-	Power	Positive supply for I/O and some logic.
E11	CMOD0	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
E12 _	PB0	I/O	TTL	GPIO port B bit 0.
LIZ	CCP0	I/O	TTL	Capture/Compare/PWM 0.
F1 -	PD7	I/O	TTL	GPIO port D bit 7.
	C0o	0	TTL	Analog comparator 0 output.
F2 _	PD6	I/O	TTL	GPIO port D bit 6.
12	Fault	I	TTL	PWM Fault.
F3	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
F10	GND	-	Power	Ground reference for logic and I/O pins.
F11	GND	-	Power	Ground reference for logic and I/O pins.
F12	GND	-	Power	Ground reference for logic and I/O pins.
G1 _	PD0	I/O	TTL	GPIO port D bit 0.
<u> </u>	PWM0	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
G2	PD1	I/O	TTL	GPIO port D bit 1.
	PWM1	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.
G3	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
G10	VDD33	-	Power	Positive supply for I/O and some logic.
G11	VDD33	-	Power	Positive supply for I/O and some logic.

Table 18-5. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
G12	VDD33	-	Power	Positive supply for I/O and some logic.
	PD3	I/O	TTL	GPIO port D bit 3.
H1	UlTx	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.
	PD2	I/O	TTL	GPIO port D bit 2.
H2	U1Rx	1	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
H3	GND	-	Power	Ground reference for logic and I/O pins.
H10	VDD33	-	Power	Positive supply for I/O and some logic.
H11	RST	I	TTL	System reset input.
H12	PF1	I/O	TTL	GPIO port F bit 1.
J1	XTALNPHY	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
J2	XTALPPHY	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.
J3	GND	-	Power	Ground reference for logic and I/O pins.
J10	GND	-	Power	Ground reference for logic and I/O pins.
J11 _	PF2	I/O	TTL	GPIO port F bit 2.
311	LED1	0	TTL	Ethernet LED 1.
J12 _	PF3	I/O	TTL	GPIO port F bit 3.
012	LED0	0	TTL	Ethernet LED 0.
K1	PG0	I/O	TTL	GPIO port G bit 0.
K2	PG1	I/O	TTL	GPIO port G bit 1.
K3	ERBIAS	I	Analog	12.4-kΩ resistor (1% precision) used internally for Ethernet PHY.
K4	GNDPHY	-	Power	GND of the Ethernet PHY.
K5	GND	-	Power	Ground reference for logic and I/O pins.
K6	GND	-	Power	Ground reference for logic and I/O pins.
K7	VDD33	-	Power	Positive supply for I/O and some logic.
K8	VDD33	-	Power	Positive supply for I/O and some logic.
K9	VDD33	-	Power	Positive supply for I/O and some logic.
K10	GND	-	Power	Ground reference for logic and I/O pins.
K11	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
K12	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
L1	PC4	I/O	TTL	GPIO port C bit 4.
L2	PC7	I/O	TTL	GPIO port C bit 7.
	PA0	I/O	TTL	GPIO port A bit 0.
L3	U0Rx	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
L4	PA3	I/O	TTL	GPIO port A bit 3.
	SSI0Fss	I/O	TTL	SSI module 0 frame signal.
L5 _	PA4	I/O	TTL	GPIO port A bit 4.
	SSI0Rx	ı	TTL	SSI module 0 receive.

Table 18-5. Signals by Pin Number (continued)

Pin Number	Pin Name	Pin Type	Buffer Type ^a	Description
L6	PA6	I/O	TTL	GPIO port A bit 6.
LO	CCP1	I/O	TTL	Capture/Compare/PWM 1.
L7	RXIN	I	Analog	RXIN of the Ethernet PHY.
L8	TXON	0	Analog	TXON of the Ethernet PHY.
L9	MDIO	I/O	TTL	MDIO of the Ethernet PHY.
L10	GND	-	Power	Ground reference for logic and I/O pins.
L11	osc0	I	Analog	Main oscillator crystal input or an external clock reference input.
L12	VDD	-	Power	Positive supply for I/O and some logic.
M1	PC5	I/O	TTL	GPIO port C bit 5.
IVI I	C1+	I	Analog	Analog comparator 1 positive input.
M2	PC6	I/O	TTL	GPIO port C bit 6.
IVIZ	CCP3	I/O	TTL	Capture/Compare/PWM 3.
	PA1	I/O	TTL	GPIO port A bit 1.
M3	UOTx	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
M4	PA2	I/O	TTL	GPIO port A bit 2.
IVI 4	SSI0Clk	I/O	TTL	SSI module 0 clock.
M5	PA5	I/O	TTL	GPIO port A bit 5.
CIVI	SSI0Tx	0	TTL	SSI module 0 transmit.
M6	PA7	I/O	TTL	GPIO port A bit 7.
M7	RXIP	ı	Analog	RXIP of the Ethernet PHY.
M8	TXOP	0	Analog	TXOP of the Ethernet PHY.
M9	PF0	I/O	TTL	GPIO port F bit 0.
M10	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
M11	OSC1	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
M12	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.2.2 Signals by Signal Name

Table 18-6. Signals by Signal Name

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
ADC0	B1	1	Analog	Analog-to-digital converter input 0.
ADC1	A1	1	Analog	Analog-to-digital converter input 1.
ADC2	В3	1	Analog	Analog-to-digital converter input 2.
C0+	A7	1	Analog	Analog comparator 0 positive input.
C0-	A6	1	Analog	Analog comparator 0 negative input.
COo	F1	0	TTL	Analog comparator 0 output.
C1+	M1	1	Analog	Analog comparator 1 positive input.
C1-	В7	1	Analog	Analog comparator 1 negative input.
CCP0	E12	I/O	TTL	Capture/Compare/PWM 0.

Table 18-6. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
CCP1	L6	I/O	TTL	Capture/Compare/PWM 1.
CCP2	D12	I/O	TTL	Capture/Compare/PWM 2.
CCP3	M2	I/O	TTL	Capture/Compare/PWM 3.
CMOD0	E11	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	B10	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
ERBIAS	K3	I	Analog	12.4-k Ω resistor (1% precision) used internally for Ethernet PHY.
Fault	F2	I	TTL	PWM Fault.
GND	B6 C4 C5 F10 F11 F12 H3 J3 J10 K5 K6 K10	-	Power	Ground reference for logic and I/O pins.
GNDA	A5 B5	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
GNDPHY	C8 C9 K4	-	Power	GND of the Ethernet PHY.
I2C0SCL	C11	I/O	OD	I ² C module 0 clock.
I2C0SDA	C12	I/O	OD	I ² C module 0 data.
LDO	E3	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
LED0	J12	0	TTL	Ethernet LED 0.
LED1	J11	0	TTL	Ethernet LED 1.
MDIO	L9	I/O	TTL	MDIO of the Ethernet PHY.
NC	A2 A3 A4 B2 B4 C1 C2 D2 K11 K12 M10	-	-	No connect. Leave the pin electrically unconnected/isolated.

Table 18-6. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description	
osc0	L11	I	Analog	Main oscillator crystal input or an external clock reference input.	
OSC1	M11	0	Analog	Main oscillator crystal output. Leave unconnected when us a single-ended clock source.	
PA0	L3	I/O	TTL	GPIO port A bit 0.	
PA1	M3	I/O	TTL	GPIO port A bit 1.	
PA2	M4	I/O	TTL	GPIO port A bit 2.	
PA3	L4	I/O	TTL	GPIO port A bit 3.	
PA4	L5	I/O	TTL	GPIO port A bit 4.	
PA5	M5	I/O	TTL	GPIO port A bit 5.	
PA6	L6	I/O	TTL	GPIO port A bit 6.	
PA7	M6	I/O	TTL	GPIO port A bit 7.	
PB0	E12	I/O	TTL	GPIO port B bit 0.	
PB1	D12	I/O	TTL	GPIO port B bit 1.	
PB2	C11	I/O	TTL	GPIO port B bit 2.	
PB3	C12	I/O	TTL	GPIO port B bit 3.	
PB4	A6	I/O	TTL	GPIO port B bit 4.	
PB5	B7	I/O	TTL	GPIO port B bit 5.	
PB6	A7	I/O	TTL	GPIO port B bit 6.	
PB7	A8	I/O	TTL	GPIO port B bit 7.	
PC0	A9	I/O	TTL	GPIO port C bit 0.	
PC1	B9	I/O	TTL	GPIO port C bit 1.	
PC2	B8	I/O	TTL	GPIO port C bit 2.	
PC3	A10	I/O	TTL	GPIO port C bit 3.	
PC4	L1	I/O	TTL	GPIO port C bit 4.	
PC5	M1	I/O	TTL	GPIO port C bit 5.	
PC6	M2	I/O	TTL	GPIO port C bit 6.	
PC7	L2	I/O	TTL	GPIO port C bit 7.	
PD0	G1	I/O	TTL	GPIO port D bit 0.	
PD1	G2	I/O	TTL	GPIO port D bit 1.	
PD2	H2	I/O	TTL	GPIO port D bit 2.	
PD3	H1	I/O	TTL	GPIO port D bit 3.	
PD4	E1	I/O	TTL	GPIO port D bit 4.	
PD5	E2	I/O	TTL	GPIO port D bit 5.	
PD6	F2	I/O	TTL	GPIO port D bit 6.	
PD7	F1	I/O	TTL	GPIO port D bit 7.	
PE0	A11	I/O	TTL	GPIO port E bit 0.	
PE1	B12	I/O	TTL	GPIO port E bit 1.	
PE2	B11	I/O	TTL	GPIO port E bit 2.	
PE3	A12	I/O	TTL	GPIO port E bit 3.	
PE4	D1	I/O	TTL	GPIO port E bit 4.	
PF0	M9	I/O	TTL	GPIO port F bit 0.	

Table 18-6. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description	
PF1	H12	I/O	TTL	GPIO port F bit 1.	
PF2	J11	I/O	TTL	GPIO port F bit 2.	
PF3	J12	I/O	TTL	GPIO port F bit 3.	
PG0	K1	I/O	TTL	GPIO port G bit 0.	
PG1	K2	I/O	TTL	GPIO port G bit 1.	
PWM0	G1	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.	
PWM1	G2	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.	
RST	H11	I	TTL	System reset input.	
RXIN	L7	I	Analog	RXIN of the Ethernet PHY.	
RXIP	M7	I	Analog	RXIP of the Ethernet PHY.	
SSIOClk	M4	I/O	TTL	SSI module 0 clock.	
SSI0Fss	L4	I/O	TTL	SSI module 0 frame signal.	
SSIORx	L5	I	TTL	SSI module 0 receive.	
SSIOTx	M5	0	TTL	SSI module 0 transmit.	
SWCLK	A9	I	TTL	JTAG/SWD CLK.	
SWDIO	В9	I/O	TTL	JTAG TMS and SWDIO.	
SWO	A10	0	TTL	JTAG TDO and SWO.	
TCK	A9	I	TTL	JTAG/SWD CLK.	
TDI	B8	I	TTL	JTAG TDI.	
TDO	A10	0	TTL	JTAG TDO and SWO.	
TMS	В9	I/O	TTL	JTAG TMS and SWDIO.	
TRST	A8	I	TTL	JTAG TRST.	
TXON	L8	0	Analog	TXON of the Ethernet PHY.	
TXOP	M8	0	Analog	TXOP of the Ethernet PHY.	
U0Rx	L3	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.	
UOTx	M3	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.	
U1Rx	H2	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.	
UlTx	H1	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.	
VCCPHY	C10 D10 D11	-	Power	VCC of the Ethernet PHY.	
VDD	L12	-	Power	Positive supply for I/O and some logic.	
VDD25	C3 D3 F3 G3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.	

Table 18-6. Signals by Signal Name (continued)

Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
VDD33	E10 G10 G11 G12 H10 K7 K8	-	Power	Positive supply for I/O and some logic.
VDDA	C6 C7	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
XTALNPHY	J1	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
XTALPPHY	J2	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.2.3 Signals by Function, Except for GPIO

Table 18-7. Signals by Function, Except for GPIO

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	ADC0	B1	I	Analog	Analog-to-digital converter input 0.
ADC	ADC1	A1	I	Analog	Analog-to-digital converter input 1.
	ADC2	В3	ļ	Analog	Analog-to-digital converter input 2.
	C0+	A7	I	Analog	Analog comparator 0 positive input.
	C0-	A6	I	Analog	Analog comparator 0 negative input.
Analog Comparators	C0o	F1	0	TTL	Analog comparator 0 output.
	C1+	M1	I	Analog	Analog comparator 1 positive input.
	C1-	В7	I	Analog	Analog comparator 1 negative input.

Table 18-7. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	ERBIAS	K3	I	Analog	12.4- $k\Omega$ resistor (1% precision) used internally for Ethernet PHY.
	GNDPHY	C8 C9 K4	-	Power	GND of the Ethernet PHY.
	LED0	J12	0	TTL	Ethernet LED 0.
	LED1	J11	0	TTL	Ethernet LED 1.
	MDIO	L9	I/O	TTL	MDIO of the Ethernet PHY.
	RXIN	L7	I	Analog	RXIN of the Ethernet PHY.
Eth a wa at	RXIP	M7	I	Analog	RXIP of the Ethernet PHY.
Ethernet	TXON	L8	0	Analog	TXON of the Ethernet PHY.
	TXOP	M8	0	Analog	TXOP of the Ethernet PHY.
	VCCPHY	C10 D10 D11	-	Power	VCC of the Ethernet PHY.
	XTALNPHY	J1	0	TTL	Ethernet PHY XTALN 25-MHz oscillator crystal output. Connect this pin to ground when using a single-ended 25-MHz clock input connected to the XTALPPHY pin.
	XTALPPHY	J2	I	TTL	Ethernet PHY XTALP 25-MHz oscillator crystal input or external clock reference input.
	CCP0	E12	I/O	TTL	Capture/Compare/PWM 0.
General-Purpose	CCP1	L6	I/O	TTL	Capture/Compare/PWM 1.
Timers	CCP2	D12	I/O	TTL	Capture/Compare/PWM 2.
	CCP3	M2	I/O	TTL	Capture/Compare/PWM 3.
I2C	I2C0SCL	C11	I/O	OD	I ² C module 0 clock.
120	I2C0SDA	C12	I/O	OD	I ² C module 0 data.
	SWCLK	A9	I	TTL	JTAG/SWD CLK.
	SWDIO	В9	I/O	TTL	JTAG TMS and SWDIO.
	SWO	A10	0	TTL	JTAG TDO and SWO.
JTAG/SWD/SWO	TCK	A9	I	TTL	JTAG/SWD CLK.
J IAG/SWD/SWO	TDI	B8	I	TTL	JTAG TDI.
	TDO	A10	0	TTL	JTAG TDO and SWO.
	TMS	В9	I/O	TTL	JTAG TMS and SWDIO.
	TRST	A8	I	TTL	JTAG TRST.
	Fault	F2	I	TTL	PWM Fault.
PWM	РWМ0	G1	0	TTL	PWM 0. This signal is controlled by PWM Generator 0.
	PWM1	G2	0	TTL	PWM 1. This signal is controlled by PWM Generator 0.

Table 18-7. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	GND	B6 C4 C5 F10 F11 F12 H3 J3 J10 K5 K6 K10	-	Power	Ground reference for logic and I/O pins.
	GNDA	A5 B5	-	Power	The ground reference for the analog circuits (ADC, Analog Comparators, etc.). These are separated from GND to minimize the electrical noise contained on VDD from affecting the analog functions.
Power	LDO	E3	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 μ F or greater. The LDO pin must also be connected to the VDD25 pins at the board level in addition to the decoupling capacitor(s).
	VDD	L12	-	Power	Positive supply for I/O and some logic.
	VDD25	C3 D3 F3 G3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD33	E10 G10 G11 G12 H10 K7 K8	-	Power	Positive supply for I/O and some logic.
	VDDA	C6 C7	-	Power	The positive supply for the analog circuits (ADC, Analog Comparators, etc.). These are separated from VDD to minimize the electrical noise contained on VDD from affecting the analog functions. VDDA pins must be supplied with a voltage that meets the specification in "Recommended DC Operating Conditions" on page 645, regardless of system implementation.
	SSI0Clk	M4	I/O	TTL	SSI module 0 clock.
SSI	SSI0Fss	L4	I/O	TTL	SSI module 0 frame signal.
	SSI0Rx	L5	I	TTL	SSI module 0 receive.
	SSIOTx	M5	0	TTL	SSI module 0 transmit.

Table 18-7. Signals by Function, Except for GPIO (continued)

Function	Pin Name	Pin Number	Pin Type	Buffer Type ^a	Description
	CMOD0	E11	I	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
	CMOD1	B10	I	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
System Control & Clocks	osc0	L11	I	Analog	Main oscillator crystal input or an external clock reference input.
	osc1	M11	0	Analog	Main oscillator crystal output. Leave unconnected when using a single-ended clock source.
	RST	H11	I	TTL	System reset input.
UART	UORx	L3	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
	UOTx	M3	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
	U1Rx	H2	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
	UlTx	H1	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

a. The TTL designation indicates the pin has TTL-compatible voltage levels.

18.2.4 GPIO Pins and Alternate Functions

Table 18-8. GPIO Pins and Alternate Functions

10	Pin Number	Multiplexed Function	Multiplexed Function
PA0	L3	U0Rx	
PA1	M3	UOTx	
PA2	M4	SSI0Clk	
PA3	L4	SSI0Fss	
PA4	L5	SSI0Rx	
PA5	M5	SSI0Tx	
PA6	L6	CCP1	
PA7	M6		
PB0	E12	CCP0	
PB1	D12	CCP2	
PB2	C11	I2C0SCL	
PB3	C12	I2C0SDA	
PB4	A6	C0-	
PB5	B7	C1-	
PB6	A7	C0+	
PB7	A8	TRST	
PC0	A9	TCK	SWCLK
PC1	B9	TMS	SWDIO
PC2	B8	TDI	
PC3	A10	TDO	SWO
PC4	L1		

Table 18-8. GPIO Pins and Alternate Functions (continued)

10	Pin Number	Multiplexed Function	Multiplexed Function
PC5	M1	C1+	
PC6	M2	CCP3	
PC7	L2		
PD0	G1	PWM0	
PD1	G2	PWM1	
PD2	H2	U1Rx	
PD3	H1	UlTx	
PD4	E1		
PD5	E2		
PD6	F2	Fault	
PD7	F1	C0o	
PE0	A11		
PE1	B12		
PE2	B11		
PE3	A12		
PE4	D1		
PF0	M9		
PF1	H12		
PF2	J11	LED1	
PF3	J12	LED0	
PG0	K1		
PG1	K2		

18.3 Connections for Unused Signals

Table 18-9 on page 641 show how to handle signals for functions that are not used in a particular system implementation for devices that are in a 100-pin LQFP package. Two options are shown in the table: an acceptable practice and a preferred practice for reduced power consumption and improved EMC characteristics. If a module is not used in a system, and its inputs are grounded, it is important that the clock to the module is never enabled by setting the corresponding bit in the **RCGCx** register.

