

LM3S8933 Microcontroller

DATA SHEET

Legal Disclaimers and Trademark Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH LUMINARY MICRO PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN LUMINARY MICRO'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, LUMINARY MICRO ASSUMES NO LIABILITY WHATSOEVER, AND LUMINARY MICRO DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF LUMINARY MICRO'S PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. LUMINARY MICRO'S PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE-SUSTAINING APPLICATIONS.

Luminary Micro may make changes to specifications and product descriptions at any time, without notice. Contact your local Luminary Micro sales office or your distributor to obtain the latest specifications before placing your product order.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Luminary Micro reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Copyright © 2007-2008 Luminary Micro, Inc. All rights reserved. Stellaris, Luminary Micro, and the Luminary Micro logo are registered trademarks of Luminary Micro, Inc. or its subsidiaries in the United States and other countries. 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.

Luminary Micro, Inc. 108 Wild Basin, Suite 350 Austin, TX 78746 Main: +1-512-279-8800 Fax: +1-512-279-8879 http://www.luminarymicro.com







Table of Contents

Revis	ion History	20
	t This Document	
	nce	
	This Manual	
	ed Documents	
Docun	nentation Conventions	22
1	Architectural Overview	
1.1	Product Features	
1.2	Target Applications	
1.3	High-Level Block Diagram	
1.4	Functional Overview	
1.4.1	ARM Cortex™-M3	
1.4.2	Motor Control Peripherals	
1.4.3	Analog Peripherals	
1.4.4	Serial Communications Peripherals	
1.4.5	System Peripherals	
1.4.6	Memory Peripherals	
1.4.7	Additional Features	
1.4.8	Hardware Details	37
2	ARM Cortex-M3 Processor Core	39
2.1	Block Diagram	40
2.2	Functional Description	40
2.2.1	Serial Wire and JTAG Debug	
2.2.2	Embedded Trace Macrocell (ETM)	
2.2.3	Trace Port Interface Unit (TPIU)	
2.2.4	ROM Table	
2.2.5	Memory Protection Unit (MPU)	
2.2.6	Nested Vectored Interrupt Controller (NVIC)	
3	Memory Map	45
4	Interrupts	47
5	JTAG Interface	50
5.1	Block Diagram	51
5.2	Functional Description	51
5.2.1	JTAG Interface Pins	52
5.2.2	JTAG TAP Controller	53
5.2.3	Shift Registers	54
5.2.4	Operational Considerations	54
5.3	Initialization and Configuration	57
5.4	Register Descriptions	57
5.4.1	Instruction Register (IR)	57
5.4.2	Data Registers	59
6	System Control	61
6.1	Functional Description	61
6.1.1	Device Identification	61

Reset Control	
Power Control	64
Clock Control	65
System Control	68
Initialization and Configuration	69
Register Map	70
Register Descriptions	71
Hibernation Module	125
Block Diagram	126
Functional Description	126
Register Access Timing	126
Clock Source	127
Battery Management	128
Real-Time Clock	129
Non-Volatile Memory	129
Power Control	129
Interrupts and Status	130
Initialization and Configuration	130
Initialization	130
RTC Match Functionality (No Hibernation)	131
External Wake-Up from Hibernation	131
RTC/External Wake-Up from Hibernation	131
Register Map	131
Register Descriptions	132
Internal Memory	145
Functional Description	145
SRAM Memory	145
Flash Memory	146
Flash Memory Initialization and Configuration	147
Flash Programming	147
Nonvolatile Register Programming	148
Register Map	148
Flash Register Descriptions (Flash Control Offset)	149
Flash Register Descriptions (System Control Offset)	
	156
General-Purpose Input/Outputs (GPIOs)	156 169
	156 169 169
General-Purpose Input/Outputs (GPIOs)	156 169 169 170
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control	156 169 169 170 171
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control Interrupt Control	156 169 169 170 171 172
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control Interrupt Control Mode Control	156 169 169 170 171 172 172
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control Interrupt Control Mode Control Commit Control	156 169 169 170 171 172 172
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control Interrupt Control Mode Control Commit Control Pad Control	156 169 169 170 171 172 172 172
General-Purpose Input/Outputs (GPIOs) Functional Description Data Control Interrupt Control Mode Control Commit Control Pad Control Identification	156 169 170 171 172 172 172 173
	Clock Control System Control Initialization and Configuration Register Map Register Descriptions Hibernation Module Block Diagram Functional Description Register Access Timing Clock Source Battery Management Real-Time Clock Non-Volatile Memory Power Control Interrupts and Status Initialization and Configuration Initialization RTC Match Functionality (No Hibernation) RTC Match/Wake-Up from Hibernation External Wake-Up from Hibernation RTC/External Wake-Up from Hibernation Register Map Register Descriptions Internal Memory Block Diagram Functional Description SRAM Memory Flash Memory Flash Memory Flash Memory Flash Memory Initialization and Configuration Flash Programming Nonvolatile Register Programming Register Map Register Map Register Map Register Programming Register Map