Table 18-9. Connections for Unused Signals (100-pin LQFP)

Function	Signal Name	Pin Number	Acceptable Practice	Preferred Practice
ADC	ADC0	1	NC	GNDA
	ADC1	2		
	ADC2	5		

Table 18-9. Connections for Unused Signals (100-pin LQFP) (continued)

Function	Signal Name	Pin Number	Acceptable Practice	Preferred Practice
	ERBIAS	41	Connect to GND through 12.4-kΩ resistor.	Connect to GND through 12.4-kΩ resistor.
Ethernet	GNDPHY	42	GND	GND
		85		
		86		
	MDIO ^a	58	NC	NC
	RXIN	37	NC	GND
	RXIP	40	NC	GND
	TXON	46	NC	GND
	TXOP	43	NC	GND
	VCCPHY	36	VDD	VDD
		83		
		84		
	XTALNPHY ^a	17	NC	NC
	XTALPPHY ^a	16	NC	GND
GPIO	All unused GPIOs	-	NC	GND
No Connects	NC	-	NC	NC
	OSC0	48	NC	GND
System Control	OSC1	49	NC	NC
	RST	64	Pull up as shown in Figure 5-1 on page 170	Connect through a capacitor to GND as close to pin as possible

a. Note that the Ethernet PHY is powered up by default. The PHY cannot be powered down unless a clock source is provided and the MDIO pin is pulled up through a 10-kΩ resistor.

Table 18-10 on page 642 show how to handle signals for functions that are not used in a particular system implementation for devices that are in a 108-pin BGA package. Two options are shown in the table: an acceptable practice and a preferred practice for reduced power consumption and improved EMC characteristics. If a module is not used in a system, and its inputs are grounded, it is important that the clock to the module is never enabled by setting the corresponding bit in the **RCGCx** register.

Table 18-10. Connections for Unused Signals, 108-pin BGA

Function	Signal Name	Pin Number	Acceptable Practice	Preferred Practice
ADC	ADC0	B1	NC	GNDA
	ADC1	A1		
	ADC2	В3		

Table 18-10. Connections for Unused Signals, 108-pin BGA (continued)

Function	Signal Name	Pin Number	Acceptable Practice	Preferred Practice
	ERBIAS	K3	Connect to GND through 12.4-kΩ resistor.	Connect to GND through 12.4-kΩ resistor.
	GNDPHY	C8	GND	GND
		C9		
		K4		
	MDIO ^a	L9	NC	NC
	RXIN	L7	NC	GND
Ethernet	RXIP	M7	NC	GND
	TXON	L8	NC	GND
	TXOP	M8	NC	GND
	VCCPHY	C10	VDD	VDD
		D10		
		D11		
	XTALNPHY ^a	J1	NC	NC
	XTALPPHY ^a	J2	NC	GND
GPIO	All unused GPIOs	-	NC	GND
No Connects	NC	-	NC	NC
	osc0	L11	NC	GND
System	osc1	M11	NC	NC
Control	RST	H11	Pull up as shown in Figure 5-1 on page 170	Connect through a capacitor to GND as close to pin as possible

a. Note that the Ethernet PHY is powered up by default. The PHY cannot be powered down unless a clock source is provided and the MDIO pin is pulled up through a 10-kΩ resistor.

19 Operating Characteristics

Table 19-1. Temperature Characteristics

Characteristic	Symbol	Value	Unit
Industrial operating temperature range	T _A	-40 to +85	°C
Extended operating temperature range	T _A	-40 to +105	°C
Unpowered storage temperature range	T _S	-65 to +150	°C

Table 19-2. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance (junction to ambient) ^a	Θ_{JA}	32	°C/W
Junction temperature ^b	T _J	$T_A + (P \cdot \Theta_{JA})$	°C

a. Junction to ambient thermal resistance θ_{JA} numbers are determined by a package simulator.

Table 19-3. ESD Absolute Maximum Ratings^a

Parameter Name	Min	Nom	Max	Unit
V _{ESDHBM}	-	-	2.0	kV
V _{ESDCDM}	-	-	1.0	kV
V _{ESDMM}	-	-	100	V

a. All Stellaris parts are ESD tested following the JEDEC standard.

b. Power dissipation is a function of temperature.

20 Electrical Characteristics

20.1 DC Characteristics

20.1.1 Maximum Ratings

The maximum ratings are the limits to which the device can be subjected without permanently damaging the device.

Note: The device is not guaranteed to operate properly at the maximum ratings.

Table 20-1. Maximum Ratings

Characteristic ^a	Symbol	\	Unit	
ona acteristic	Symbol	Min	Max	Oille
I/O supply voltage (V _{DD})	V _{DD}	0	4	V
Core supply voltage (V _{DD25})	V _{DD25}	0	3	V
Analog supply voltage (V _{DDA})	V_{DDA}	0	4	V
Ethernet PHY supply voltage (V _{CCPHY})	V _{CCPHY}	0	4	V
Input voltage	V	-0.3	5.5	V
Input voltage for a GPIO configured as an analog input	V_{IN}	-0.3	V _{DD} + 0.3	V
Maximum current per output pins	I	-	25	mA
Maximum input voltage on a non-power pin when the microcontroller is unpowered	V _{NON}	-	300	mV

a. Voltages are measured with respect to GND.

Important: This device contains circuitry to protect the inputs against damage due to high-static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either GND or VDD).

20.1.2 Recommended DC Operating Conditions

For special high-current applications, the GPIO output buffers may be used with the following restrictions. With the GPIO pins configured as 8-mA output drivers, a total of four GPIO outputs may be used to sink current loads up to 18 mA each. At 18-mA sink current loading, the V_{OL} value is specified as 1.2 V. The high-current GPIO package pins must be selected such that there are only a maximum of two per side of the physical package or BGA pin group with the total number of high-current GPIO outputs not exceeding four for the entire package.

Table 20-2. Recommended DC Operating Conditions

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{DD}	I/O supply voltage	3.0	3.3	3.6	V
V _{DD25}	Core supply voltage	2.25	2.5	2.75	V
V_{DDA}	Analog supply voltage	3.0	3.3	3.6	V
V _{CCPHY}	Ethernet PHY supply voltage	3.0	3.3	3.6	V
V _{IH}	High-level input voltage	2.0	-	5.0	V
V _{IL}	Low-level input voltage	-0.3	-	1.3	V

Table 20-2. Recommended DC Operating Conditions (continued)

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{OH}	High-level output voltage	2.4	-	-	V
V _{OL} ^a	Low-level output voltage	-	-	0.4	V
	High-level source current, V _{OH} =2.4 V				
I	2-mA Drive	2.0	-	-	mA
Іон	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	-	-	mA
	Low-level sink current, V _{OL} =0.4 V				
1	2-mA Drive	2.0	-	-	mA
loL	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	-	-	mA

a. $\rm V_{OL}$ and $\rm V_{OH}$ shift to 1.2 V when using high-current GPIOs.

20.1.3 On-Chip Low Drop-Out (LDO) Regulator Characteristics

Table 20-3. LDO Regulator Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{LDOOUT}	Programmable internal (logic) power supply output value	2.25	2.5	2.75	V
	Output voltage accuracy	-	2%	-	%
t _{PON}	Power-on time	-	-	100	μs
t _{ON}	Time on	-	-	200	μs
t _{OFF}	Time off	-	-	100	μs
V _{STEP}	Step programming incremental voltage	-	50	-	mV
C _{LDO}	External filter capacitor size for internal power supply	1.0	-	3.0	μF

20.1.4 **GPIO Module Characteristics**

Table 20-4. GPIO Module DC Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
R _{GPIOPU}	GPIO internal pull-up resistor	50	-	110	kΩ
R _{GPIOPD}	GPIO internal pull-down resistor	55	-	180	kΩ
I _{LKG}	GPIO input leakage current ^a	-	-	2	μΑ

a. The leakage current is measured with GND or V_{DD} applied to the corresponding pin(s). The leakage of digital port pins is measured individually. The port pin is configured as an input and the pullup/pulldown resistor is disabled.

20.1.5 Power Specifications

The power measurements specified in the tables that follow are run on the core processor using SRAM with the following specifications (except as noted):

- V_{DDA} = 3.3 V
- V_{DDPHY} = 3.3 V
- Temperature = 25°C
- Clock Source (MOSC) =3.579545 MHz Crystal Oscillator
- Main oscillator (MOSC) = enabled
- Internal oscillator (IOSC) = disabled

Table 20-5. Detailed Power Specifications

Parameter	Parameter Name	ameter Name Conditions		V _{DD} , V _{DDA} ,	2.5 V V _{DD25}		Unit
			Nom	Max	Nom	Max	
	Run mode 1	V _{DD25} = 2.50 V	48	pending ^a	108	pending ^a	mA
	(Flash loop)	Code= while(1){} executed out of Flash					
		Peripherals = All ON					
		System Clock = 50 MHz (with PLL)					
	Run mode 2	V _{DD25} = 2.50 V	5	pending ^a	52	pending ^a	mA
	(Flash loop)	Code= while(1){} executed out of Flash					
		Peripherals = All OFF					
I _{DD_RUN}		System Clock = 50 MHz (with PLL)					
	Run mode 1	V _{DD25} = 2.50 V	48	pendinga	100	pending ^a	mA
	(SRAM loop)	Code= while(1){} executed in SRAM					
		Peripherals = All ON					
		System Clock = 50 MHz (with PLL)					
	Run mode 2	V _{DD25} = 2.50 V	5	pending ^a	45	pending ^a	mA
	(SRAM loop)	Code= while(1){} executed in SRAM					
		Peripherals = All OFF					
		System Clock = 50 MHz (with PLL)					
I _{DD_SLEEP}	Sleep mode	V _{DD25} = 2.50 V	5	pending ^a	16	pending ^a	mA
		Peripherals = All OFF					
		System Clock = 50 MHz (with PLL)					
I _{DD_DEEPSLEEP}	Deep-Sleep mode	LDO = 2.25 V	4.6	pending ^a	0.21	pending ^a	mA
		Peripherals = All OFF					
		System Clock = IOSC30KHZ/64					

a. Pending characterization completion.

20.1.6 Flash Memory Characteristics

Table 20-6. Flash Memory Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
PE _{CYC}	Number of guaranteed program/erase cycles before failure ^a	10,000	100,000	-	cycles

Table 20-6. Flash Memory Characteristics (continued)

Parameter	Parameter Name	Min	Nom	Max	Unit
T _{RET}	Data retention at average operating temperature of 85°C (industrial) or 105°C (extended)	10	-	-	years
T _{PROG}	Word program time	20	-	-	μs
T _{ERASE}	Page erase time	20	-	-	ms
T _{ME}	Mass erase time	-	-	250	ms

a. A program/erase cycle is defined as switching the bits from 1-> 0 -> 1.

20.1.7 Ethernet Controller

Table 20-7. Ethernet Controller DC Characteristics

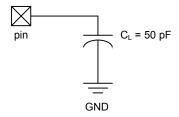
Parameter	Parameter Name	Value	Unit
R _{EBIAS}	Value of the pull-down resistor on the ERBIAS pin	12.4K ± 1 %	Ω

20.2 AC Characteristics

20.2.1 Load Conditions

Unless otherwise specified, the following conditions are true for all timing measurements. Timing measurements are for 4-mA drive strength.

Figure 20-1. Load Conditions



20.2.2 Clocks

Table 20-8. Phase Locked Loop (PLL) Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
f _{ref_crystal}	Crystal reference ^a	3.579545	-	8.192	MHz
f _{ref_ext}	External clock reference ^a	3.579545	-	8.192	MHz
f _{pll}	PLL frequency ^b	-	400	-	MHz
T _{READY}	PLL lock time	-	-	0.5	ms

a. The exact value is determined by the crystal value programmed into the XTAL field of the **Run-Mode Clock Configuration** (RCC) register.

Table 20-9 on page 649 shows the actual frequency of the PLL based on the crystal frequency used (defined by the XTAL field in the **RCC** register).

b. PLL frequency is automatically calculated by the hardware based on the \mathtt{XTAL} field of the RCC register.

Table 20-9. Actual PLL Frequency

XTAL	Crystal Frequency (MHz)	PLL Frequency (MHz)	Error
0x4	3.5795	400.904	0.0023%
0x5	3.6864	398.1312	0.0047%
0x6	4.0	400	-
0x7	4.096	401.408	0.0035%
0x8	4.9152	398.1312	0.0047%
0x9	5.0	400	-
0xA	5.12	399.36	0.0016%
0xB	6.0	400	-
0xC	6.144	399.36	0.0016%
0xD	7.3728	398.1312	0.0047%
0xE	8.0	400	0.0047%
0xF	8.192	398.6773333	0.0033%

Table 20-10. Clock Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
f _{IOSC}	Internal 12 MHz oscillator frequency	8.4	12	15.6	MHz
f _{IOSC30KHZ}	Internal 30 KHz oscillator frequency	15	30	45	KHz
f _{MOSC}	Main oscillator frequency	1	-	8.192	MHz
t _{MOSC_per}	Main oscillator period	125	-	1000	ns
f _{ref_crystal_bypass}	Crystal reference using the main oscillator (PLL in BYPASS mode) ^a	1	-	8.192	MHz
f _{ref_ext_bypass}	External clock reference (PLL in BYPASS mode) ^a	0	-	50	MHz
f _{system_clock}	System clock	0	-	50	MHz

a. The ADC must be clocked from the PLL or directly from a 16-MHz clock source to operate properly.

Table 20-11. Crystal Characteristics

Parameter Name		Value				
Frequency	8	6	4	3.5	MHz	
Frequency tolerance ^a	±50	±50	±50	±50	ppm	
Oscillation mode	Parallel	Parallel	Parallel	Parallel	-	
Motional capacitance (typ)	27.8	37.0	55.6	63.5	pF	
Motional inductance (typ)	14.3	19.1	28.6	32.7	mH	
Equivalent series resistance (max)	120	160	200	220	Ω	
Shunt capacitance (max)	10	10	10	10	pF	
Load capacitance (typ)	16	16	16	16	pF	
Drive level (typ)	100	100	100	100	μW	

a. This tolerance provides a guard band for temperature stability and aging drift.

20.2.2.1 System Clock Specifications with ADC Operation

Table 20-12. System Clock Characteristics with ADC Operation

Parameter	Parameter Name	Min	Nom	Max	Unit
f _{sysadc}	System clock frequency when the ADC module is	16	-	-	MHz
	operating (when PLL is bypassed)				

20.2.3 JTAG and Boundary Scan

Table 20-13. JTAG Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
J1	f _{TCK}	TCK operational clock frequency	0	-	10	MHz
J2	t _{TCK}	TCK operational clock period	100	-	-	ns
J3	t _{TCK_LOW}	TCK clock Low time	-	t _{TCK} /2	-	ns
J4	t _{TCK_HIGH}	TCK clock High time	-	t _{TCK} /2	-	ns
J5	t _{TCK_R}	TCK rise time	0	-	10	ns
J6	t _{TCK_F}	TCK fall time	0	-	10	ns
J7	t _{TMS_SU}	TMS setup time to TCK rise	20	-	-	ns
J8	t _{TMS_HLD}	TMS hold time from TCK rise	20	-	-	ns
J9	t _{TDI_SU}	TDI setup time to TCK rise	25	-	-	ns
J10	t _{TDI_HLD}	TDI hold time from TCK rise	25	-	-	ns
		2-mA drive		23	35	ns
J11	TCK fall to Data	4-mA drive	_	15	26	ns
t _{TDO_ZDV}	Valid from High-Z	8-mA drive	_	14	25	ns
		8-mA drive with slew rate control		18	29	ns
		2-mA drive		21	35	ns
J12	TCK fall to Data Valid from Data	4-mA drive	_	14	25	ns
t _{TDO_DV}	Valid	8-mA drive		13	24	ns
		8-mA drive with slew rate control		18	28	ns
		2-mA drive		9	11	ns
J13	тск fall to High-Z	4-mA drive	_	7	9	ns
t _{TDO_DVZ}	from Data Valid	8-mA drive		6	8	ns
		8-mA drive with slew rate control		7	9	ns
J14	t _{TRST}	TRST assertion time	100	-	-	ns
J15	t _{TRST_SU}	TRST setup time to TCK rise	10	-	-	ns

Figure 20-2. JTAG Test Clock Input Timing

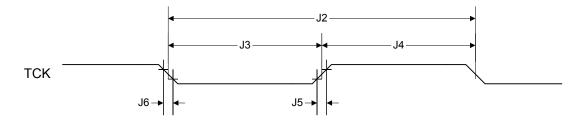


Figure 20-3. JTAG Test Access Port (TAP) Timing

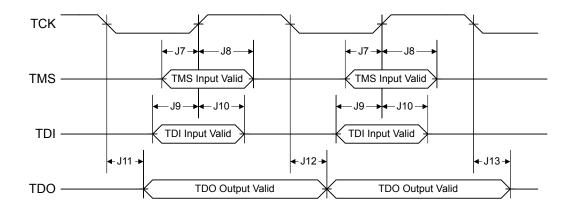
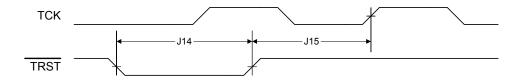


Figure 20-4. JTAG TRST Timing



20.2.4 Reset

Table 20-14. Reset Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
R1	V _{TH}	Reset threshold	-	2.0	-	V
R2	V _{BTH}	Brown-Out threshold	2.85	2.9	2.95	V
R3	T _{POR}	Power-On Reset timeout	-	10	-	ms
R4	T _{BOR}	Brown-Out timeout	-	500	-	μs
R5	T _{IRPOR}	Internal reset timeout after POR	6	-	11	ms
R6	T _{IRBOR}	Internal reset timeout after BOR ^a	0	-	1	μs

Table 20-14. Reset Characteristics (continued)

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
R7	T _{IRHWR}	Internal reset timeout after hardware reset $(\overline{\mbox{\scriptsize RST}}\mbox{ pin})$	0	-	1	ms
R8	T _{IRSWR}	Internal reset timeout after software-initiated system reset ^a	2.5	-	20	μs
R9	T _{IRWDR}	Internal reset timeout after watchdog reset ^a	2.5	-	20	μs
R10	T _{VDDRISE}	Supply voltage (V _{DD}) rise time (0V-3.3V)	-	-	100	ms
R11	T _{MIN}	Minimum RST pulse width	2	-	-	μs

a. 20 * t _{MOSC_per}

Figure 20-5. External Reset Timing (RST)

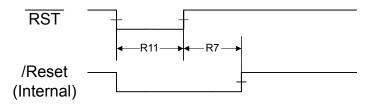


Figure 20-6. Power-On Reset Timing

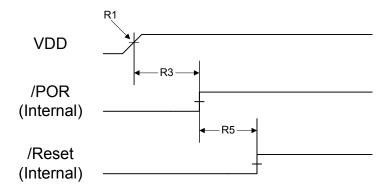


Figure 20-7. Brown-Out Reset Timing

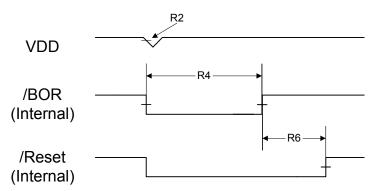


Figure 20-8. Software Reset Timing

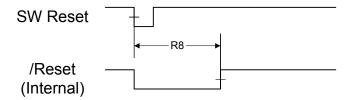
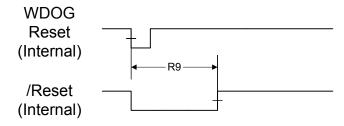


Figure 20-9. Watchdog Reset Timing



20.2.5 Sleep Modes

Table 20-15. Sleep Modes AC Characteristics^a

Parameter No	Parameter	Parameter Name	Min	Nom	Max	Unit
D1	t _{WAKE_S}	Time to wake from interrupt in sleep or deep-sleep mode, not using the PLL	-	-	7	system clocks
D2	t _{WAKE_PLL_S}	Time to wake from interrupt in sleep or deep-sleep mode when using the PLL	-	-	T _{READY}	ms

a. Values in this table assume the IOSC is the clock source during sleep or deep-sleep mode.

20.2.6 General-Purpose I/O (GPIO)

Note: All GPIOs are 5 V-tolerant.

Table 20-16. GPIO Characteristics

Parameter	Parameter Name	Condition	Min	Nom	Max	Unit
		2-mA drive		17	26	ns
	GPIO Rise Time (from 20% to 80%	4-mA drive		9	13	ns
GFION	of V _{DD})	8-mA drive	-	6	9	ns
		8-mA drive with slew rate control		10	12	ns
		2-mA drive		17	25	ns
	GPIO Fall Time (from 80% to 20%	4-mA drive		8	12	ns
GPIOF	of V _{DD})	8-mA drive	-	6	10	ns
		8-mA drive with slew rate control		11	13	ns

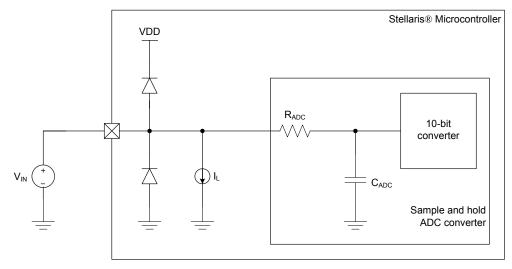
20.2.7 Analog-to-Digital Converter

Table 20-17. ADC Characteristics^a

Parameter	Parameter Name	Min	Nom	Max	Unit
	Maximum single-ended, full-scale analog input voltage	-	-	3.0	V
V _{ADCIN}	Minimum single-ended, full-scale analog input voltage	0.0	-	-	V
	Maximum differential, full-scale analog input voltage	-	-	1.5	V
	Minimum differential, full-scale analog input voltage	0.0	-	-	V
N	Resolution		10		bits
f _{ADC}	ADC internal clock frequency ^b	14	16	18	MHz
t _{ADCCONV}	Conversion time ^c		4		μs
f ADCCONV	Conversion rate ^c		250		k samples/s
t _{LT}	Latency from trigger to start of conversion	-	2	-	system clocks
IL	ADC input leakage	-	-	±3.0	μA
R _{ADC}	ADC equivalent resistance	-	-	10	kΩ
C _{ADC}	ADC equivalent capacitance	0.9	1.0	1.1	pF
EL	Integral nonlinearity error	-	-	±3	LSB
E _D	Differential nonlinearity error	-	-	±2	LSB
E _O	Offset error	-	-	+6 ^d	LSB
E _G	Full-scale gain error	-	-	±3	LSB
E _{TS}	Temperature sensor accuracy	-	-	±5	°C

a. The ADC reference voltage is 3.0 V. This reference voltage is internally generated from the 3.3 VDDA supply by a band gap circuit.

Figure 20-10. ADC Input Equivalency Diagram



b. The ADC must be clocked from the PLL or directly from an external clock source to operate properly.

c. The conversion time and rate scale from the specified number if the ADC internal clock frequency is any value other than 16 MHz.

d. The offset error listed above is the conversion result with 0 V applied to the ADC input.

Table 20-18. ADC Module Internal Reference Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{REFI}	Internal voltage reference for ADC	-	3.0	-	V
E _{IR}	Internal voltage reference error	-	-	±2.5	%

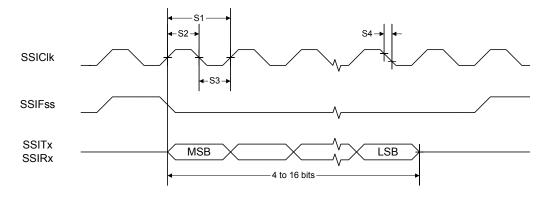
20.2.8 Synchronous Serial Interface (SSI)

Table 20-19. SSI Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
S1	t _{clk_per}	SSIClk cycle time	2	-	65024	system clocks
S2	t _{clk_high}	SSIClk high time	-	0.5	-	t clk_per
S3	t _{clk_low}	SSIC1k low time	-	0.5	-	t clk_per
S4	t _{clkrf}	SSIC1k rise/fall time ^a	-	6	10	ns
S5	t _{DMd}	Data from master valid delay time	0	-	1	system clocks
S6	t _{DMs}	Data from master setup time	1	-	-	system clocks
S7	t _{DMh}	Data from master hold time	2	-	-	system clocks
S8	t _{DSs}	Data from slave setup time	1	-	-	system clocks
S9	t _{DSh}	Data from slave hold time	2	-	-	system clocks

a. Note that the delays shown are using 8-mA drive strength.

Figure 20-11. SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing Measurement



SSICIK

SSIFss

SSIFss

SSIFx

MSB

LSB

SSIRx

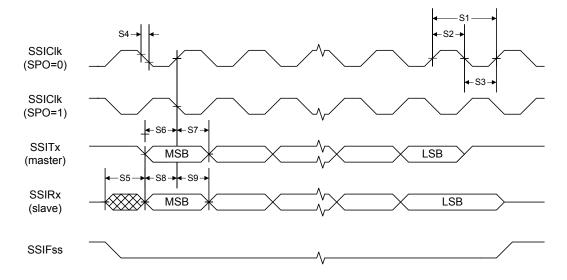
MSB

LSB

A to 16 bits output data

Figure 20-12. SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer





20.2.9 Inter-Integrated Circuit (I²C) Interface

Table 20-20. I²C Characteristics

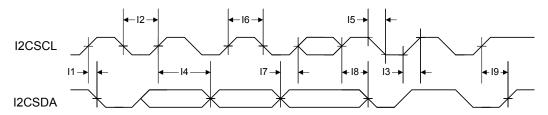
Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
I1 ^a	t _{SCH}	Start condition hold time	36	-	-	system clocks
I2 ^a	t _{LP}	Clock Low period	36	-	-	system clocks
I3 ^b	t _{SRT}	12CSCL/12CSDA rise time (V $_{IL}$ =0.5 V to V $_{IH}$ =2.4 V)	-	-	(see note b)	ns

Table 20-20. I ² C Characteristics (contin	ued)
-------------------------------------------------------	------

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
I4 ^a	t _{DH}	Data hold time	2	-	-	system clocks
I5 ^c	t _{SFT}	12CSCL/I2CSDA fall time (V $_{IH}$ =2.4 V to V $_{IL}$ =0.5 V)	-	9	10	ns
I6 ^a	t _{HT}	Clock High time	24	-	-	system clocks
I7 ^a	t _{DS}	Data setup time	18	-	-	system clocks
I8 ^a	t _{SCSR}	Start condition setup time (for repeated start condition only)	36	-	-	system clocks
I9 ^a	t _{SCS}	Stop condition setup time	24	-	-	system clocks

- a. Values depend on the value programmed into the TPR bit in the I²C Master Timer Period (I2CMTPR) register; a TPR programmed for the maximum I2CSCL frequency (TPR=0x2) results in a minimum output timing as shown in the table above. The I²C interface is designed to scale the actual data transition time to move it to the middle of the I2CSCL Low period. The actual position is affected by the value programmed into the TPR; however, the numbers given in the above values are minimum values.
- b. Because I2CSCL and I2CSDA are open-drain-type outputs, which the controller can only actively drive Low, the time I2CSCL or I2CSDA takes to reach a high level depends on external signal capacitance and pull-up resistor values.
- c. Specified at a nominal 50 pF load.

Figure 20-14. I²C Timing



20.2.10 Ethernet Controller

Table 20-21. 100BASE-TX Transmitter Characteristics^a

Parameter Name	Min	Nom	Max	Unit
Peak output amplitude	950	-	1050	mVpk
Output amplitude symmetry	98	-	102	%
Output overshoot	-	-	5	%
Rise/Fall time	3	-	5	ns
Rise/Fall time imbalance	-	-	500	ps
Duty cycle distortion	-	-	-	ps
Jitter	-	-	1.4	ns

a. Measured at the line side of the transformer.

Table 20-22. 100BASE-TX Transmitter Characteristics (informative)^a

Parameter Name	Min	Nom	Max	Unit
Return loss	16	-	-	dB
Open-circuit inductance	350	-	-	μH

a. The specifications in this table are included for information only. They are mainly a function of the external transformer and termination resistors used for measurements.

Table 20-23. 100BASE-TX Receiver Characteristics

Parameter Name	Min	Nom	Max	Unit
Signal detect assertion threshold	600	700	-	mVppd
Signal detect de-assertion threshold	350	425	-	mVppd
Differential input resistance	-	20	-	kΩ
Jitter tolerance (pk-pk)	4	-	-	ns
Baseline wander tracking	-75	-	+75	%
Signal detect assertion time	-	-	1000	μs
Signal detect de-assertion time	-	-	4	μs

Table 20-24. 10BASE-T Transmitter Characteristics^a

Parameter Name	Min	Nom	Max	Unit
Peak differential output signal	2.2	-	2.8	V
Harmonic content	27	-	-	dB
Link pulse width	-	100	-	ns
Start-of-idle pulse width	-	300	-	ns
		350		

a. The Manchester-encoded data pulses, the link pulse and the start-of-idle pulse are tested against the templates and using the procedures found in Clause 14 of *IEEE 802.3*.

Table 20-25. 10BASE-T Transmitter Characteristics (informative)^a

Parameter Name	Min	Nom	Max	Unit
Output return loss	15	-	-	dB
Output impedance balance	29-17log(f/10)	-	-	dB
Peak common-mode output voltage	-	-	50	mV
Common-mode rejection	-	-	100	mV
Common-mode rejection jitter	-	-	1	ns

a. The specifications in this table are included for information only. They are mainly a function of the external transformer and termination resistors used for measurements.

Table 20-26. 10BASE-T Receiver Characteristics

Parameter Name	Min	Nom	Max	Unit
Jitter tolerance (pk-pk)	30	-	-	ns
Input squelched threshold	500	600	700	mVppd
Differential input resistance	-	20	-	kΩ
Common-mode rejection	25	-	-	V

Table 20-27. Isolation Transformers^a

Name	Value	Condition
Turns ratio	1 CT : 1 CT	+/- 5%
Open-circuit inductance	350 uH (min)	@ 10 mV, 10 kHz
Leakage inductance	0.40 uH (max)	@ 1 MHz (min)
Inter-winding capacitance	25 pF (max)	
DC resistance	0.9 Ohm (max)	

Table 20-27. Isolation Transformers (continued)

Name	Value	Condition
Insertion loss	0.4 dB (typ)	0-65 MHz
HIPOT	1500	Vrms

a. Two simple 1:1 isolation transformers are required at the line interface. Transformers with integrated common-mode chokes are recommended for exceeding FCC requirements. This table gives the recommended line transformer characteristics.

Note: The 100Base-TX amplitude specifications assume a transformer loss of 0.4 dB. For the transmit line transformer with higher insertion losses, up to 1.2 dB of insertion loss can be compensated by selecting the appropriate setting in the Transmit Amplitude Selection (TXO) bits in the **MR19** register.

Table 20-28. Ethernet Reference Crystal^a

Name	Value	Condition
Frequency	25.00000	MHz
Frequency tolerance ^b	±50	PPM
Oscillation mode	Parallel resonance, fundamental mode	
Parameters at 25° C ±2° C; Drive level = 0.5 mW		
Drive level (typ)	50-100	μW
Shunt capacitance (max)	10	pF
Motional capacitance (min)	10	fF
Series resistance (max)	60	Ω
Spurious response (max)	> 5 dB below main within 500 kHz	

a. If the internal crystal oscillator is used, select a crystal that meets these specifications.

Figure 20-15. External XTLP Oscillator Characteristics

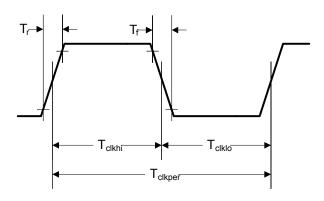


Table 20-29. External XTLP Oscillator Characteristics

Parameter Name	Symbol	Min	Nom	Max	Unit
XTLN Input Low Voltage	XTLN _{ILV}	-	1	0.8	-

b. This tolerance provides a guard band for temperature stability and aging drift.

Table 20-29. External XTLP Oscillator Characteristics (continued)

Parameter Name	Symbol	Min	Nom	Max	Unit
XTLP Frequency ^a	XTLP _f	-	25.0	-	-
XTLP Period ^b	T _{clkper}	-	40	-	-
XTLP Duty Cycle	XTLP _{DC}	40	-	60	%
		40		60	
Rise/Fall Time	T_r , T_f	-	-	4.0	ns
Absolute Jitter	T _{JITTER}	-	-	0.1	ns

a. IEEE 802.3 frequency tolerance ±50 ppm.

20.2.11 Analog Comparator

Table 20-30. Analog Comparator Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{OS}	Input offset voltage	-	±10	±25	mV
V _{CM}	Input common mode voltage range	0	-	V _{DD} -1.5	V
C _{MRR}	Common mode rejection ratio	50	-	-	dB
T _{RT}	Response time	-	-	1	μs
T _{MC}	Comparator mode change to Output Valid	-	-	10	μs

Table 20-31. Analog Comparator Voltage Reference Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
R _{HR}	Resolution high range	-	V _{DD} /31	-	LSB
R _{LR}	Resolution low range	-	V _{DD} /23	-	LSB
A _{HR}	Absolute accuracy high range	-	-	±1/2	LSB
A _{LR}	Absolute accuracy low range	-	-	±1/4	LSB

b. IEEE 802.3 frequency tolerance ±50 ppm.

A Serial Flash Loader

A.1 Serial Flash Loader

The Stellaris[®] serial flash loader is a preprogrammed flash-resident utility used to download code to the flash memory of a device without the use of a debug interface. The serial flash loader uses a simple packet interface to provide synchronous communication with the device. The flash loader runs off the crystal and does not enable the PLL, so its speed is determined by the crystal used. The two serial interfaces that can be used are the UART0 and SSI0 interfaces. For simplicity, both the data format and communication protocol are identical for both serial interfaces.

A.2 Interfaces

Once communication with the flash loader is established via one of the serial interfaces, that interface is used until the flash loader is reset or new code takes over. For example, once you start communicating using the SSI port, communications with the flash loader via the UART are disabled until the device is reset.

A.2.1 UART

The Universal Asynchronous Receivers/Transmitters (UART) communication uses a fixed serial format of 8 bits of data, no parity, and 1 stop bit. The baud rate used for communication is automatically detected by the flash loader and can be any valid baud rate supported by the host and the device. The auto detection sequence requires that the baud rate should be no more than 1/32 the crystal frequency of the board that is running the serial flash loader. This is actually the same as the hardware limitation for the maximum baud rate for any UART on a Stellaris device which is calculated as follows:

Max Baud Rate = System Clock Frequency / 16

In order to determine the baud rate, the serial flash loader needs to determine the relationship between its own crystal frequency and the baud rate. This is enough information for the flash loader to configure its UART to the same baud rate as the host. This automatic baud-rate detection allows the host to use any valid baud rate that it wants to communicate with the device.

The method used to perform this automatic synchronization relies on the host sending the flash loader two bytes that are both 0x55. This generates a series of pulses to the flash loader that it can use to calculate the ratios needed to program the UART to match the host's baud rate. After the host sends the pattern, it attempts to read back one byte of data from the UART. The flash loader returns the value of 0xCC to indicate successful detection of the baud rate. If this byte is not received after at least twice the time required to transfer the two bytes, the host can resend another pattern of 0x55, 0x55, and wait for the 0xCC byte again until the flash loader acknowledges that it has received a synchronization pattern correctly. For example, the time to wait for data back from the flash loader should be calculated as at least 2*(20(bits/sync)/baud rate (bits/sec)). For a baud rate of 115200, this time is 2*(20/115200) or 0.35 ms.

A.2.2 SSI

The Synchronous Serial Interface (SSI) port also uses a fixed serial format for communications, with the framing defined as Motorola format with SPH set to 1 and SPO set to 1. See "Frame Formats" on page 445 in the SSI chapter for more information on formats for this transfer protocol. Like the UART, this interface has hardware requirements that limit the maximum speed that the SSI clock can run. This allows the SSI clock to be at most 1/12 the crystal frequency of the board running

the flash loader. Since the host device is the master, the SSI on the flash loader device does not need to determine the clock as it is provided directly by the host.

A.3 Packet Handling

All communications, with the exception of the UART auto-baud, are done via defined packets that are acknowledged (ACK) or not acknowledged (NAK) by the devices. The packets use the same format for receiving and sending packets, including the method used to acknowledge successful or unsuccessful reception of a packet.

A.3.1 Packet Format

All packets sent and received from the device use the following byte-packed format.

```
struct
{
  unsigned char ucSize;
  unsigned char ucCheckSum;
  unsigned char Data[];
};
```

ucSize The first byte received holds the total size of the transfer including

the size and checksum bytes.

ucChecksum This holds a simple checksum of the bytes in the data buffer only.

The algorithm is Data[0]+Data[1]+...+ Data[ucSize-3].

Data This is the raw data intended for the device, which is formatted in

some form of command interface. There should be ucSize-2 bytes of data provided in this buffer to or from the device.

A.3.2 Sending Packets

The actual bytes of the packet can be sent individually or all at once; the only limitation is that commands that cause flash memory access should limit the download sizes to prevent losing bytes during flash programming. This limitation is discussed further in the section that describes the serial flash loader command, COMMAND_SEND_DATA (see "COMMAND_SEND_DATA (0x24)" on page 664).

Once the packet has been formatted correctly by the host, it should be sent out over the UART or SSI interface. Then the host should poll the UART or SSI interface for the first non-zero data returned from the device. The first non-zero byte will either be an ACK (0xCC) or a NAK (0x33) byte from the device indicating the packet was received successfully (ACK) or unsuccessfully (NAK). This does not indicate that the actual contents of the command issued in the data portion of the packet were valid, just that the packet was received correctly.

A.3.3 Receiving Packets

The flash loader sends a packet of data in the same format that it receives a packet. The flash loader may transfer leading zero data before the first actual byte of data is sent out. The first non-zero byte is the size of the packet followed by a checksum byte, and finally followed by the data itself. There is no break in the data after the first non-zero byte is sent from the flash loader. Once the device communicating with the flash loader receives all the bytes, it must either ACK or NAK the packet to indicate that the transmission was successful. The appropriate response after sending a NAK to the flash loader is to resend the command that failed and request the data again. If needed, the host may send leading zeros before sending down the ACK/NAK signal to the flash loader, as the

flash loader only accepts the first non-zero data as a valid response. This zero padding is needed by the SSI interface in order to receive data to or from the flash loader.

A.4 Commands

The next section defines the list of commands that can be sent to the flash loader. The first byte of the data should always be one of the defined commands, followed by data or parameters as determined by the command that is sent.

A.4.1 COMMAND_PING (0X20)

This command simply accepts the command and sets the global status to success. The format of the packet is as follows:

```
Byte[0] = 0x03;
Byte[1] = checksum(Byte[2]);
Byte[2] = COMMAND_PING;
```

The ping command has 3 bytes and the value for COMMAND_PING is 0x20 and the checksum of one byte is that same byte, making Byte[1] also 0x20. Since the ping command has no real return status, the receipt of an ACK can be interpreted as a successful ping to the flash loader.