General-Purpose Timers	211
Block Diagram	211
Functional Description	212
GPTM Reset Conditions	213
32-Bit Timer Operating Modes	213
16-Bit Timer Operating Modes	214
-	
32-Bit Real-Time Clock (RTC) Mode	219
, ,	
· · · ·	
· · · · · · · · · · · · · · · · · · ·	
·	
-	
·	
· · · · · · · · · · · · · · · · · · ·	
·	
•	
· · · · · · · · · · · · · · · · · · ·	
· ·	
Test Modes	275
Internal Temperature Sensor	275
Register Map	276
Register Descriptions	277
Universal Asynchronous Receivers/Transmitters (UARTs)	303
Functional Description	304
Transmit/Receive Logic	304
· ·	
• •	
·	
·	
·	
Initialization and Configuration	
	General-Purpose Timers Block Diagram Functional Description GPTM Reset Conditions 32-Bit Timer Operating Modes 16-Bit Timer Operating Modes 16-Bit Timer Operating Modes 16-Bit Timer Operating Modes 32-Bit Real-Time Clock (RTC) Mode 16-Bit Noe-Shot/Periodic Timer Mode 32-Bit Real-Time Clock (RTC) Mode 16-Bit Input Edge Count Mode 16-Bit Input Edge Timing Mode 16-Bit Input Edge Timing Mode 16-Bit Input Edge Search Mode 16-Bit Input Edge Search Mode 16-Bit Input Edge Timing Mode 16-Bi

	Register Map	309
13.5	Register Descriptions	310
14	Synchronous Serial Interface (SSI)	344
14.1	Block Diagram	344
14.2	Functional Description	344
14.2.1	Bit Rate Generation	345
14.2.2	FIFO Operation	345
14.2.3	Interrupts	345
14.2.4	Frame Formats	346
14.3	Initialization and Configuration	353
14.4	Register Map	354
14.5	Register Descriptions	355
15	Inter-Integrated Circuit (I ² C) Interface	381
15.1	Block Diagram	
15.2	Functional Description	381
15.2.1	I ² C Bus Functional Overview	
15.2.2	Available Speed Modes	384
15.2.3	Interrupts	
15.2.4	·	
15.2.5	Command Sequence Flow Charts	
15.3	Initialization and Configuration	
15.4	Register Map	
15.5	Register Descriptions (I ² C Master)	
15.6	Register Descriptions (I2C Slave)	
16	Controller Area Network (CAN) Module	416
16.1	Controller Area Network Overview	
16.2	Controller Area Network Features	416
	Controller Area Network Features	
16.3	Controller Area Network Block Diagram	417
16.3 16.4		417 417
16.2 16.3 16.4 16.4.1 16.4.2	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization	417 417 418
16.3 16.4 16.4.1 16.4.2	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation	417 417 418 418
16.3 16.4 16.4.1 16.4.2 16.4.3	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization	417 417 418 418 419
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects	417 417 418 418 419 419
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object	417 418 418 419 419 420
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object	417 418 418 419 419 420 420
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame	417 418 418 419 419 420 420 420
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame	417 418 418 419 419 420 420 420
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.9	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receive/Transmit Priority	417 418 418 419 420 420 420 420 421
16.3 16.4 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.9	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame	417 418 418 419 420 420 420 421 421
16.3 16.4 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.9 16.4.10	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame Receive/Transmit Priority Configuring a Receive Message Object	417 418 418 419 420 420 421 421 421
16.3 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.10 16.4.11	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame Receive/Transmit Priority Configuring a Receive Message Object Handling of Received Message Objects Handling of Interrupts	417 418 418 419 420 420 420 421 421 421 422 422
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.10 16.4.11	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame Receive/Transmit Priority Configuring a Receive Message Object Handling of Received Message Objects	417 418 418 419 420 420 421 421 421 422 422 423
16.3 16.4 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.10 16.4.11 16.4.13	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame Receive/Transmit Priority Configuring a Receive Message Object Handling of Received Message Objects Handling of Interrupts Bit Timing Configuration Error Considerations Bit Time and Bit Rate	417 418 418 419 420 420 421 421 422 423 423 423
16.3 16.4 16.4.1 16.4.2 16.4.3 16.4.4 16.4.5 16.4.6 16.4.7 16.4.8 16.4.10 16.4.11 16.4.13	Controller Area Network Block Diagram Controller Area Network Functional Description Initialization Operation Transmitting Message Objects Configuring a Transmit Message Object Updating a Transmit Message Object Accepting Received Message Objects Receiving a Data Frame Receiving a Remote Frame Receive/Transmit Priority Configuring a Receive Message Object Handling of Received Message Objects Handling of Interrupts Bit Timing Configuration Error Considerations	417 418 418 419 420 420 421 421 421 422 423 423 425

17	Ethernet Controller	
17.1	Block Diagram	458
17.2	Functional Description	458
17.2.1	Internal MII Operation	459
17.2.2	PHY Configuration/Operation	459
17.2.3	MAC Configuration/Operation	
17.2.4	Interrupts	
17.3	Initialization and Configuration	463
17.4	Ethernet Register Map	
17.5	Ethernet MAC Register Descriptions	
17.6	MII Management Register Descriptions	483
18	Analog Comparators	502
18.1	Block Diagram	503
18.2	Functional Description	503
18.2.1	Internal Reference Programming	505
18.3	Initialization and Configuration	506
18.4	Register Map	506
18.5	Register Descriptions	507
19	Pin Diagram	515
20	Signal Tables	517
20.1	100-Pin LQFP Package Pin Tables	
20.2	108-Pin BGA Package Pin Tables	
24	Operating Characteristics	543
21	Operating Characteristics	
	. •	
22	Electrical Characteristics	544
	Electrical Characteristics	544 544
22 22.1 22.1.1	Electrical Characteristics	544 544 544
22 22.1 22.1.1 22.1.2	Electrical Characteristics	544 544 544 544
22 22.1 22.1.1 22.1.2 22.1.3	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics	544 544 544 544 545
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4	Electrical Characteristics	544 544 544 544 545 545
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications	544 544 544 545 545 545
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics	544 544 544 545 545 547 547
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation	544 544 544 545 545 547 547
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics	544 544 544 545 545 547 547 547
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter	544 544 544 545 545 547 547 547 547 547
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator	544 544 545 545 545 547 547 547 547 548 549
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C	544 544 545 545 547 547 547 547 547 548 549 549
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller	544 544 544 545 547 547 547 547 547 548 549 550
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module	544 544 544 545 547 547 547 547 547 548 549 550 553
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I²C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI)	544 544 544 545 547 547 547 547 548 549 550 553 553
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8 22.2.8	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI) JTAG and Boundary Scan	544 544 544 545 547 547 547 547 548 549 550 553 553
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8 22.2.8 22.2.9 22.2.10	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI) JTAG and Boundary Scan General-Purpose I/O	544 544 544 545 547 547 547 547 547 549 550 553 555 556
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8 22.2.8 22.2.9 22.2.10	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI) JTAG and Boundary Scan General-Purpose I/O Reset	544 544 544 545 547 547 547 547 548 549 550 553 555 556 557
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8 22.2.8 22.2.9 22.2.10	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI) JTAG and Boundary Scan General-Purpose I/O	544 544 544 545 547 547 547 547 548 549 550 553 555 556 557
22 22.1 22.1.1 22.1.2 22.1.3 22.1.4 22.1.5 22.1.6 22.2 22.2.1 22.2.2 22.2.3 22.2.4 22.2.5 22.2.6 22.2.7 22.2.8 22.2.8 22.2.9 22.2.10 22.2.11	Electrical Characteristics DC Characteristics Maximum Ratings Recommended DC Operating Conditions On-Chip Low Drop-Out (LDO) Regulator Characteristics Power Specifications Flash Memory Characteristics Hibernation AC Characteristics Load Conditions Clocks Analog-to-Digital Converter Analog Comparator I ² C Ethernet Controller Hibernation Module Synchronous Serial Interface (SSI) JTAG and Boundary Scan General-Purpose I/O Reset	544 544 544 545 547 547 547 547 547 548 549 550 553 555 556 557 559