A.4.2 COMMAND_GET_STATUS (0x23)

This command returns the status of the last command that was issued. Typically, this command should be sent after every command to ensure that the previous command was successful or to properly respond to a failure. The command requires one byte in the data of the packet and should be followed by reading a packet with one byte of data that contains a status code. The last step is to ACK or NAK the received data so the flash loader knows that the data has been read.

```
Byte[0] = 0x03
Byte[1] = checksum(Byte[2])
Byte[2] = COMMAND_GET_STATUS
```

A.4.3 COMMAND_DOWNLOAD (0x21)

This command is sent to the flash loader to indicate where to store data and how many bytes will be sent by the COMMAND_SEND_DATA commands that follow. The command consists of two 32-bit values that are both transferred MSB first. The first 32-bit value is the address to start programming data into, while the second is the 32-bit size of the data that will be sent. This command also triggers an erase of the full area to be programmed so this command takes longer than other commands. This results in a longer time to receive the ACK/NAK back from the board. This command should be followed by a COMMAND_GET_STATUS to ensure that the Program Address and Program size are valid for the device running the flash loader.

The format of the packet to send this command is a follows:

```
Byte[0] = 11
Byte[1] = checksum(Bytes[2:10])
Byte[2] = COMMAND_DOWNLOAD
Byte[3] = Program Address [31:24]
Byte[4] = Program Address [23:16]
Byte[5] = Program Address [15:8]
Byte[6] = Program Address [7:0]
Byte[7] = Program Size [31:24]
```

```
Byte[8] = Program Size [23:16]
Byte[9] = Program Size [15:8]
Byte[10] = Program Size [7:0]
```

A.4.4 COMMAND_SEND_DATA (0x24)

This command should only follow a COMMAND_DOWNLOAD command or another COMMAND_SEND_DATA command if more data is needed. Consecutive send data commands automatically increment address and continue programming from the previous location. The caller should limit transfers of data to a maximum 8 bytes of packet data to allow the flash to program successfully and not overflow input buffers of the serial interfaces. The command terminates programming once the number of bytes indicated by the COMMAND_DOWNLOAD command has been received. Each time this function is called it should be followed by a COMMAND_GET_STATUS to ensure that the data was successfully programmed into the flash. If the flash loader sends a NAK to this command, the flash loader does not increment the current address to allow retransmission of the previous data.

```
Byte[0] = 11
Byte[1] = checksum(Bytes[2:10])
Byte[2] = COMMAND_SEND_DATA
Byte[3] = Data[0]
Byte[4] = Data[1]
Byte[5] = Data[2]
Byte[6] = Data[3]
Byte[7] = Data[4]
Byte[8] = Data[5]
Byte[9] = Data[6]
Byte[10] = Data[7]
```

A.4.5 COMMAND_RUN (0x22)

This command is used to tell the flash loader to execute from the address passed as the parameter in this command. This command consists of a single 32-bit value that is interpreted as the address to execute. The 32-bit value is transmitted MSB first and the flash loader responds with an ACK signal back to the host device before actually executing the code at the given address. This allows the host to know that the command was received successfully and the code is now running.

```
Byte[0] = 7
Byte[1] = checksum(Bytes[2:6])
Byte[2] = COMMAND_RUN
Byte[3] = Execute Address[31:24]
Byte[4] = Execute Address[23:16]
Byte[5] = Execute Address[15:8]
Byte[6] = Execute Address[7:0]
```

A.4.6 COMMAND_RESET (0x25)

This command is used to tell the flash loader device to reset. This is useful when downloading a new image that overwrote the flash loader and wants to start from a full reset. Unlike the COMMAND_RUN command, this allows the initial stack pointer to be read by the hardware and set up for the new code. It can also be used to reset the flash loader if a critical error occurs and the host device wants to restart communication with the flash loader.

```
Byte[0] = 3
Byte[1] = checksum(Byte[2])
Byte[2] = COMMAND_RESET
```

The flash loader responds with an ACK signal back to the host device before actually executing the software reset to the device running the flash loader. This allows the host to know that the command was received successfully and the part will be reset.

B Register Quick Reference

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
The Co	rtex-M3	Process	or	l								l			
Ku, type i	₹/vv, , reset	- (see page	9 00)				D.4	λΤΑ							
								TA							
R1. type F	R/W reset	- (see page	÷ 56)												
, 9,60	211, , 10001	(occ page	, , ,				DA	ATA .							
								TA.							
R2, type F	R/W, , reset	- (see page	e 56)												
							DA	λΤΑ							
							DA	ATA .							
R3, type F	R/W, , reset	- (see page	e 56)												
								ATA .							
							DA	TA							
R4, type F	R/W, , reset	- (see page	9 56)												
								TA.							
DE time I	2/14/ =====	/222 222	FC)				DF	ATA							
rco, type i	√vv, , reset	- (see page	; 00)				D.4	TA.							
								TA							
R6. type F	R/W reset	- (see page	e 56)												
., 31	,,	()	,				DA	TA.							
							DA	ATA .							
R7, type F	R/W, , reset	- (see page	e 56)												
							DA	ATA .							
							DA	ATA .							
R8, type F	R/W, , reset	- (see page	e 56)												
								ATA							
							DA	ATA							
R9, type F	R/W, , reset	- (see page	9 56)												
								ATA ATA							
R10 type	R/W. rese	ot - (see pag	ne 56)				DF	un.							
, type	, , 1636	(occ pay	,				D4	λΤΑ							
								TA							
R11, type	R/W, , rese	t - (see pag	je 56)												
							DA	ATA .							
							DA	ATA .							
R12, type	R/W, , rese	t - (see pag	ge 56)												
								λΤΑ							
							DA	ATA							
SP, type F	R/W, , reset	- (see page	9 57)												
								iP iP							
IR type !	R/W recot	OYFFFF FF	FF (see pag	ne 58)				"							
Lix, type i	, , 16361	VALI 1.1°F	. i (see pa(gc 50)			J II	NK							
								NK							
PC, type I	R/W, , reset	: - (see page	e 59)												
			-				Р	С							
								C							

21	20	20	20	27	26	OF.	24	22	22	21	20	10	10	17	16
31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20	19	18	17	16 0
	e R/W, , rese				10		0	-	0	3				•	0
N	Z	C	V	Q	ICI	/ IT	THUMB								
			I / IT									ISR	NUM		
PRIMASH	K, type R/W,	, reset 0x	0000.0000 (see page 6	4)										
			,												
															PRIMASK
FAULTM	ASK, type R	/W, , reset	0x0000.000	00 (see pag	e 65)						-				
															FAULTMASI
BASEPR	I, type R/W,	, reset 0x0	000.0000 (s	see page 66	5)										
									BASEPRI						
CONTRO	L, type R/W	, , reset 0x	0000.0000	(see page 6	i7)										
														ASP	TMPL
Cortex	-M3 Peri _l	oherals													
Systen	n Timer (SysTick) Registe	ers											
Base 0x	E000.E000)													
STCTRL,	type R/W, c	ffset 0x01	0, reset 0x0	0000.0000											
															COUNT
													CLK_SRC	INTEN	ENABLE
STRELO	AD, type R/\	V, offset 0	x014, reset	0x0000.00	00										
											REL	.OAD			
							RELO	DAD							
STCURR	ENT, type R	/WC, offse	t 0x018, res	et 0x0000.	0000										
											CUR	RENT			
							CURF	RENT							
	-M3 Peri														
	I Vectore		upt Cont	roller (N	VIC) Reg	jisters									
	E000.E000														
EN0, type	e R/W, offse	t 0x100, re	set 0x0000.	.0000				_							
							IN								
	D.144 . 65						IN	l							
EN1, type	e R/W, offse	t 0x104, re	set uxuuuu.	.0000								I			
									IN	IT.					
DISO tun	e R/W, offse	t 0v180 m	seat Ov0000	10000						•••					
DiSU, typ	e K/VV, Olise	it 0x 100, 10	SSEL UXUUUU				IN	т							
							IN								
DIS1. tvn	e R/W, offse	of 0x184. re	eset 0x0000	.0000											
o., .yp															
									IN	JT		I			
PEND0, t	ype R/W, of	fset 0x200	, reset 0x00	000.0000											
-,-	,						IN	T							
							IN								
PEND1, t	ype R/W, of	fset 0x204	, reset 0x00	000.0000											
									IN	IT					
UNPEND	0, type R/W	offset 0x2	280, reset 0	x0000.0000)										
							IN	T							
							IN	T							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				×0000.0000					-						
								1	IN	NT		1			
ACTIVE0,	type RO, o	ffset 0x30	0, reset 0x0	0000.0000											
							11	NT							
							11	NT							
ACTIVE1,	type RO, o	ffset 0x30	4, reset 0x0	0000.0000											
									IN	IT.					
PRI0, type	R/W, offse	et 0x400, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI1, type	R/W, offse	et 0x404, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI2, type	R/W, offse	et 0x408, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI3, type	R/W, offse	et 0x40C, r	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI4, type	R/W, offse	et 0x410, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI5, type	R/W, offse	et 0x414, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI6, type	R/W, offse	et 0x418, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI7, type	R/W, offse	et 0x41C, r	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI8, type	R/W, offse	et 0x420, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI9, type	R/W, offse	et 0x424, re	eset 0x0000	0.0000											
	INTD								INTC						
	INTB								INTA						
PRI10, typ	oe R/W, offs	et 0x428,	reset 0x000	00.000											
	INTD								INTC						
	INTB								INTA						
SWTRIG,	type WO, o	ffset 0xF0	0, reset 0x0	0000.0000											
												IN	TID		
Cortex-	M3 Peri	herals													
	Control		(SCB) Re	egisters											
	E000.E000														
CPUID, ty	pe RO, offs	et 0xD00,	reset 0x411	1F.C231											
			II	MP					VA	AR.			С	ON	
					PAF	RTNO							R	EV	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	type R/W,	offset 0xD0													
NMISET			PENDSV	UNPENDSV	PENDSTSET	PENDSTCLR		ISRPRE	ISRPEND					VEC	PEND
		PEND		RETBASE								VEC	CACT		
VTABLE, 1	type R/W, c	ffset 0xD08	8, reset 0x(0000.0000											
		BASE							OFFSET						
			OFF	SET											
APINT, typ	oe R/W, offs	set 0xD0C,	reset 0xFA	05.0000											
							VECT	ΓΚΕΥ							
ENDIANESS						PRIGROUP	•						SYSRESREQ	VECTCLRACT	VECTRESE
SYSCTRL	, type R/W,	offset 0xD	10, reset 0	x0000.0000											
											SEVONPEND		SLEEPDEEP	SLEEPEXIT	
CFGCTRL	, type R/W	offset 0xD	14, reset 0	x0000.0000)										
						STKALIGN	BFHFNMIGN				DIV0	UNALIGNED		MAINPEND	BASETHE
SYSPRI1.	type R/W.	offset 0xD1	8, reset 0x	0000.0000		1						1			
· · · · ·	,		,						USAGE						
	BUS								MEM						
SVSDD12		offset 0xD1	C reset 0x	,0000 0000											
OTOTICE,	SVC	onset oxb i	0, 16361 07												
	300														
01/000010	. 504														
SYSPRI3,		offset 0xD2	U, reset Ux	0000.0000											
	TICK								PENDSV						
									DEBUG						
SYSHNDO	TRL, type	R/W, offset	0xD24, re:	set 0x0000.	0000										
													USAGE	BUS	MEM
SVC	BUSP	MEMP	USAGEP	TICK	PNDSV		MON	SVCA				USGA		BUSA	MEMA
FAULTSTA	AT, type R/\	N1C, offset	0xD28, re:	set 0x0000.	0000										
						DIV0	UNALIGN					NOCP	INVPC	INVSTAT	UNDEF
BFARV			BSTKE	BUSTKE	IMPRE	PRECISE	IBUS	MMARV			MSTKE	MUSTKE		DERR	IERR
HFAULTS'	TAT, type R	/W1C, offse	et 0xD2C, ı	eset 0x000	0.0000										
DBG	FORCED														
														VECT	
MMADDR	type R/W,	offset 0xD	34, reset -												
							AD	DR							
							AD								
FAULTAD	DR. type R	/W, offset 0	xD38. rese	t -											
	, -,	,					AD	DR							
							AD								
0	MO David						7.0								
	M3 Peri			_											
Memory	y Protec	tion Unit	(MPU)	Register	S										
	E000.E000														
MPUTYPE	, type RO,	offset 0xD9	90, reset 0	0000.0800											
											IRE	SION			
			DRE	GION											SEPARATI
MPUCTRL	, type R/W	, offset 0xD	94, reset 0	x0000.0000)										
													PRIVDEFEN	HFNMIENA	ENABLE
MPUNUMI	BER, type	R/W, offset	0xD98, res	et 0x0000.0	0000										
	, -, -, -,	, ,	,												
														NUMBER	
														NONDER	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUBASE	, type R/W	, offset 0xI	D9C, reset	0x0000.000	0										
							AD	DR				1			
					ADDR						VALID			REGION	
MPUBASE	1, type R/V	V, offset 0x	xDA4, reset	0x0000.00	00										
					4000		AD	DR			\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \			DEGLON	
					ADDR						VALID			REGION	
MPUBASE	2, type R/V	V, offset 0x	kDAC, rese	t 0x0000.00	000										
					ADDR		AL	DR			VALID			DECION	
MDUDACE		N -ff4 0:	vDD4 maga	. 0000 00							VALID			REGION	
MPUBASE	3, type R/V	v, orrset ux	kDB4, reset	UXUUUU.UU	100		A.D.	NDD.							
					ADDR		AL	DR			VALID			REGION	
MDIIATTD	tuno P/M	offeet Ov	OA0, reset 0	~000 000							VALID			KEGION	
WIFUATIK	, type K/vv,	Oliset UXL	XN		U	AP					TEX		S	С	В
				l RD		AF					IEA	SIZE	3	C	ENABLE
MDIIATTO	1 type P/M	V offeet Ov	DA8, reset		00							OIZL			LIVADEL
WII OAITIK	i, type iv	, onset ox	XN			AP					TEX		S	С	В
				l RD							TEX	SIZE	0	0	ENABLE
MPUATTR	2. type R/V	V. offset Ox	DB0, reset		00							0.22			
5711.114	_, ., po	., 0.1.001 02.	XN			AP					TEX		S	С	В
				I RD								SIZE			ENABLE
MPUATTR	3, type R/V	V, offset 0x	DB8, reset	0x0000.00	00										
			XN			AP					TEX		S	С	В
			SF	RD								SIZE			ENABLE
System	Control														
-	00F.E000														
DID0, type	RO, offset	0x000, res	set - (see pa	age 182)											
		VER									CLA	ASS			
			MA	JOR							MIN	NOR			
PBORCTL	, type R/W,	offset 0x0	30, reset 0:	x0000.7FFI) (see page	184)									
														BORIOR	
LDOPCTL,	, type R/W,	offset 0x0	34, reset 0:	×0000.0000	(see page	185)						•			
												VA	ADJ		
RIS, type F	RO, offset (0x050, rese	et 0x0000.0	000 (see pa	age 186)										
									PLLLRIS					BORRIS	
IMC, type I	R/W, offset	0x054, res	set 0x0000.	0000 (see p	age 187)										
									PLLLIM					BORIM	
MISC, type	e R/W1C, o	ffset 0x058	3, reset 0x0	000.0000 (see page 18	38)									
									PLLLMIS					BORMIS	
RESC, typ	e R/W, offs	et 0x05C,	reset - (see	page 189)											
											0141	MET	BOD	POP	EVE
D00 :	DAW "	4.0000	40.0===		105						SW	WDT	BOR	POR	EXT
KCC, type	K/W, offse	τ υχυ60, re	set 0x078E		page 190)		CDIV		LIGEOUS - TO		LIOFERS	I	DIAMAD"		
		DIMPEN		ACG		SY	SDIV	FAI	USESYSDIV		USEPWMDIV		PWMDIV	IOCODIC	MOCODIO
DI LOSO :		PWRDN		BYPASS	\		XI	TAL		08	CSRC			IOSCDIS	MOSCDIS
PLLCFG, t	ype KU, of	iset uxu64	, reset - (se	e page 194 I	•)										
						_							D		
						F							R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	oe R/W, offs	set 0x070, r	eset 0x078	30.2810 (see											
USERCC2		DWDDNO		DVDACCO		DIV2		I		OCCCDC2					
DOL BOL I	(050 to 10	PWRDN2	0.444	BYPASS2		407)				OSCSRC2					
DSLPCLK	(CFG, type	R/W, offset	0x144, res	set uxu78u.	· · ·				I			I			
					DSDIV	ORIDE				DSOSCSRC					
DID1 tun	e RO, offse	t 0×004 ros	net (200 n	200 100)						DSOSCSRC					
ыы, тур		ER	set - (see p	age 190)		λM					DAD	TNO			
	PINCOUNT				17	AIVI			TEMP			KG	ROHS	OL	JAL
	RO, offset		et 0x007F.0	1 002F (see n	age 200)			l					110110		
200, 1,700	,	- CACCO, 100		, (000 p	ugo 2007		SRA	MSZ							
								SHSZ							
DC1. type	RO, offset	0x010. res	et 0x0011.3	31BF (see r	page 201)										
., ., po	, =	,		(300)	3. =3.,						PWM				ADC
	MINS	YSDIV				MAXAI	DCSPD	MPU		TEMPSNS	PLL	WDT	SWO	SWD	JTAG
DC2, type	RO, offset	0x014, res	et 0x0307.	1013 (see p	age 203)	1		1				1	1	I.	
-				, ,		COMP1	COMP0						TIMER2	TIMER1	TIMER0
			I2C0								SSI0			UART1	UART0
DC3, type	RO, offset	0x018, res	et 0x8F07.	87C3 (see p	page 205)									1	
32KHZ				CCP3	CCP2	CCP1	CCP0						ADC2	ADC1	ADC0
PWMFAULT					C1PLUS	C1MINUS	C0O	C0PLUS	COMINUS					PWM1	PWM0
DC4, type	RO, offset	0x01C, res	set 0x5000.	007F (see p	page 207)										
	EPHY0		EMAC0												
									GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
RCGC0, t	ype R/W, of	ffset 0x100	, reset 0x00	0000040 (se	ee page 209	9)									
											PWM				ADC
						MAXAI	DCSPD					WDT			
SCGC0, t	ype R/W, of	fset 0x110,	reset 0x00	000040 (se	e page 211)									
											PWM				ADC
						MAXAI	DCSPD					WDT			
DCGC0, t	ype R/W, of	ffset 0x120	, reset 0x00	0000040 (se	ee page 213	3)									
											PWM				ADC
												WDT			
RCGC1, t	ype R/W, of	ffset 0x104	, reset 0x00	000000 (se	ee page 215								I	I	
			10.75			COMP1	COMP0				00:0		TIMER2	TIMER1	TIMER0
			12C0								SSI0			UART1	UART0
SCGC1, t	ype R/W, of	tset 0x114,	reset 0x00	0000000 (se	ee page 217		00:1-:								
			1200			COMP1	COMP0				0010		TIMER2	TIMER1	TIMER0
D0004 1	ma Dani	Stock Code 1	12C0	2000000 /	011) ·					SSI0			UART1	UART0
DCGC1, t	ype R/W, of	rrset Ux124	, reset 0x0(JUUUUUO (S6	ee page 219	COMP1	COMP0						TIMEDO	TIMEDA	TIMEDO
			I2C0			COMPT	COMPO				SSI0		TIMER2	TIMER1 UART1	UART0
DCCC2 4	uno B/M -	feet 0v400		000000 (**	ne poge 201)					3310			UARTI	UARIU
KUGUZ, I	ype R/W, of EPHY0	iset UX1U8,	EMAC0	 	ee page 221)									
	EFRIU		EIVIACU						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
SCGC2 4	ype R/W, of	feat Ov110	roset five	000000 (22	e nage 223)			GFIOG	GFIOF	GFIUE	GFIOD	GFIOC	GFIUB	GrIUA
30002, t	EPHY0	iset UXTI8,	EMAC0	 	e paye 223	,									
	L. 1110		LIVIACO						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
DCGC2 +	ype R/W, of	ffset 0v129	reset five	0000000 (54	e nace 225	5)			J 100	51 101	0. IOL	01100	J. 100	OI 100	JI IUA
DOGOZ, I	EPHY0	1361 UX 120	EMAC0		Le page 220	·)									
	LEHIU		LIVIACU						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
									0, 100	0, 101	01 10L	J 0, 10D	0, 100	0, 100	OI IOA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCR0, t	ype R/W, of □	fset 0x040,	reset 0x00	000000 (se	e page 227))									
											PWM	WDT			ADC
CDCD4 4	una DAM ad	F= =4 0×044		000000 (22		\						WDT			
SRCR1, t	ype R/W, of	TSET UXU44,	reset uxuu	ooooo (se	e page 228		COMPO						TIMEDO	TIMED4	TIMEDO
			12C0			COMP1	COMP0				SSI0		TIMER2	TIMER1 UART1	TIMER0 UART0
SDCD2 to	ype R/W, of	feat 0v048		000000 (60	nage 230)	\					0010			OARTI	OAICIO
ORORZ, t	EPHY0	1361 02040,	EMAC0	(30	page 200,	,									
	Little		LIVIAGO						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
Interna	l Memor	v										l			
Flash N	/lemory (400F.D000	Control I	Register	s (Flash	Control	Offset)									
FMA, type	e R/W, offse	et 0x000, re	set 0x0000	.0000											
															OFFSET
							OFF	SET							
FMD, type	e R/W, offse	et 0x004, re	set 0x0000	.0000											
							DA	TΑ							
							DA	·ΤΑ							
FMC, type	e R/W, offse	et 0x008, re	set 0x0000	.0000											
							WR	KEY							
												COMT	MERASE	ERASE	WRITE
FCRIS, ty	pe RO, offs	et 0x00C, r	eset 0x000	0.0000											
														PRIS	ARIS
FCIM, typ	e R/W, offs	et 0x010, re	eset 0x0000	.0000											
														PMASK	AMASK
FCMISC,	type R/W10	, offset uxi	014, reset 0	X0000.0000)							1			
														PMISC	AMISC
														FIVIISC	AIVIIGC
	I Memor	-					• •								
	/lemory l 400F.E000		on Regis	ters (Sys	stem Co	ntrol Of	fset)								
	type R/W,		0 roost 0v2	.4											
USLUKL,	type K/vv,	UIISEL UX 14	o, reset ox	, ı											
											US	I SEC			
FMPRE0.	type R/W,	offset 0x13	0 and 0x200), reset 0xF	FFF.FFFF										
,	· · · · · · · · · · · · · · · · · · ·						READ I	ENABLE							
								ENABLE							
FMPPE0,	type R/W,	offset 0x134	4 and 0x400), reset 0xF	FFF.FFFF										
							PROG_	ENABLE							
							PROG_	ENABLE							
USER_DE	3G, type R/	W, offset 0x	1D0, reset	0xFFFF.FF	FE										
NW								DATA							
						DA	ATA							DBG1	DBG0
USER_RE	EG0, type R	/W, offset 0	x1E0, rese	t 0xFFFF.FI	FF										
NW								DATA							
							DA	TA							
USER_RE	EG1, type R	/W, offset 0	x1E4, rese	t 0xFFFF.FI	FF										
NW								DATA							
							DA	λTA							