A.2	Interfaces	563
A.2.1	UART	563
A.2.2	SSI	563
A.3	Packet Handling	564
A.3.1	Packet Format	564
A.3.2	Sending Packets	564
A.3.3	Receiving Packets	564
A.4	Commands	565
A.4.1	COMMAND_PING (0X20)	565
A.4.2	COMMAND_GET_STATUS (0x23)	565
A.4.3	COMMAND_DOWNLOAD (0x21)	565
A.4.4	COMMAND_SEND_DATA (0x24)	566
A.4.5	COMMAND_RUN (0x22)	566
A.4.6	COMMAND_RESET (0x25)	566
В	Register Quick Reference	568
С	Ordering and Contact Information	587
C.1	Ordering Information	
C.2	Kits	587
C.3	Company Information	588
C.4	Support Information	588

List of Figures

Figure 1-1.	Stellaris® 8000 Series High-Level Block Diagram	31
Figure 2-1.	CPU Block Diagram	40
Figure 2-2.	TPIU Block Diagram	41
Figure 5-1.	JTAG Module Block Diagram	51
Figure 5-2.	Test Access Port State Machine	54
Figure 5-3.	IDCODE Register Format	59
Figure 5-4.	BYPASS Register Format	60
Figure 5-5.	Boundary Scan Register Format	60
Figure 6-1.	External Circuitry to Extend Reset	62
Figure 6-2.	Power Architecture	65
Figure 6-3.	Main Clock Tree	67
Figure 7-1.	Hibernation Module Block Diagram	126
Figure 7-2.	Clock Source Using Crystal	127
Figure 7-3.	Clock Source Using Dedicated Oscillator	128
Figure 8-1.	Flash Block Diagram	145
Figure 9-1.	GPIO Port Block Diagram	170
Figure 9-2.	GPIODATA Write Example	171
Figure 9-3.	GPIODATA Read Example	171
Figure 10-1.	GPTM Module Block Diagram	212
Figure 10-2.	16-Bit Input Edge Count Mode Example	216
Figure 10-3.	16-Bit Input Edge Time Mode Example	217
Figure 10-4.	16-Bit PWM Mode Example	218
Figure 11-1.	WDT Module Block Diagram	247
Figure 12-1.	ADC Module Block Diagram	271
Figure 12-2.	Differential Sampling Range, V _{IN_ODD} = 1.5 V	274
Figure 12-3.	Differential Sampling Range, V _{IN ODD} = 0.75 V	
Figure 12-4.	Differential Sampling Range, V _{IN ODD} = 2.25 V	
Figure 12-5.	Internal Temperature Sensor Characteristic	
Figure 13-1.	UART Module Block Diagram	
Figure 13-2.	UART Character Frame	305
Figure 13-3.	IrDA Data Modulation	307
Figure 14-1.	SSI Module Block Diagram	344
Figure 14-2.	TI Synchronous Serial Frame Format (Single Transfer)	347
Figure 14-3.	TI Synchronous Serial Frame Format (Continuous Transfer)	347
Figure 14-4.	Freescale SPI Format (Single Transfer) with SPO=0 and SPH=0	348
Figure 14-5.	Freescale SPI Format (Continuous Transfer) with SPO=0 and SPH=0	348
Figure 14-6.	Freescale SPI Frame Format with SPO=0 and SPH=1	349
Figure 14-7.	Freescale SPI Frame Format (Single Transfer) with SPO=1 and SPH=0	350
Figure 14-8.	Freescale SPI Frame Format (Continuous Transfer) with SPO=1 and SPH=0	350
Figure 14-9.	Freescale SPI Frame Format with SPO=1 and SPH=1	351
Figure 14-10.	MICROWIRE Frame Format (Single Frame)	352
Figure 14-11.	MICROWIRE Frame Format (Continuous Transfer)	353
Figure 14-12.	MICROWIRE Frame Format, SSIFss Input Setup and Hold Requirements	353
Figure 15-1.	I ² C Block Diagram	
Figure 15-2.	I ² C Bus Configuration	382