15	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
READ_ENABLE READ_ENABLE FMPRE2, type RW, offset 0x205, reset 0x0000.0000 READ_ENABLE READ																0
READ_ENABLE READ_	FMPRE1,	type R/W,	offset 0x20	4, reset 0x	0000.FFFF											
MRPE2, type RW, offset 0x200, reset 0x0000.0000								READ_	ENABLE							
READ_ENABLE READ_E								READ_	ENABLE							
READ_ENABLE PMPRE3, type RW, offset 0x404, reset 0x0000,0000 READ_ENABLE READ_ENABLE READ_ENABLE READ_ENABLE PROS_ENABLE ###################################	FMPRE2,	type R/W,	offset 0x20	8, reset 0x	0000.0000											
FMPRE3, type R/W, offset 0x404, reset 0x0000.0000 READ_ENABLE READ_ENABLE PROC_ENABLE PROC_E								READ_	ENABLE							
READ_ENABLE READ_ENABLE READ_ENABLE PROG_ENABLE PROG_E								READ_	ENABLE							
READ_ENABLE FMPPE1, type R/W, offset 0x404, reset 0x0000.FFFF PROG_ENABLE PROG_E	FMPRE3,	type R/W,	offset 0x20	C, reset 0x	0000.0000											
FMPPE1, type R/W, offset 0x404, reset 0x0000.FFFF PROG_ENABLE PROG_ENABLE FMPPE2, type R/W, offset 0x408, reset 0x0000.0000 PROG_ENABLE FMPPE3, type R/W, offset 0x402, reset 0x0000.0000 PROG_ENABLE Seneral-Purpose Input/Outputs (GPIOs) PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE DATA PROG_ENABLE PROG_ENABLE DIRE PROG_ENABLE PROG_E								READ_	ENABLE							
PROG_ENABLE PROG_E								READ_	ENABLE							
PROG_ENABLE PROG_	MPPE1,	type R/W,	offset 0x40	4, reset 0x0	0000.FFFF											
FMPPE2, type R/W, offset 0x408, reset 0x0000.0000 PROG_ENABLE PROG_ENABLE FMPPE3, type R/W, offset 0x40C, reset 0x0000.0000 PROG_ENABLE								PROG_	ENABLE							
PROG_ENABLE PROG_E								PROG_	ENABLE							
PROG_ENABLE FMPPE3, type R/W, offset 0x40C, reset 0x0000.0000 PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE PROG_ENABLE General-Purpose Input/Outputs (GPIOs) 3PIO Port A base: 0x4000.4000 3PIO Port B base: 0x4000.5000 3PIO Port D base: 0x4000.7000 3PIO Port D base: 0x4000.7000 3PIO Port D base: 0x4000.4000 3PIO Port F base: 0x4000.5000 3PIODATA, type R/W, offset 0x4004, reset 0x0000.0000 (see page 271) DATA 3PIODIR, type R/W, offset 0x404, reset 0x0000.0000 (see page 272) JDIR 3PIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IBE 3PIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 275) IBE 3PIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 276) IBE 3PIOIS, type R/W, offset 0x414, reset 0x0000.0000 (see page 277) IBE 3PIOIS, type R/W, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIONIS, type R/W, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIONIS, type R/W, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIONIS, type R/W, offset 0x414, reset 0x0000.0000 (see page 278) MIS 3PIONIS, type R/O, offset 0x414, reset 0x0000.0000 (see page 279) MIS	MPPE2,	type R/W,	offset 0x40	8, reset 0x0	0000.0000											
PROS_ENABLE PROS_ENABLE PROS_ENABLE General-Purpose Input/Outputs (GPIOs) SPIO POT A base: 0x4000.4000 SPIO POT A base: 0x4000.5000 SPIO POT C base: 0x4000.5000 SPIO POT C base: 0x4000.7000 SPIO POT C base: 0x4000.7000 SPIO POT BOT E base: 0x4000.4000 SPIO POT G base: 0x4000.7000 SPIO POT G base: 0x40000.7000 SPIO POT G base:																
PROG_ENABLE PROG_ENABLE General-Purpose Input/Outputs (GPIOs) GPIO Port A base: 0x4000 4000 GPIO Port B base: 0x4000 5000 GPIO Port D base: 0x4000 5000 GPIO Port D base: 0x4000 7000 GPIO Port D base: 0x4000 7000 GPIO Port D base: 0x4000 7000 GPIO Port G base: 0x4000 7000 GPIO Port G base: 0x4000 7000 GPIO Port G base: 0x4000 7000 GPIODIR, type RIW, offset 0x000, reset 0x0000 0000 (see page 271) DATA GPIODIR, type RIW, offset 0x400, reset 0x0000 0000 (see page 272) IS GPIOIBE, type RIW, offset 0x404, reset 0x0000 0000 (see page 273) IS GPIOIBE, type RIW, offset 0x400, reset 0x0000 0000 (see page 274) IBE GPIOIEV, type RIW, offset 0x410, reset 0x0000 0000 (see page 275) IEV GPIOIN, type RIW, offset 0x411, reset 0x0000 0000 (see page 277) RIS GPIOIN, type RIW, offset 0x414, reset 0x0000 0000 (see page 277) RIS GPIOIN, type RIW, offset 0x414, reset 0x0000 0000 (see page 278) MIS GPIOIR, type RIV, offset 0x414, reset 0x40000 0000 (see page 278) MIS GPIOIR, type WIC, offset 0x416, reset 0x0000 0000 (see page 279)								PROG_	ENABLE							
### PROG_ENABLE General-Purpose Input/Outputs (GPIOS) \$PIO Port & base: 0x4000 4000 \$PIO Port & base: 0x4000 5000 \$PIO Port & base: 0x4002 5000 \$PIO Port & base: 0x4002 5000 \$PIO Port & base: 0x4002 5000 \$PIODATA, type R/W, offset 0x400, reset 0x000, reset 0x0000, 0000 (see page 271) DATA \$PIODATA, type R/W, offset 0x404, reset 0x0000, 0000 (see page 272) ### PIODATA \$PIODIR, type R/W, offset 0x404, reset 0x0000, 0000 (see page 273) ### BIOS \$PIOIS, type R/W, offset 0x406, reset 0x0000, 0000 (see page 274) ### BIOS \$PIOISE, type R/W, offset 0x407, reset 0x0000, 0000 (see page 276) ### BIOS \$PIOIS, type R/W, offset 0x410, reset 0x0000, 0000 (see page 276) ### BIOS \$PIOIS, type R/W, offset 0x414, reset 0x0000, 0000 (see page 277) ### BIOS \$PIOIS, type R/W, offset 0x414, reset 0x0000, 0000 (see page 278) ### BIOS \$PIOMIS, type R/W, offset 0x414, reset 0x0000, 0000 (see page 278) ### BIOS ### BI	MPPE3,	type R/W,	offset 0x40	C, reset 0x	0000.0000											
General-Purpose Input/Outputs (GPIOs) GPIO Port A base: 0x4000.4000 GPIO Port A base: 0x4000.5000 GPIO Port C base: 0x4000.6000 GPIO Port C base: 0x4000.6000 GPIO Port C base: 0x4000.7000 GPIO Port Dase: 0x4000.24000 GPIO Port E base: 0x4002.4000 GPIO Port E base: 0x4002.4000 GPIO Port E base: 0x4002.4000 GPIO Port E base: 0x4002.5000 GPIO Port G base: 0x4002.6000 GPIODATA, type RIW, offset 0x400, reset 0x0000.0000 (see page 271) DATA DATA SPIODIR, type RIW, offset 0x400, reset 0x0000.0000 (see page 272) IS GPIOIS, type RIW, offset 0x404, reset 0x0000.0000 (see page 273) IS GPIOIS, type RIW, offset 0x404, reset 0x0000.0000 (see page 274) IBE GPIOIS, type RIW, offset 0x400, reset 0x0000.0000 (see page 275) IEV GPIOIN, type RIW, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIOINS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) IME GPIOINS, type RO, offset 0x414, reset 0x0000.0000 (see page 278) MIS GPIOINS, type RO, offset 0x411, reset 0x0000.0000 (see page 278) MIS GPIOINS, type RO, offset 0x411, reset 0x0000.0000 (see page 279)																
GPIO PORT & Dasse: 0x44000.40000 GPIO PORT & Dasse: 0x4000.50000 GPIO PORT & Dasse: 0x4000.50000 GPIO PORT & Dasse: 0x4000.50000 GPIO PORT & Dasse: 0x40002.40000 GPIO PORT & Dasse: 0x40002.40000 GPIO PORT & Dasse: 0x40002.40000 GPIO PORT & Dasse: 0x40002.50000 GPIODATA, type R/W, offset 0x4000, reset 0x0000.0000 (see page 271) DATA BPIODIR, type R/W, offset 0x4004, reset 0x0000.0000 (see page 272) DIR GPIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IBE GPIOIEL, type R/W, offset 0x406, reset 0x0000.0000 (see page 275) IBE GPIOIEN, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIOIN, type R/W, offset 0x4114, reset 0x0000.0000 (see page 277) RIS GPIOINS, type RO, offset 0x414, reset 0x0000.0000 (see page 278) MIS GPIOINS, type RO, offset 0x416, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W/C, offset 0x416, reset 0x0000.0000 (see page 279)								PROG_	ENABLE							
DATA SPIODIR, type R/W, offset 0x400, reset 0x0000.0000 (see page 272) DIR SPIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS SPIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE SPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV SPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME SPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS SPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS SPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIO Po	ort F base: ort G base	0x4002.5 0x4002.6	5000 5000		0 /	.074)									
GPIODIR, type R/W, offset 0x400, reset 0x0000.0000 (see page 272) DIR GPIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS GPIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE GPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIODAT	A, type R/V	V, offset 0x	000, reset (0x0000.000	0 (see page	271)						1			
GPIODIR, type R/W, offset 0x400, reset 0x0000.0000 (see page 272) DIR GPIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS GPIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE GPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)																
DIR SPIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS SPIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE SPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV SPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME SPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS SPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS SPIORIS, type RO, offset 0x416, reset 0x0000.0000 (see page 279)	anionin		- 554 045	20 4 0	0000 0000	(70)					L	PAIA			
3PIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS 3PIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE 3PIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV 3PIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME 3PIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS 3PIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	3PIODIK	type R/w,	Offset UX4U	Ju, reset ux	1	(see page 2	(72)						1			
3PIOIS, type R/W, offset 0x404, reset 0x0000.0000 (see page 273) IS 3PIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE 3PIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV 3PIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME 3PIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS 3PIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)													DIR			
IS GPIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE GPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	SPIOIS t	vne R/W of	fset 0x404	reset 0x00) 100 0000 (s	ee nage 27	3)		<u> </u>							
3PIOIBE, type R/W, offset 0x408, reset 0x0000.0000 (see page 274) IBE SPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV SPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME SPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS SPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS SPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	, .	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		, 10001 0.00		oo pago 2.										
GPIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)													IS IS			
3PIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV 3PIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME 3PIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS 3PIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIOIBE,	type R/W,	offset 0x40	08, reset 0x	0000.0000	(see page 2	74)									
3PIOIEV, type R/W, offset 0x40C, reset 0x0000.0000 (see page 275) IEV 3PIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME 3PIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS 3PIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS 3PIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)																
GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)													IBE			
GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIOIEV,	type R/W,	offset 0x40	C, reset 0x	0000.0000	(see page 2	75)									
GPIOIM, type R/W, offset 0x410, reset 0x0000.0000 (see page 276) IME GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)																
GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)												I	IEV			
GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIOIM, 1	ype R/W, o	ffset 0x410	, reset 0x0	000.000 (s	ee page 27	6)									
GPIORIS, type RO, offset 0x414, reset 0x0000.0000 (see page 277) RIS GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)																
GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)												I	ME			
GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIORIS	type RO, c	ffset 0x414	4, reset 0x0	000.0000 (see page 27	77)									
GPIOMIS, type RO, offset 0x418, reset 0x0000.0000 (see page 278) MIS GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)																
GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	CDICANO	tura 50		0	2000 0000 /	ns 0	70)		<u> </u>				KID			
GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)	GPIUMIS	, type RO, d	orset UX41	o, reset 0x0	,000.0000 (s	see page 27	r 6)									
GPIOICR, type W1C, offset 0x41C, reset 0x0000.0000 (see page 279)													MIS			
	SDICICE	type W4C	offeet 0v4	1C reset 0	×0000 0000	(see page	270)					'	VIIO			
	SCIOICK	type wit,	JIISEL UX4	, reset U		(see page	213)									
ic													IC			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOAFS	EL, type R/\	N, offset 0	x420, reset	t - (see pag	e 280)							ļ			
	, ,,,,	,	,		,										
											AF	SEL			
CDIODDS	R, type R/W	L offeet Ov	500 reset		E (see pag	no 282)		l							
OI IODINE	it, type it it	, onset ox	500, 16361		(SCC pag	202)									
											Di	7) (0			
											DI	RV2			
GPIODR4	R, type R/W	, offset 0x	504, reset	0x0000.000	0 (see pag	e 283)		1							
											DI	RV4			
GPIODR8	R, type R/W	, offset 0x	508, reset	0x0000.000	0 (see pag	e 284)									
											DI	RV8			
GPIOODF	R, type R/W,	offset 0x5	OC, reset 0	x0000.000	(see page	285)									
											0	DE .			
GPIOPUE	R, type R/W,	offeet Ove	10 reset	(see nage ?	286)			I							
SFIOPUR	, type R/W,	Oliset UXS	10, 16561 -	(see page 2	.00)										
												UE			
											P	UE			
GPIOPDR	R, type R/W,	offset 0x5	14, reset 0:	x0000.0000	(see page	287)									
											Р	DE			
GPIOSLR	, type R/W,	offset 0x5	18, reset 0	k0000.0000	(see page	288)									
											S	RL			
GPIODEN	I, type R/W,	offset 0x5	1C. reset -	(see page 2	289)										
	, .,,,		,												
											D	I EN			
CDIOL OC	Y tune D/M	/ affa at 0:	F20 #224	00000 000	14 /222 222	200)									
GPIOLOC	K, type R/W	, onset ux	520, reset	UXUUUU.UUL	r (see pag	le 290)		.014							
								CK							
							LC	CK							
GPIOCR,	type -, offse	et 0x524, re	eset - (see	page 291)											
											(CR			
GPIOPeri	phID4, type	RO, offset	t 0xFD0, re	set 0x0000	.0000 (see	page 293)									
											Р	ID4			
GPIOPeri	phID5, type	RO. offset	t 0xFD4. re	set 0x0000	.0000 (see	page 294)		1							
	, ., po	, 550	2.,10		111 (000	30.)									
											D	ID5			
CDIOS	nhIDO 1	DO -7	OVED C		0000 /	name 007'		L			P				
GPIOPeri	phID6, type	KU, offset	UXFD8, re	set ux0000	.uuuu (see	page 295)									
											Р	ID6			
GPIOPeri	phID7, type	RO, offset	t 0xFDC, re	set 0x0000	.0000 (see	page 296)									
											Р	ID7			
GPIOPeri	phID0, type	RO, offset	t 0xFE0, re	set 0x0000	.0061 (see	page 297)									
												ID0			
ODIOD- '	ahlD4 5	DO -#- 1	10vFF4		0000 /	2000					Р	ID0			
GPIOPeri	phID1, type	RO, offset	t 0xFE4, re	set 0x0000	.0000 (see	page 298)					r	100			
GPIOPeri	phID1, type	RO, offset	t 0xFE4, re	set 0x0000	.0000 (see	page 298)						IDU			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOPeripl	hID2, type	RO, offset	0xFE8, res	set 0x0000.	.0018 (see p	page 299)						1			
											DI	D2			
CDIODarial	hID2 turns	DO offeet	0.4550		0004 (222	200					- FI	DZ .			
GPIOPeripl	nibs, type	KO, onsei	UXFEC, re	Set uxuuuu 	.0001 (see	page 300)									
											DI	D3			
CDIODCOIII	IDO tuno l	O offeet	VEEU roo	 	000 (222 2	201)					FI	D3			
GPIOPCelli	ibu, type i	to, onset	JAFFU, 1656		(See pa	age 301)									
											CI	D0			
GPIOPCelli	ID1 type I	O offect (VEE4 rose	 	0E0 (see po	age 303)									
GFIOFCEIII	ib i, type i	(O, Oliset (7, 1050		oi o (see pa	ige 302)									
											CI	D1			
GPIOPCelli	ID2 type I	O offect (VEES roce	 	005 (see pa	ae 303)									
GFIOFCEIII	IDZ, type i	(O, Oliset (JAI 1 0, 1656		oos (see pa	ige 303)									
											CI	D2			
GPIOPCelli	ID3 type I	RO offeat	XFFC res	et Oxonon n	0R1 (see n	age 304)									
OI IOF Gelli		CO, OHSEL (, , i e s		(See p	age 504)									
											CI	D3			
0	D		_												
General- Timer0 ba: Timer1 ba: Timer2 ba:	se: 0x40 se: 0x40	03.0000 03.1000	•												
GPTMCFG,	type R/W	, offset 0x0	000, reset 0	x0000.000	0 (see page	317)									
														GPTMCFG	
GPTMTAME	R, type R/	W, offset 0:	x004, reset	0x0000.00	00 (see pag	je 318)									
												TAAMS	TACMR	TAI	MR
GPTMTBM	R, type R/	W, offset 0	x008, reset	0x0000.00	00 (see pag	je 320)									
												TBAMS	TBCMR	ТВ	MR
GPTMCTL,	type R/W	offset 0x0	0C, reset 0	x0000.000	(see page	322)									
	TBPWML	TBOTE		TBE	VENT	TBSTALL	TBEN		TAPWML	TAOTE	RTCEN	TAE	/ENT	TASTALL	TAEN
GPTMIMR,	type R/W	offset 0x0	18, reset 0:	x0000.0000	(see page	325)									
					CBEIM	СВМІМ	ТВТОІМ					RTCIM	CAEIM	CAMIM	TATOIM
GPTMRIS, 1	type RO,	offset 0x01	C, reset 0x	0000.0000	(see page 3	327)									
					CBERIS	CBMRIS	TBTORIS					RTCRIS	CAERIS	CAMRIS	TATORIS
GPTMMIS,	type RO.	offset 0x02	0, reset 0x	0000.0000	(see page 3	328)									
	'														
					CBEMIS	CBMMIS	TBTOMIS					RTCMIS	CAEMIS	CAMMIS	TATOMIS
GPTMICR, 1	type W1C	, offset 0x0)24, reset 0	x0000.0000		329)									
-,						,									
					CBECINT	CBMCINT	TBTOCINT					RTCCINT	CAECINT	CAMCINT	TATOCINT
GPTMTAILE	R, type R/	W, offset 0	x028. reset	0xFFFF.FF											
	, -, ,, ,	,	,		. (200 pu	J · /	TAII	_RH							
							TAII								
GPTMTBILI	R. type R/	W. offset N	x02C, rese	t 0x0000 FF	FFF (see na	ige 332)									
	, ., po 10	, 0.1001 0			(See pa										
							TBI	l RI							
							וטו	\-							

	I	I		T											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPTMTAN	MATCHR, ty	pe R/W, of	ffset 0x030	, reset 0xFl	FFF.FFFF (see page 33	33)								
							TAN	ИRH							
							TAI	MRL							
GPTMTBN	//ATCHR, ty	pe R/W, of	ffset 0x034	, reset 0x0	000.FFFF (s	see page 33	34)								
							TBI	I ИRL							
GPTMTAP	P type P/	N offeet O	x038, reset	0×0000 00	00 (see nac	10 335)									
OI TIMITAL	it, type it	, onset o	X000, 1636t		(see pag	JC 333)									
											ТА	DOD.			
											IA	PSR			
GPTMTBP	R, type R/	W, offset 0	x03C, reset	t 0x0000.00	000 (see pa	ge 336)		1				1			
											ТВ	PSR			
GPTMTAP	MR, type F	R/W, offset	0x040, res	et 0x0000.0	0000 (see pa	age 337)									
											TAP	SMR			
GPTMTBP	PMR, type F	R/W, offset	0x044, res	et 0x0000.0	0000 (see p	age 338)									
											TBP	I PSMR			
GPTMT4P	type RO	offset Ovo	48, reset 0	(FFFF FFF	(see page	339)		1							
J. IMIAN	., ., pe NO,	SHOOL UAU	.5, 16361 07		(occ page	300)	T4	RH							
							IA	.RL							
GPTMTBR	R, type RO,	offset 0x0	4C, reset 0	x0000.FFF	F (see page	340)		1				1			
							TB	RL							
	log Time 1000.0000														
WDTLOAD	D, type R/W	l, offset 0x	000, reset (xFFFF.FF	F (see pag	e 345)									
							WDT	Load							
							WDT	Load							
WDTVALU	JE, type RC), offset 0x	004, reset	0xFFFF.FF	FF (see pac	ie 346)									
							WDT	Value							
								Value							
WDTCTI	tuno P/M	offeet OvOC	08, reset 0x	0000 0000	(see page 3	247)									
WDICIL,	type R/vv,	Uniset uxut	Jo, reset ux	1	(see page c) 4 7)									
														DESEN	13.1753.1
														RESEN	INTEN
WDTICR, t	type WO, o	ffset 0x00	C, reset - (s	see page 34	8)										
								IntClr							
							WDT	IntClr							
WDTRIS, t	type RO, o	fset 0x010), reset 0x0	000.0000 (s	see page 34	9)									
															WDTRIS
WDTMIS. 1	type RO, o	ffset 0x014	1, reset 0x0	000.0000 (s	see page 35	50)									
,	-, -,-			(, 3:	,									
															WDTMIS
WDTTEST	tune Date	offeet Or 1	110 #5==4.0	×0000 0000	1 (000 000	251)									11D TIVIIO
MDI IE91	, type K/W	JIISEL UX4	118, reset 0		, (see page	JU1)									
							ОТ:::								
							STALL								
WDTLOCK	K, type R/W	l, offset 0x	C00, reset	0x0000.000	00 (see pag	e 352)									
							WDT	Lock							
							WDT	Lock							
WDTPerip	hID4, type	RO, offset	0xFD0, res	set 0x0000.	0000 (see p	page 353)									
											PI	I ID4			
											• • •				

			T												T
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTPerip	hID5, type I	RO, offset	0xFD4, res	et 0x0000.0	0000 (see p	age 354)		1							
											PI	D5			
WDTPerip	hID6, type I	RO, offset	0xFD8, res	et 0x0000.0	0000 (see p	age 355)									
											PI	D6			
WDTPerip	hID7, type I	RO, offset	0xFDC, res	set 0x0000.	0000 (see p	age 356)									
											PI	D7			
WDTPerip	hID0, type I	RO, offset	0xFE0, res	et 0x0000.0	0005 (see p	age 357)									
						,									
											PI	D0			
WDTPerin	hID1, type I	RO offset	OxFF4 res	et 0x0000 (1018 (see n	age 358)									
TTD II CIIP	1115 1, type 1	110, 011001	- UXI E-4, 100		3010 (300 p	age occ)									
											DI	 D1			
14/DED :						050)					F1	U 1			
WDIPerip	hID2, type I	KU, offset	uxr⊑ö, res	et 0x0000.0	see p) ארטע	age 359)									
											PI	D2			
WDTPerip	hID3, type I	RO, offset	0xFEC, res	et 0x0000.	0001 (see p	age 360)									
											PI	D3			
WDTPCell	IID0, type R	O, offset 0	0xFF0, rese	t 0x0000.00	00D (see pa	ge 361)									
											CI	D0			
WDTPCell	IID1, type R	O, offset 0	0xFF4, rese	t 0x0000.00	F0 (see pa	ge 362)									
						- ,									
											CI	I D1			
WDTPCell	IID2, type R	O offset (TYFF8 rese	t 0×0000 00	105 (see na	ne 363)									
		-,	J. 1 0, 1000		(000 pa;	, ,									
											CI	D2			
WDTDO-II	UDO 4 D	0 - 55 4 6		4.00000	0D4 (004)		<u> </u>			Ci	DZ			
WDTPCell	IID3, type R	O, onset u	JXFFC, rese	t uxuuuu.u	ивт (see pa	ige 364)						I			
											CI	D3			
Analog-	-to-Digita	al Conv	erter (AD	IC)											
Base 0x4	1003.8000														
ADCACTS	SS, type R/W	V, offset 0	x000, reset	0x0000.000	00 (see pag	e 375)									
												ASEN3	ASEN2	ASEN1	ASEN0
ADCRIS, t	type RO, off	set 0x004	, reset 0x00)00.0000 (se	ee page 376	6)									_
												INR3	INR2	INR1	INR0
ADCIM. tv	pe R/W, offs	set 0x008	reset 0x00		e page 377	')									
, ty	, , , Oli	- 3. 5.000,	,	13.5550 (30	- page off	,									
												MASK3	MASK2	MASK1	MASK0
ADOIGO 1	nuna Danida		000	00000 000	0 /005 ==	270)						INIMONO	IVIAGRZ	IVIAORI	INICOLU
ADCISC, t	type R/W1C	, omset ux	.ooc, reset	UXUUUU.UU0	u (see page	: 3/8)									
												IN3	IN2	IN1	IN0
ADCOSTA	AT, type R/W	/1C, offset	t 0x010, res	et 0x0000.0	0000 (see pa	age 379)									
												OV3	OV2	OV1	OV0
ADCEMUX	X, type R/W,	offset 0x	014, reset 0	x0000.0000	(see page	380)									-
	EM	13			FN	/ И2			EN	Л1			FI	M0	
	_IV														

24	20	20	20	07	26	25	24	22	22	24	20	10	10	17	16
31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19	18	17	16 0
	AT, type R/W				-		ŭ	'			,			'	
	, .,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,	,		(000)	lago oco,									
												UV3	UV2	UV1	UV0
ADCSSPF	RI, type R/W	, offset 0x0)20, reset (Dx0000.321) (see pag	e 384)						1			
		SS	33			S	S2			S	31			SS	30
ADCPSSI	, type WO,	offset 0x02	8, reset - (s	see page 38	6)										
												SS3	SS2	SS1	SS0
ADCSAC,	type R/W,	offset 0x03	0, reset 0x	0000.0000 (see page	387)									
														AVG	
ADCSSMI	UX0, type R			et 0x0000.00	000 (see p										
		MU					JX6			MU				MU	
4 DOCCC	FI.O. 4: -	MU		4.0-0000	00.7-		JX2			MU	1.3.1			ML	UXU
	ΓL0, type R					-	De	TOF	ICE	ENDE	DE	T04	IE4	END4	D4
TS7	IE7	END7 END3	D7 D3	TS6 TS2	IE6	END6 END2	D6 D2	TS5 TS1	IE5 IE1	END5 END1	D5 D1	TS4 TS0	IE4 IE0	END4 END0	D4 D0
	FO0, type R			L		LINDZ	DZ	101	i.e.i	LINDT	D1	100	ILO	LINDO	
ADOUGH II	OU, type i	o, onser o	X040, 1636	l - (see page	333)										
										D.A	ιΤΑ				
ADCSSFII	FO1, type R	O, offset 0	x068, rese	t - (see page	e 393)										
			,		,										
										DA	ΤΑ	1			
ADCSSFII	FO2, type R	O, offset 0	x088, rese	t - (see page	e 393)										
										DA	TA				
ADCSSFI	FO3, type R	O, offset 0	x0A8, rese	t - (see pag	e 393)										
										DA	TΑ				
ADCSSFS	STAT0, type	RO, offset	0x04C, res	set 0x0000.	0100 (see	page 394)									
			FULL				EMPTY		HF	PTR			TF	PTR	
ADCSSFS	STAT1, type	RO, offset	0x06C, res	set 0x0000.	0100 (see	page 394)						I			
			FULL				EMPTY		1.15	PTR			т.	PTR	
ADCSSES	STAT2 type	PO offect		not 0×0000	0100 (200	page 304)	CIVIFIT		ПГ	TIK			11	TIK	
ADCOOF	STAT2, type	KO, Oliset	UXUGG, TES		o ioo (see	page 394)									
			FULL				EMPTY		HF	PTR			TE	PTR	
ADCSSFS	STAT3, type	RO. offset		set 0x0000.	0100 (see	page 394)									
	, .,,,,,,	,			(333	page 22.7									
			FULL				EMPTY		HF	PTR			TF	PTR	
ADCSSMI	UX1, type R	/W, offset (et 0x0000.00	000 (see p	age 395)		1							
		MU	IX3			М	JX2			MU	JX1			ML	JX0
ADCSSMI	UX2, type R	/W, offset (0x080, rese	et 0x0000.00	000 (see p	age 395)				-					
		MU	IX3			М	JX2			MU	JX1			ML	JX0
ADCSSCT	ΓL1, type R	W, offset 0	x064, rese	t 0x0000.00	00 (see pa	ige 396)									
TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCSSCT	L2, type R	W, offset 0	x084, rese	t 0x0000.00	000 (see pa	ge 396)						1			
TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
	JX3, type R						DZ	101	10.1	LIND	Di	100	ILO	LINDO	
ADOGGING	oxo, type it		JX07-0, 100		(300)	uge ooo,									
														М	JX0
ADCSSCT	L3, type R	/W, offset 0	x0A4, rese	t 0x0000.00	002 (see pa	ige 399)						1			
												TS0	IE0	END0	D0
ADCTMLE	3, type R/W	, offset 0x1	00, reset 0	x0000.0000	(see page	400)									
															LB
	al Asyn		s Recei	vers/Tra	nsmitte	rs (UAR1	ſs)								
	oase: 0x40 oase: 0x40														
UARTDR,	type R/W,	offset 0x00	0, reset 0x	0000.0000	(see page 4	10)									
,															
				OE	BE	PE	FE				DA	ATA			
UARTRSE	R/UARTECF	R, type RO,	offset 0x0	04, reset 0x	(0000.0000	(Reads) (s	ee page 41	2)							
												OE	BE	PE	FE
UARTRSE	R/UARTECF	R, type WO	offset 0x0	004, reset 0	x0000.0000	(Writes) (s	see page 4	12)							
											DA	ATA			
UARTER,	type RO, o	Hset 0x018	, reset 0x0	000.0090 (s	see page 41	14)									
								TXFE	RXFF	TXFF	RXFE	BUSY			
IIARTII PE	R, type R/W	offset Oxi	120 reset 0	X0000 0000	n (see nage	416)		INIL	IXII	IXII	IXIL	1 8031			
OAITHEI I	t, type latt	, onoce oxe	20, 10001		(occ page	7 10)									
											ILPE	I DVSR			
UARTIBRI	D, type R/W	/, offset 0x	024, reset (0x0000.000	0 (see page	e 417)									
							DIV	/INT							
UARTFBR	RD, type R/\	N, offset 0x	028, reset	0x0000.000	00 (see pag	je 418)									
												DIVE	RAC		
UARTLCR	RH, type R/\	N, offset 0>	02C, reset	0x0000.00	00 (see pag	ge 419)									
										E					
LIADTOT:	4 5.0	-#	00 1 -			404)		SPS	WI	-EN	FEN	STP2	EPS	PEN	BRK
UARTCTL	., type R/W,	omset 0x0	ას, reset 0:	xuuuu.0300 	(see page	421)									
						RXE	TXE	LBE					SIRLP	SIREN	UARTEN
UARTIFI 9	S, type R/W	offset flyf	34, reset N	X0000.0013	2 (see nage		IAL	LDL					GIINLE	OIIXLIN	OAKILIN
STATE L	., ., po 1044	, JJUL UXU	, 10061 0		- (555 page	0,									
											RXIFLSEL			TXIFLSEL	
UARTIM, t	type R/W, o	ffset 0x038	3, reset 0x0	0000.0000 (see page 4	25)							1		
					OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM				
UARTRIS,	type RO, c	offset 0x03	C, reset 0x	0000.0000	(see page 4	127)									
					OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS				

								T					T		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14 5, type RO, o	13	12 0 reset 0x0	11	10	9	8	7	6	5	4	3	2	1	0
JAKTIVIIS	, type KO, o	inset uxu4	u, reset uxu	000.0000	(see page 4	-20)									
					OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS				
UARTICR	t, type W1C,	offset 0x0	44. reset 0x	k0000.0000			. 20			1741110	10.000				
	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		(000 page										
					OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC				
UARTPer	iphID4, type	RO, offse	t 0xFD0, re:	set 0x0000	0.0000 (see	page 431)		1							
											PI	D4			
UARTPer	iphID5, type	RO, offse	t 0xFD4, re	set 0x0000	.0000 (see	page 432)									
											PI	D5			
UARTPer	iphID6, type	RO, offse	t 0xFD8, re	set 0x0000	.0000 (see	page 433)									
											PI	D6			
UARTPer	iphID7, type	RO, offse	t 0xFDC, re	set 0x0000	0.0000 (see	page 434)									
											PI	D7			
UARTPer	iphID0, type	RO, offse	t 0xFE0, res	set 0x0000	.0011 (see	page 435)									
												D0			
LIADTDaw	inhID4 tuna	DO offee	4 OvFF4 ===		0000 (222	222 426)					PI	D0			
UARTPER	iphID1, type	KO, onse	t uxrE4, re	set uxuuuu	.0000 (see	page 436)									
											PI	D1			
ΠΔRTPer	iphID2, type	RO offse	t OxFE8 res	set OxOOOO	0018 (see	nage 437)					• • • • • • • • • • • • • • • • • • • •				
OAITT CI	.pb2, type	110, 01100	C UXI 20, 10			page 401)									
											PI	D2			
UARTPer	iphID3, type	RO, offse	t 0xFEC, re	set 0x0000	0.0001 (see	page 438)		1							
											PI	D3			
UARTPC	ellID0, type F	RO, offset	0xFF0, rese	et 0x0000.0	000D (see p	age 439)									
											CI	D0			
UARTPC	ellID1, type F	RO, offset	0xFF4, rese	et 0x0000.0	00F0 (see p	age 440)									
											CI	D1			
UARTPC	ellID2, type F	RO, offset	0xFF8, rese	et 0x0000.0	0005 (see p	age 441)									
											CI	D2			
UARTPC	ellID3, type I	RO, offset	0xFFC, res	et 0x0000.	00B1 (see p	page 442)									
												D2			
•				O.),							CI	D3			
	onous Se: 0x4000		erface (S	SI)											
SSICR0, t	type R/W, of	fset 0x000	, reset 0x00	000.0000 (s	see page 45	6)									
			SC	CR				SPH	SPO	F	RF		D	SS	
SSICR1, t	type R/W, of	fset 0x004	, reset 0x00	000.0000 (s	ee page 45	8)									
												SOD	MS	SSE	LBM
SSIDR, ty	pe R/W, offs	et 0x008,	reset 0x000	0.0000 (se	e page 460)									
							DA	ATA							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSISR, ty	pe RO, offs	et 0x00C,	reset 0x000	0.0003 (see	e page 461)										
											BSY	RFF	RNE	TNF	TFE
SSICPSR,	type R/W,	offset 0x0	10, reset 0x	0000.0000	(see page 46	63)									
											000				
CCUM A.m	DAM offe	-4.0-04.4	A 0×000	0.0000 (224	naga 464)						CPSI	DVSR			
SSIIWI, typ	e R/W, ons	et uxu14,	reset 0x000	0.0000 (see	e page 464)										
												TXIM	RXIM	RTIM	RORIM
SSIRIS, ty	pe RO, offs	et 0x018,	reset 0x000	l 00.0008 (se	e page 466)							<u> </u>			
		<u>, </u>			, ,										
												TXRIS	RXRIS	RTRIS	RORRIS
SSIMIS, ty	pe RO, offs	set 0x01C,	, reset 0x00	00.0000 (se	e page 467)	1									
												TXMIS	RXMIS	RTMIS	RORMIS
SSIICR, ty	pe W1C, of	fset 0x020	0, reset 0x0	000.0000 (s	ee page 468	3)									
														RTIC	RORIC
SSIPeriph	IID4, type R	O, offset (DXFD0, rese	t 0x0000.00	000 (see pag	je 469)									
											PI	D4			
SSIPeriph	ID5. type R	O. offset (0xFD4. rese	t 0x0000.00	000 (see pag	ie 470)					•	D-1			
ос от.р		-,	1,1000		(oco pag	,0 0,									
											PI	D5			
SSIPeriph	ID6, type R	O, offset (0xFD8, rese	t 0x0000.00	000 (see pag	je 471)									
											PI	D6			
SSIPeriph	ID7, type R	O, offset (0xFDC, rese	et 0x0000.0	000 (see pag	ge 472)						_			
											PI	D7			
SSIPeriph	ID0, type R	O, offset (0xFE0, rese	t 0x0000.00	022 (see pag	je 473)									
											DI	D0			
SSIDorinh	ID1 type P	O offect (DyEE4 roso	t 0×0000 00	000 (see pag	10 474)					FI	D0			
SSIFETIPII	iiDi, type K	O, Uliset (JAI L4, 1656		(see pag	(C 474)									
											PI	D1			
SSIPeriph	ID2, type R	O, offset (0xFE8, rese	t 0x0000.00	018 (see pag	je 475)									
											PI	D2			
SSIPeriph	ID3, type R	O, offset (0xFEC, rese	t 0x0000.0	001 (see pag	ge 476)									
											PI	D3			
SSIPCellII	D0, type RC	, offset 0x	kFF0, reset	0x0000.000	D (see page	477)									
											2	D0			
eeine-iii	D4 turn D0	offert 0	EE4 massis	0~0000 005	0 (000 700)	. 479)					CI	D0			
SOIPCEIIIL	, type RC	, onset 0)	KFF4, FESET	UXUUUU.UUF	o (see page	4/0)									
											CI	D1			
SSIPCellI	D2. type RC	offset (1)	(FF8, reset	0x0000.000	05 (see page	479)					- 01				
	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,	.,		(-,									
											CI	D2			