Figure 15-3.	START and STOP Conditions	. 382
Figure 15-4.	Complete Data Transfer with a 7-Bit Address	. 383
Figure 15-5.	R/S Bit in First Byte	. 383
Figure 15-6.	Data Validity During Bit Transfer on the I ² C Bus	. 383
Figure 15-7.	Master Single SEND	386
Figure 15-8.	Master Single RECEIVE	. 387
Figure 15-9.	Master Burst SEND	388
Figure 15-10.	Master Burst RECEIVE	389
Figure 15-11.	Master Burst RECEIVE after Burst SEND	. 390
Figure 15-12.	Master Burst SEND after Burst RECEIVE	. 391
Figure 15-13.	Slave Command Sequence	. 392
Figure 16-1.	CAN Module Block Diagram	417
Figure 16-2.	CAN Bit Time	424
Figure 17-1.	Ethernet Controller Block Diagram	458
Figure 17-2.	Ethernet Controller	. 458
Figure 17-3.	Ethernet Frame	460
Figure 18-1.	Analog Comparator Module Block Diagram	503
Figure 18-2.	Structure of Comparator Unit	. 504
Figure 18-3.	Comparator Internal Reference Structure	505
Figure 19-1.	100-Pin LQFP Package Pin Diagram	. 515
Figure 19-2.	108-Ball BGA Package Pin Diagram (Top View)	. 516
Figure 22-1.	Load Conditions	547
Figure 22-2.	I ² C Timing	550
Figure 22-3.	External XTLP Oscillator Characteristics	552
Figure 22-4.	Hibernation Module Timing	. 553
Figure 22-5.	SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing Measurement	. 554
Figure 22-6.	SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer	. 554
Figure 22-7.	SSI Timing for SPI Frame Format (FRF=00), with SPH=1	. 555
Figure 22-8.	JTAG Test Clock Input Timing	. 556
Figure 22-9.	JTAG Test Access Port (TAP) Timing	. 556
-	JTAG TRST Timing	
Figure 22-11.	External Reset Timing (RST)	. 557
Figure 22-12.	Power-On Reset Timing	558
•	Brown-Out Reset Timing	
Figure 22-14.	Software Reset Timing	. 558
Figure 22-15.	Watchdog Reset Timing	558
•	100-Pin LQFP Package	
Figure 23-2.	108-Ball BGA Package	561

List of Tables

Table 1.	Revision History	20
Table 2.	Documentation Conventions	22
Table 3-1.	Memory Map	45
Table 4-1.	Exception Types	47
Table 4-2.	Interrupts	48
Table 5-1.	JTAG Port Pins Reset State	52
Table 5-2.	JTAG Instruction Register Commands	57
Table 6-1.	System Control Register Map	70
Table 7-1.	Hibernation Module Register Map	132
Table 8-1.	Flash Protection Policy Combinations	146
Table 8-2.	Flash Resident Registers	148
Table 8-3.	Flash Register Map	149
Table 9-1.	GPIO Pad Configuration Examples	173
Table 9-2.	GPIO Interrupt Configuration Example	173
Table 9-3.	GPIO Register Map	175
Table 10-1.	Available CCP Pins	212
Table 10-2.	16-Bit Timer With Prescaler Configurations	215
Table 10-3.	Timers Register Map	221
Table 11-1.	Watchdog Timer Register Map	248
Table 12-1.	Samples and FIFO Depth of Sequencers	271
Table 12-2.	Differential Sampling Pairs	273
Table 12-3.	ADC Register Map	276
Table 13-1.	UART Register Map	309
Table 14-1.	SSI Register Map	354
Table 15-1.	Examples of I ² C Master Timer Period versus Speed Mode	384
Table 15-2.	Inter-Integrated Circuit (I ² C) Interface Register Map	393
Table 15-3.	Write Field Decoding for I2CMCS[3:0] Field (Sheet 1 of 3)	398
Table 16-1.	Transmit Message Object Bit Settings	419
Table 16-2.	Receive Message Object Bit Settings	421
Table 16-3.	CAN Protocol Ranges	424
Table 16-4.	CAN Register Map	427
Table 17-1.	TX & RX FIFO Organization	461
Table 17-2.	Ethernet Register Map	464
Table 18-1.	Comparator 0 Operating Modes	504
Table 18-2.	Comparator 1 Operating Modes	504
Table 18-3.	Comparator 2 Operating Modes	505
Table 18-4.	Internal Reference Voltage and ACREFCTL Field Values	505
Table 18-5.	Analog Comparators Register Map	507
Table 20-1.	Signals by Pin Number	517
Table 20-2.	Signals by Signal Name	521
Table 20-3.	Signals by Function, Except for GPIO	525
Table 20-4.	GPIO Pins and Alternate Functions	528
Table 20-5.	Signals by Pin Number	529
Table 20-6.	Signals by Signal Name	534
Table 20-7.	Signals by Function, Except for GPIO	538
Table 20-8.	GPIO Pins and Alternate Functions	541

Table 21-1.	Temperature Characteristics	543
Table 21-2.	Thermal Characteristics	543
Table 22-1.	Maximum Ratings	544
Table 22-2.	Recommended DC Operating Conditions	544
Table 22-3.	LDO Regulator Characteristics	545
Table 22-4.	Detailed Power Specifications	546
Table 22-5.	Flash Memory Characteristics	547
Table 22-6.	Hibernation Module DC Characteristics	547
Table 22-7.	Phase Locked Loop (PLL) Characteristics	547
Table 22-8.	Clock Characteristics	548
Table 22-9.	Crystal Characteristics	548
Table 22-10.	ADC Characteristics	548
Table 22-11.	Analog Comparator Characteristics	549
Table 22-12.	Analog Comparator Voltage Reference Characteristics	549
Table 22-13.	I ² C Characteristics	549
Table 22-14.	100BASE-TX Transmitter Characteristics	550
Table 22-15.	100BASE-TX Transmitter Characteristics (informative)	550
Table 22-16.	100BASE-TX Receiver Characteristics	550
Table 22-17.	10BASE-T Transmitter Characteristics	550
Table 22-18.	10BASE-T Transmitter Characteristics (informative)	551
Table 22-19.	10BASE-T Receiver Characteristics	551
Table 22-20.	Isolation Transformers	551
Table 22-21.	Ethernet Reference Crystal	552
Table 22-22.	External XTLP Oscillator Characteristics	552
Table 22-23.	Hibernation Module AC Characteristics	553
Table 22-24.	SSI Characteristics	553
Table 22-25.	JTAG Characteristics	555
Table 22-26.	GPIO Characteristics	557
Table 22-27.	Reset Characteristics	557
Table C-1.	Part Ordering Information	587