				I				T	1 4- 1						
31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22	21 5	20	19 3	18	17	16 0
			FFC, reset				0		6	<u> </u>	4	3	2	1	U
SSIPCelliD	os, type RO	, onset ux	rrc, reset		on (see pag	je 460)									
											CI	l D3			
Inter-Int	ografod	Circuit	(I ² C) Inte	arface											
I ² C Mas		Circuit	(1 0) 1110	errace											
	e: 0x4002	0000													
			0, reset 0x0	000.0000											
,,,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		,												
											SA				R/S
I2CMCS, ty	ype RO, off	set 0x004.	, reset 0x00	l 000.0000 (R	eads)										
	, ,	·	,		,										
									BUSBSY	IDLE	ARBLST	DATACK	ADRACK	ERROR	BUSY
I2CMCS, ty	ype WO, of	set 0x004	l, reset 0x00	000.0000 (V	Vrites)							ļ.			
												ACK	STOP	START	RUN
I2CMDR, ty	ype R/W, of	fset 0x008	8, reset 0x0	000.0000											
											DA	TA			
I2CMTPR,	type R/W, o	offset 0x00	0C, reset 0x	0000.0001											
												TPR			
I2CMIMR, 1	type R/W, o	ffset 0x01	10, reset 0x	0000.0000											
															IM
I2CMRIS, t	type RO, of	fset 0x014	l, reset 0x0	000.000											
															RIS
I2CMMIS, t	type RO, of	fset 0x018	8, reset 0x0	000.0000								ı			
10011100 /															MIS
IZCMICR, t	type WO, of	rset uxu1	C, reset 0x0	1000.0000				1							
															10
ISCMCB 6	uno D/M of	foot 0v02	0, reset 0x0	000 0000											IC
izcivick, ty	ype R/vv, oi	ISEL UXUZI	u, reset uxu	1											
										SFE	MFE				LPBK
l4 l4		0114	(120) 14	•						OI L	IVII L				LIBIX
Inter-int		Circuit	(I ² C) Inte	егтасе											
	'e se: 0x4002	0000													
			00 rooot 0v	2000 0000											
IZOSUAK,	type r./vv, (JIISEL UXÖ	00, reset 0x												
												OAR			
I2CSCSR	type RO. of	fset 0x80	4, reset 0x0	0000.0000 (Reads)							J, II (
,	-yp5 110, 01		.,												
													FBR	TREQ	RREQ
I2CSCSR.	type WO, o	ffset 0x80)4, reset 0x(0000.0000 (Writes)										
,	J, •		,		,										
															DA
I2CSDR, ty	pe R/W, of	fset 0x808	3, reset 0x0	000.000											
											DA	I TA			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CSIMR,	type R/W,	offset 0x800	C, reset 0x0	0000.0000				I				1			
															DATAIM
ISCEDIS 1	type PO e	ffset 0x810,	rosot 0v00	00 0000											DATAIN
1203113, 1	type NO, o	iiset uxo iu,	Teset 0x00	00.000											
															DATARIS
I2CSMIS	type RO o	ffset 0x814	reset 0x00	000 0000											D/ ti/ ti tio
,	.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		, 10001 0200												
															DATAMIS
I2CSICR, 1	type WO, c	offset 0x818	, reset 0x00	000.0000											
															DATAIC
Etherne	et Contro	oller				1		1	-	1					
Etherne		00.													
	1004.8000)													
MACRIS/N	MACIACK,	type RO, of	fset 0x000,	reset 0x00	00.0000 (F	Reads)									
									PHYINT	MDINT	RXER	FOV	TXEMP	TXER	RXINT
MACRIS/N	MACIACK,	type WO, o	ffset 0x000	, reset 0x00	000.0000 (\	Writes)									
									PHYINT	MDINT	RXER	FOV	TXEMP	TXER	RXINT
MACIM, ty	pe R/W, of	ffset 0x004,	reset 0x00	00.007F											
									PHYINTM	MDINTM	RXERM	FOVM	TXEMPM	TXERM	RXINTM
MACRCTI	L, type R/W	, offset 0x0	08, reset 0	x0000.0008	3										
											RSTFIFO	BADCRC	PRMS	AMUL	RXEN
MACTCTL	, type R/W	, offset 0x0	OC, reset 0	x0000.0000)										
											DUPLEX		CRC	PADEN	TXEN
MACDATA	A, type RO,	offset 0x01	10, reset 0x	0000.0000	(Reads)										
								DATA							
							RXI	DATA							
MACDATA	A, type WO	, offset 0x0	10, reset 0x	c0000.0000	(Writes)										
								DATA							
	DAM -	- # 4 O O4 4	4 0 - 0	200 0000			IXL	DATA							
WACIAU, t	type R/W, C	offset 0x014	MAC								MAG	OCT2			
			MACC									OCT3 OCT1			
MACIA1 1	type R/W o	offset 0x018									WAO				
WIAGIAT, C	lype ratt, c	JIISEL OXO IO	, 10301 020	000.0000											
			MAC	DCT6							MAC	OCT5			
MACTHR.	type R/W.	offset 0x01													
,	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,														
												THE	RESH		
		V. offset 0x0)20, reset 0	x0000.0000)			1		1					
MACMCTI	L, type R/V														
МАСМСТІ	L, type R/V														
MACMCTI	L, type R/V									REGADR				WRITE	START
		offset 0x02	24, reset 0x	0000.0080						REGADR				WRITE	START
			24, reset 0x	0800.0080						REGADR				WRITE	START

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
МАСМТХ	D, type R/W	, offset 0x	02C, reset (0x0000.000	0										
					1		MD	TX							
MACMRX	D, type R/W	/. offset 0x	030. reset (0x0000.000	0										
	_, ,,,,	,	,												
							MD	DY							
MA OND 4	00 -#			00.0000			IVID								
MACNP, t	ype RO, off	set uxu34,	reset uxuu	UU.UUUU I											
												NI	PR		
MACTR, 1	type R/W, of	fset 0x038	, reset 0x00	000.000											
															NEWTX
Ethern	et Contro	oller													
	nagemen														
MR0, type	e R/W, addre														
RESET	LOOPBK	SPEEDSL	ANEGEN	PWRDN	ISO	RANEG	DUPLEX	COLT							
MR1, type	e RO, addre	ss 0x01, re	set 0x7849)											
	100X_F	100X_H	10T_F	10T_H					MFPS	ANEGC	RFAULT	ANEGA	LINK	JAB	EXTD
MR2, type	e RO, addre	ss 0x02, re	set 0x000E												
							OUI[21:61							
MR3 type	e RO, addre	ss OxO3 re	set 0x7237	,											
iii to, typi	o ito, addio		[5:0]					N/	N				R	N	
								IV	IIV .				- K	IN	
	e R/W, addre		eset 0x01E	.1											
NP		RF					A3	A2	A1	A0			S		
		_													
MR5, type	e RO, addre	ss 0x05, re	set 0x0000												
MR5, type	e RO, addre ACK	ss 0x05, re	set 0x0000		1	A[7:0]						S		
NP		RF			1	A[7:0]						S		
NP	ACK	RF				A[7:0]				PDF	LPNPA	S	PRX	LPANEGA
NP MR6, type	ACK e RO, addre	RF ss 0x06, re	set 0x0000			A[7:0]				PDF	LPNPA	S	PRX	LPANEGA
MR6, type	ACK e RO, addre	RF ss 0x06, re	reset 0x0000	40	NL10	A[7:0]			APOL		LPNPA	S		
MR6, type MR16, type RPTR	ACK e RO, addre pe R/W, add	RF ss 0x06, re ress 0x10,	reset 0x0000	40 SQEI	NL10	A[7:0]			APOL	PDF RVSPOL	LPNPA	S	PRX PCSBP	LPANEGA
MR6, type MR16, type RPTR	ACK e RO, addre	RF ss 0x06, re ress 0x10,	reset 0x0000	40 SQEI	NL10	A[APOL		LPNPA	S		RXCC
MR6, type MR16, type RPTR	ACK e RO, addre pe R/W, add	RF ss 0x06, re ress 0x10,	reset 0x0000 TXHIM reset 0x000	40 SQEI	NL10	A[APOL		LPNPA	S		RXCC
MR6, type MR16, type RPTR MR17, type	ACK e RO, addre pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11,	reset 0x0000	40 SQEI	NL10			JABBER_INT	RXER_INT		RVSPOL		S LSCHG_INT	PCSBP	RXCC
MR6, type MR16, type RPTR MR17, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11,	reset 0x0000 TXHIM reset 0x000	40 SQEI			ANEGCOMP_LE	JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR6, typo MR16, typo MR16, typ RPTR MR17, typ JABBER_IE	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE	reset 0x0000 TXHIM reset 0x000	40 SQEI 000				JABBER_INT	RXER_INT		RVSPOL			PCSBP	
MR6, typo MR16, typo MR16, typ RPTR MR17, typ JABBER_IE	ACK e RO, addre pe R/W, add INPOL pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE	reset 0x0000 TXHIM reset 0x000 PDF_IE	40 SQEI 00 LPACK_IE	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR16, type MR16, type MR17, type MR17, type MR18, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r	reset 0x0000 TXHIM reset 0x000 PDF_IE reset 0x000 ANEGF	40 SQEI 00 LPACK_IE				JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addr	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r	reset 0x0000 TXHIM reset 0x000 PDF_IE reset 0x000 ANEGF	40 SQEI 00 LPACK_IE	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR16, type MR16, type MR16, type RPTR MR17, type MR18, type MR18, type MR19, type T	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r	reset 0x0000 TRAIM TOUR PDF_IE TRAIN ANEGF	40 SQEI 00 LPACK_IE 0 DPLX	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR16, type MR16, type MR16, type RPTR MR17, type MR18, type MR18, type MR19, type T	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addr	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r	reset 0x0000 TRAIM TOUR PDF_IE TRAIN ANEGF	40 SQEI 00 LPACK_IE 0 DPLX	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT	RXER_INT		RVSPOL			PCSBP	RXCC
MR16, type MR16, type MR16, type RPTR MR17, type MR18, type MR18, type MR19, type T	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r	reset 0x0000 TRAIM TOUR PDF_IE TRAIN ANEGF	40 SQEI 00 LPACK_IE 0 DPLX	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT			RVSPOL		LSCHG_INT	PCSBP	RXCC
MR16, type MR16, type MR16, type RPTR MR17, type MR18, type MR19, type T MR23, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17,	reset 0x0000 TRHIM PDF_IE PESET 0x0000 ANEGF reset 0x400 reset 0x000	40 SQEI 00 LPACK_IE 0 DPLX 00	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE	JABBER_INT		PRX_INT	RVSPOL		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type RPTR MR17, type MR17, type MR18, type MR19, type MR23, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17,	reset 0x0000 TRHIM PDF_IE PESET 0x0000 ANEGF reset 0x400 reset 0x000	40 SQEI 00 LPACK_IE 0 DPLX 00	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE			PRX_INT	RVSPOL		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type T MR23, type MR24, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addr pe R/W, add XO pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18,	reset 0x0000 TRHIM PDF_IE PESET 0x0000 ANEGF reset 0x400 reset 0x000	40 SQEI 00 LPACK_IE 0 DPLX 00	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type T MR23, type MR24, type Analog	ACK e RO, addre oe R/W, add INPOL oe R/W, add RXER_IE oe RO, addr oe R/W, add xO oe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, i ress 0x13, ress 0x17, ress 0x18,	reset 0x0000 TRHIM PDF_IE PESET 0x0000 ANEGF reset 0x400 reset 0x000	40 SQEI 00 LPACK_IE 0 DPLX 00	LSCHG_IE	RFAULT_IE	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR19, type T MR23, type MR24, type Analog Base 0x	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre pe R/W, add xO pe R/W, add pe R/W, add pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18,	reset 0x0000 TXHIM reset 0x000 PDF_IE Reset 0x000 ANEGF reset 0x40 reset 0x000	40 SQEI 000 LPACK_IE 0 DPLX 000 10	LSCHG_IE RATE	RFAULT_IE RXSD	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR19, type T MR23, type MR24, type Analog Base 0x	ACK e RO, addre oe R/W, add INPOL oe R/W, add RXER_IE oe RO, addr oe R/W, add xO oe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18,	reset 0x0000 TXHIM reset 0x000 PDF_IE Reset 0x000 ANEGF reset 0x40 reset 0x000	40 SQEI 000 LPACK_IE 0 DPLX 000 10	LSCHG_IE RATE	RFAULT_IE RXSD	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR19, type T MR23, type MR24, type Analog Base 0x	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre pe R/W, add xO pe R/W, add pe R/W, add pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18,	reset 0x0000 TXHIM reset 0x000 PDF_IE Reset 0x000 ANEGF reset 0x40 reset 0x000	40 SQEI 000 LPACK_IE 0 DPLX 000 10	LSCHG_IE RATE	RFAULT_IE RXSD	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT D[3:0] C_SD	ANEGCOMP_INT
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type T MR23, type MR24, type Analog Base 0x ACMIS, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addr pe R/W, add XO pe R/W, add pe R/W, add yo pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18, offset 0x00	reset 0x0000 TAHIM reset 0x000 PDF_IE reset 0x000 ANEGF reset 0x00 reset 0x00 reset 0x00 00, reset 0x	40 SQEI 00 LPACK_IE 0 DPLX 00 CO	RATE (see page 5	RFAULT_IE RXSD 571)	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT	RXCC
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type T MR23, type MR24, type Analog Base 0x ACMIS, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addre pe R/W, add xO pe R/W, add pe R/W, add pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18, offset 0x00	reset 0x0000 TAHIM reset 0x000 PDF_IE reset 0x000 ANEGF reset 0x00 reset 0x00 reset 0x00 00, reset 0x	40 SQEI 00 LPACK_IE 0 DPLX 00 CO	RATE (see page 5	RFAULT_IE RXSD 571)	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT D[3:0] C_SD	ANEGCOMP_INT
MR16, type MR16, type MR16, type MR17, type MR17, type MR18, type MR19, type T MR23, type MR24, type Analog Base 0x ACMIS, type	ACK e RO, addre pe R/W, add INPOL pe R/W, add RXER_IE pe RO, addr pe R/W, add XO pe R/W, add pe R/W, add yo pe R/W, add	RF ss 0x06, re ress 0x10, ress 0x11, PRX_IE ess 0x12, r ress 0x13, ress 0x17, ress 0x18, offset 0x00	reset 0x0000 TAHIM reset 0x000 PDF_IE reset 0x000 ANEGF reset 0x00 reset 0x00 reset 0x00 00, reset 0x	40 SQEI 00 LPACK_IE 0 DPLX 00 CO	RATE (see page 5	RFAULT_IE RXSD 571)	ANEGCOMP_IE		LED ²	PRX_INT	RVSPOL PDF_INT		LSCHG_INT	PCSBP RFAULT_INT D[3:0] C_SD	RXCC

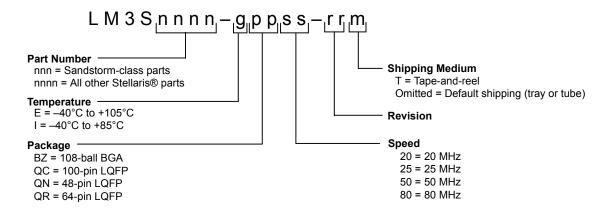
				T											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACINTEN,	I, type R/W,	offset 0x00)8, reset 0x	k0000.0000	(see page 5	573)									
														IN1	IN0
ACREFCT	TL, type R/V	V, offset 0x	010, reset	0x0000.000	0 (see page	e 574)									
						EN	RNG						VF	REF	
ACSTAT0), type RO, o	offset 0x02	0, reset 0x(0000.0000 (see page 57	75)									
														OVAL	
ACSTAT1	, type RO, o	offset 0x04	0, reset 0x(0000.0000 (see page 57	75)									
														OVAL	
ACCTL0,	type R/W, o	ffset 0x024	1, reset 0x0	0000.0000 (see page 57	76)									
				TOEN	ASF	RCP		TSLVAL	TS	EN	ISLVAL	ISI	ΞN	CINV	
ACCTL1.	type R/W, c	ffset 0x044	1, reset 0x0	0000.0000 (see page 57	76)									
,	,-			1	, 5	,									
				TOEN	ASF	RCP		TSLVAL	TS	EN	ISLVAL	ISI	ΞN	CINV	
Dules 1	Alidab Br -	dulete:	(D)A/84)	1 . 5	7.51								-		
	Nidth Mo 4002.8000		(PVVIVI)												
PWMCTL,	, type R/W,	offset 0x00	0, reset 0x	(0000.0000	(see page 5	587)									
															000
															GlobalSync0
															99
PWMSYN	IC, type R/V	, offset 0x	004, reset (0x0000.000	0 (see page	588)									
															Sync0
PWMENA	ABLE, type I	R/W. offset	0x008. res	et 0x0000.0	0000 (see pa	age 589)									
	,,,,,,,			1	(-9									
														PWM1En	PWM0En
D\A/MINI\/E	ERT, type R	IM offeet (V00C #000	+ 0~000 0	000 (000 00	ao E00)								1 WWW.IE.	1 WHOLI
PVVIVIINVE	ERI, type R	w, onset u	xuuc, rese	et uxuuuu.ut	ooo (see pa	ge 590)									
														DIA/A441	DIA/AAOL
														PWM1Inv	PVVIVIUINV
PWMFAU	ILT, type R/\	v. offset 0x	010, reset	0x0000.000	00 (see page	e 591)									
		,													
		,												_	_
														Fault1	Fault0
PWMINTE	EN, type R/V		014, reset	0x0000.000	00 (see page	e 592)								Fault1	Fault0
PWMINTE	EN, type R/V		014, reset	0x0000.000	00 (see page	e 592)								Fault1	Fault0
PWMINTE	EN, type R/V		014, reset	0x0000.000	00 (see page	e 592)								Fault1	
	EN, type R/N	V, offset 0×				,								Fault1	IntFault
		V, offset 0×				,								Fault1	IntFault
		V, offset 0×				,								Fault1	IntFault IntPWM0
PWMRIS,		V, offset 0x	, reset 0x0	0000.0000 (s	see page 59	3)								Fault1	IntFault IntPWM0
PWMRIS,	type RO, o	V, offset 0x	, reset 0x0	0000.0000 (s	see page 59	3)								Fault1	IntFault IntPWM0
PWMRIS,	type RO, o	V, offset 0x	, reset 0x0	0000.0000 (s	see page 59	3)								Fault1	IntFault IntPWM0 IntFault IntPWM0
PWMRIS,	type RO, o	V, offset 0x ffset 0x018	, reset 0x0 01C, reset	0000.0000 (s	see page 59	3) e 594)								Fault1	IntFault IntPWM0 IntFault IntPWM0
PWMRIS,	type RO, o	V, offset 0x ffset 0x018	, reset 0x0 01C, reset	0000.0000 (s	see page 59	3) e 594)								Fault1	IntFault IntPWM0 IntFault IntPWM0
PWMRIS,	type RO, o	V, offset 0x ffset 0x018	, reset 0x0 01C, reset	0000.0000 (s	see page 59	3) e 594)								Fault1	IntFault IntPWM0 IntFault IntPWM0 IntFault IntPWM0
PWMRIS, PWMISC,	type RO, o	V, offset 0x ffset 0x018 C, offset 0x O, offset 0	, reset 0x0 01C, reset x020, reset	0000.0000 (s	00 (see page 59	3) e 594) ge 595)								Fault1	IntFault IntPWM0 IntFault IntPWM0
PWMRIS, PWMISC,	type RO, o	V, offset 0x ffset 0x018 C, offset 0x O, offset 0	, reset 0x0 01C, reset x020, reset	0000.0000 (s	00 (see page 59	3) e 594) ge 595)								Fault1	IntFault IntPWM0 IntFault IntPWM0 IntFault IntPWM0
PWMRIS, PWMISC,	type RO, o	V, offset 0x ffset 0x018 C, offset 0x O, offset 0	, reset 0x0 01C, reset x020, reset	0000.0000 (s	00 (see page 59	3) e 594) ge 595)					CmpAUpd		Debug	Fault1	IntFault IntPWM0 IntFault IntPWM0 IntFault IntPWM0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0INT	EN, type F	R/W, offset 0)x044, rese	t 0x0000.00	00 (see pag	ge 598)		1							
		TrCmpBD	TrCmpBU	TrCmpAD	TrCmpAU	TrCntLoad	TrCntZero			IntCmpBD	IntCmpBU	IntCmpAD	IntCmpAU	IntCntLoad	IntCntZero
PWM0RIS	s, type RO,	offset 0x04	8, reset 0x	0000.0000 (see page 6	01)									
										IntCmpBD	IntCmpBU	IntCmpAD	IntCmpAU	IntCntLoad	IntCntZero
PWM0ISC	type R/W	/1C, offset 0	x04C, rese	t 0x0000.00	000 (see pa	ge 602)									
										IntCmpBD	IntCmpBU	IntCmpAD	IntCmpAU	IntCntLoad	IntCntZero
PWM0LO	AD, type R	/W, offset 0	x050, reset	0x0000.00	00 (see pag	je 603)		1							
								l							
DWMOCO	IINT tops	PO offerst (NO54 ====	+ 0×0000 00	MM (800 75	no 604\	LC	oad							
PANINGO	UNI, type	RO, offset (AUD4, FESE	. 020000.00	oo (see pag	ye 004)									
							Co	ount							
PWM0CM	IPA, type R	W, offset 0	x058. reset	0x0000.00	00 (see pag	ie 605)									
					(***	,,									
							Coi	mpA							
PWM0CM	IPB, type R	R/W, offset 0	x05C, rese	t 0x0000.00	000 (see pag	ge 606)									
							Coi	mpB				•			
PWM0GE	NA, type R	2/W, offset 0	x060, reset	0x0000.00	00 (see pag	ge 607)									
				ActCr	npBD	ActCı	mpBU	ActC	mpAD	ActC	mpAU	Actl	_oad	Act	Zero
PWM0GE	NB, type R	V/W, offset 0	x064, reset	0x0000.00	00 (see pag	ge 610)									
				Anton	mnPD	A 04C	mpDLI	A c+C	mnAD.	Acto	mpALL	A =41	_oad	A a4'	Zero
DWMODB	CTI type	R/W, offset	0×068 roo	ActCr			mpBU	ACIU	mpAD	ACTU	mpAU	ACU	_UdU	ACL	<u> </u>
PANINDE	CTL, type	R/VV, OIISEL	UXUGG, TESE		ooo (see pa	ige 013)									
															Enable
PWM0DB	RISE, type	R/W, offset	0x06C, res	set 0x0000.	0000 (see n	page 614)									
	, -, pe	11, 11100			(22 0 p	. 3)									
									Ris	eDelay		I			
PWM0DB	FALL, type	R/W, offse	t 0x070, res	set 0x0000.	0000 (see p	age 615)									
									Fa	llDelay					

C Ordering and Contact Information

C.1 Ordering Information

The figure below defines the full set of potential orderable part numbers for all the Stellaris[®] LM3S microcontrollers. See the Package Option Addendum for the valid orderable part numbers for the LM3S6432 microcontroller.



C.2 Part Markings

The Stellaris microcontrollers are marked with an identifying number. This code contains the following information:

- The first line indicates the part number, for example, LM3S9B90.
- In the second line, the first eight characters indicate the temperature, package, speed, revision, and product status. For example in the figure below, IQC80C0X indicates an Industrial temperature (I), 100-pin LQFP package (QC), 80-MHz (80), revision C0 (C0) device. The letter immediately following the revision indicates product status. An X indicates experimental and requires a waiver; an S indicates the part is fully qualified and released to production.
- The remaining characters contain internal tracking numbers.



C.3 Kits

The Stellaris Family provides the hardware and software tools that engineers need to begin development quickly.

- Reference Design Kits accelerate product development by providing ready-to-run hardware and comprehensive documentation including hardware design files
- Evaluation Kits provide a low-cost and effective means of evaluating Stellaris microcontrollers before purchase
- Development Kits provide you with all the tools you need to develop and prototype embedded applications right out of the box

See the website at www.ti.com/stellaris for the latest tools available, or ask your distributor.

C.4 Support Information

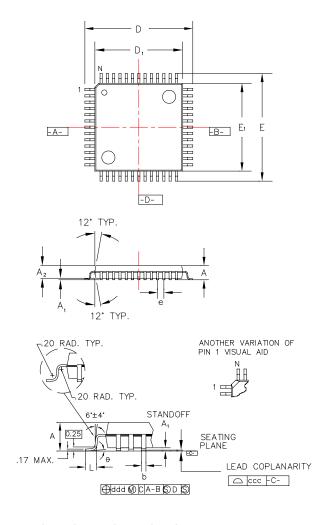
For support on Stellaris products, contact the TI Worldwide Product Information Center nearest you: http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm.

D Package Information

D.1 100-Pin LQFP Package

D.1.1 Package Dimensions

Figure D-1. Stellaris LM3S6432 100-Pin LQFP Package Dimensions



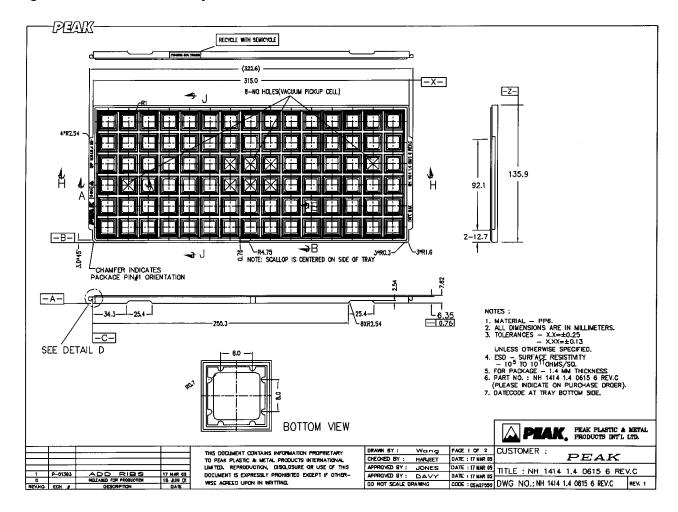
Note: The following notes apply to the package drawing.

- 1. All dimensions shown in mm.
- 2. Dimensions shown are nominal with tolerances indicated.
- 3. Foot length 'L' is measured at gage plane 0.25 mm above seating plane.

Cymbolo	Leads	100L
Symbols	Leaus	100L
A	Max.	1.60
A ₁	-	0.05 Min./0.15 Max
A ₂	±0.05	1.40
D	±0.20	16.00
D ₁	±0.05	14.00
E	±0.20	16.00
E ₁	±0.05	14.00
L	+0.15/-0.10	0.60
е	Basic	0.50
b	+0.05	0.22
θ	-	0°-7°
ddd	Max.	0.08
ccc	Max.	0.08
JEDEC Re	ference Drawing	MS-026
Variatio	n Designator	BED

D.1.2 Tray Dimensions

Figure D-2. 100-Pin LQFP Tray Dimensions



D.1.3 Tape and Reel Dimensions

Note: In the figure that follows, pin 1 is located in the top right corner of the device.

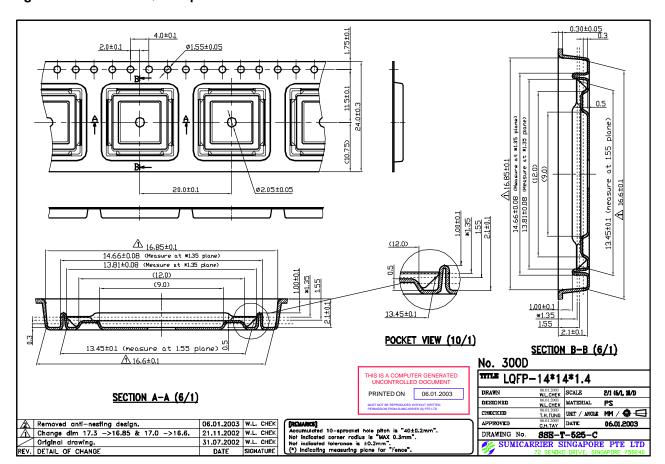
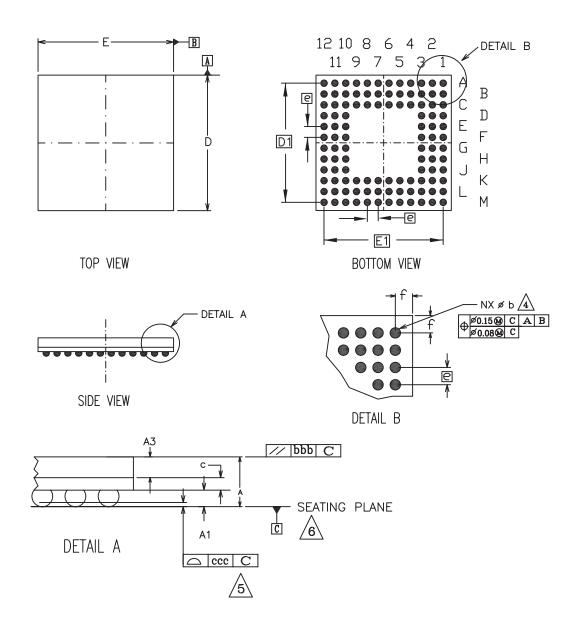


Figure D-3. 100-Pin LQFP Tape and Reel Dimensions

D.2 108-Ball BGA Package

D.2.1 Package Dimensions

Figure D-4. Stellaris LM3S6432 108-Ball BGA Package Dimensions



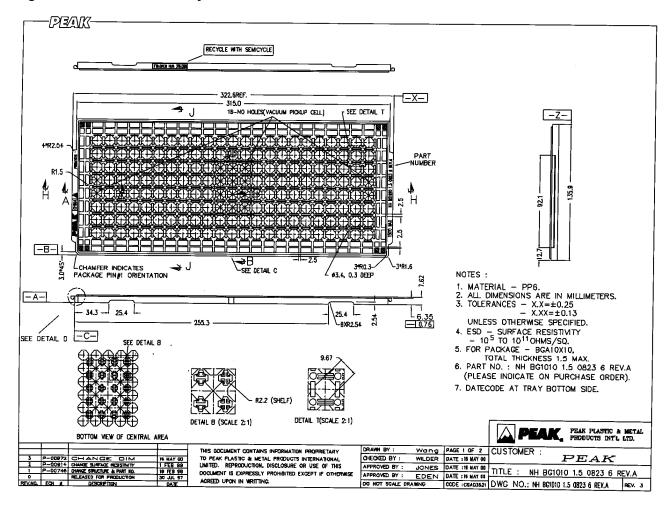
Note: The following notes apply to the package drawing.

- 1. ALL DIMENSIONS ARE IN MILLIMETERS.
- 2. 'e' REPRESENTS THE BASIC SOLDER BALL GRID PITCH.
- 3. "M" REPRESENTS THE BASIC SOLDER BALL MATRIX SIZE.
 AND SYMBOL 'N' IS THE NUMBER OF BALLS AFTER DEPOPULATING.
- \triangle 'b' IS MEASURABLE AT THE MAXIMUM SOLDER BALL DIAMETER AFTER REFLOW PARALLEL TO PRIMARY DAIUM $\boxed{\hspace{-0.05cm}C}$.
- ⚠ DIMENSION 'ccc' IS MEASURED PARALLEL TO PRIMARY DATUM [].
- PRIMARY DATUM [] AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
- 7. PACKAGE SURFACE SHALL BE MATTE FINISH CHARMILLES 24 TO 27.
- 8. SUBSTRATE MATERIAL BASE IS BT RESIN.
- 9. THE OVERALL PACKAGE THICKNESS "A" ALREADY CONSIDERS COLLAPSE BALLS
- 10. DIMENSIONING AND TOLERANCING PER ASME Y14.5M 1994.

Symbols	MIN	NOM	MAX						
A	1.22	1.36	1.50						
A1	0.29	0.34	0.39						
A3	0.65	0.70	0.75						
С	0.28	0.32	0.36						
D	9.85	10.00	10.15						
D1		8.80 BSC							
E	9.85	9.85 10.00							
E1		8.80 BSC							
b	0.43	0.48	0.53						
bbb		.20							
ddd		.12							
е		0.80 BSC							
f	-	0.60	-						
M									
n	108								
	REF: JI	EDEC MO-219F							

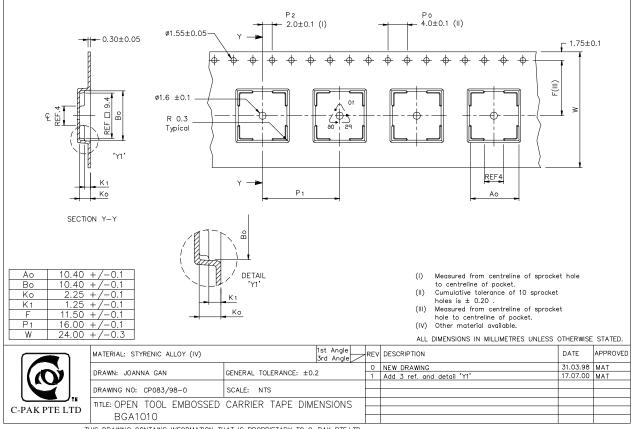
D.2.2 Tray Dimensions

Figure D-5. 108-Ball BGA Tray Dimensions



Tape and Reel Dimensions D.2.3

Figure D-6. 108-Ball BGA Tape and Reel Dimensions



THIS DRAWING CONTAINS INFORMATION THAT IS PROPRIETARY TO C-PAK PTE.LTD.





8-Feb-2015

PACKAGING INFORMATION

Orderable Device	Status	Package Type	Package Drawing	Pins	Package Qty	Eco Plan	Lead/Ball Finish	MSL Peak Temp	Op Temp (°C)	Device Marking (4/5)	Samples
LM3S6432-IBZ50-A2	NRND	NFBGA	ZCR	108	184	Green (RoHS & no Sb/Br)	SNAGCU	Level-3-260C-168 HR	-40 to 85	LM3S6432 IBZ50 ZCR	
LM3S6432-IBZ50-A2T	NRND	NFBGA	ZCR	108	1000	Green (RoHS & no Sb/Br)	SNAGCU	Level-3-260C-168 HR	-40 to 85	LM3S6432 IBZ50 ZCR	
LM3S6432-IQC50-A2	NRND	LQFP	PZ	100	90	Green (RoHS & no Sb/Br)	SN	Level-3-260C-168 HR	-40 to 85	LM3S6432 IQC50	
LM3S6432-IQC50-A2T	NRND	LQFP	PZ	100	1000	Green (RoHS & no Sb/Br)	SN	Level-3-260C-168 HR	-40 to 85	LM3S6432 IQC50	

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

- (3) MSL, Peak Temp. The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.
- (4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.
- (5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.



PACKAGE OPTION ADDENDUM

8-Feb-2015

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

PACKAGE MATERIALS INFORMATION

www.ti.com 18-Jul-2014

TAPE AND REEL INFORMATION





	Dimension designed to accommodate the component width
	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing			Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
LM3S6432-IQC50-A2T	LQFP	PZ	100	1000	330.0	24.4	17.0	17.0	2.1	20.0	24.0	Q2

www.ti.com 18-Jul-2014



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
LM3S6432-IQC50-A2T	LQFP	PZ	100	1000	367.0	367.0	45.0

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have not been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products Applications

logic.ti.com

Audio www.ti.com/audio Automotive and Transportation www.ti.com/automotive **Amplifiers** amplifier.ti.com Communications and Telecom www.ti.com/communications **Data Converters** dataconverter.ti.com Computers and Peripherals www.ti.com/computers **DLP® Products** www.dlp.com Consumer Electronics www.ti.com/consumer-apps DSP dsp.ti.com **Energy and Lighting** www.ti.com/energy Clocks and Timers www.ti.com/clocks Industrial www.ti.com/industrial Interface interface.ti.com Medical www.ti.com/medical Logic Security www.ti.com/security

Power Mgmt Space, Avionics and Defense www.ti.com/space-avionics-defense power.ti.com

Microcontrollers www.ti.com/video microcontroller.ti.com Video and Imaging

www.ti-rfid.com

OMAP Applications Processors TI E2E Community www.ti.com/omap e2e.ti.com

Wireless Connectivity www.ti.com/wirelessconnectivity