List of Registers

System Cor	ntrol	
Register 1:	Device Identification 0 (DID0), offset 0x000	72
Register 2:	Brown-Out Reset Control (PBORCTL), offset 0x030	74
Register 3:	LDO Power Control (LDOPCTL), offset 0x034	75
Register 4:	Raw Interrupt Status (RIS), offset 0x050	76
Register 5:	Interrupt Mask Control (IMC), offset 0x054	77
Register 6:	Masked Interrupt Status and Clear (MISC), offset 0x058	78
Register 7:	Reset Cause (RESC), offset 0x05C	
Register 8:	Run-Mode Clock Configuration (RCC), offset 0x060	80
Register 9:	XTAL to PLL Translation (PLLCFG), offset 0x064	84
Register 10:	Run-Mode Clock Configuration 2 (RCC2), offset 0x070	
Register 11:	Deep Sleep Clock Configuration (DSLPCLKCFG), offset 0x144	87
Register 12:	Device Identification 1 (DID1), offset 0x004	88
Register 13:	Device Capabilities 0 (DC0), offset 0x008	90
Register 14:	Device Capabilities 1 (DC1), offset 0x010	91
Register 15:	Device Capabilities 2 (DC2), offset 0x014	93
Register 16:	Device Capabilities 3 (DC3), offset 0x018	95
Register 17:	Device Capabilities 4 (DC4), offset 0x01C	97
Register 18:	Run Mode Clock Gating Control Register 0 (RCGC0), offset 0x100	99
Register 19:	Sleep Mode Clock Gating Control Register 0 (SCGC0), offset 0x110	101
Register 20:	Deep Sleep Mode Clock Gating Control Register 0 (DCGC0), offset 0x120	103
Register 21:	Run Mode Clock Gating Control Register 1 (RCGC1), offset 0x104	105
Register 22:	Sleep Mode Clock Gating Control Register 1 (SCGC1), offset 0x114	108
Register 23:	Deep Sleep Mode Clock Gating Control Register 1 (DCGC1), offset 0x124	111
Register 24:	Run Mode Clock Gating Control Register 2 (RCGC2), offset 0x108	114
Register 25:	Sleep Mode Clock Gating Control Register 2 (SCGC2), offset 0x118	116
Register 26:	Deep Sleep Mode Clock Gating Control Register 2 (DCGC2), offset 0x128	118
Register 27:	Software Reset Control 0 (SRCR0), offset 0x040	120
Register 28:	Software Reset Control 1 (SRCR1), offset 0x044	121
Register 29:	Software Reset Control 2 (SRCR2), offset 0x048	123
Hibernation	Module	125
Register 1:	Hibernation RTC Counter (HIBRTCC), offset 0x000	133
Register 2:	Hibernation RTC Match 0 (HIBRTCM0), offset 0x004	
Register 3:	Hibernation RTC Match 1 (HIBRTCM1), offset 0x008	135
Register 4:	Hibernation RTC Load (HIBRTCLD), offset 0x00C	136
Register 5:	Hibernation Control (HIBCTL), offset 0x010	
Register 6:	Hibernation Interrupt Mask (HIBIM), offset 0x014	139
Register 7:	Hibernation Raw Interrupt Status (HIBRIS), offset 0x018	
Register 8:	Hibernation Masked Interrupt Status (HIBMIS), offset 0x01C	141
Register 9:	Hibernation Interrupt Clear (HIBIC), offset 0x020	142
Register 10:	Hibernation RTC Trim (HIBRTCT), offset 0x024	143
Register 11:	Hibernation Data (HIBDATA), offset 0x030-0x12C	144
Internal Me	mory	
Register 1:	Flash Memory Address (FMA), offset 0x000	150
Register 2:	Flash Memory Data (FMD), offset 0x004	151

Register 3:	Flash Memory Control (FMC), offset 0x008	152
Register 4:	Flash Controller Raw Interrupt Status (FCRIS), offset 0x00C	154
Register 5:	Flash Controller Interrupt Mask (FCIM), offset 0x010	155
Register 6:	Flash Controller Masked Interrupt Status and Clear (FCMISC), offset 0x014	156
Register 7:	USec Reload (USECRL), offset 0x140	157
Register 8:	Flash Memory Protection Read Enable 0 (FMPRE0), offset 0x130 and 0x200	158
Register 9:	Flash Memory Protection Program Enable 0 (FMPPE0), offset 0x134 and 0x400	159
Register 10:	User Debug (USER_DBG), offset 0x1D0	160
Register 11:	User Register 0 (USER_REG0), offset 0x1E0	161
Register 12:	User Register 1 (USER_REG1), offset 0x1E4	162
Register 13:	Flash Memory Protection Read Enable 1 (FMPRE1), offset 0x204	163
Register 14:	Flash Memory Protection Read Enable 2 (FMPRE2), offset 0x208	164
Register 15:	Flash Memory Protection Read Enable 3 (FMPRE3), offset 0x20C	165
Register 16:	Flash Memory Protection Program Enable 1 (FMPPE1), offset 0x404	166
Register 17:	Flash Memory Protection Program Enable 2 (FMPPE2), offset 0x408	167
Register 18:	Flash Memory Protection Program Enable 3 (FMPPE3), offset 0x40C	168
General-Pu	rpose Input/Outputs (GPIOs)	169
Register 1:	GPIO Data (GPIODATA), offset 0x000	
Register 2:	GPIO Direction (GPIODIR), offset 0x400	
Register 3:	GPIO Interrupt Sense (GPIOIS), offset 0x404	
Register 4:	GPIO Interrupt Both Edges (GPIOIBE), offset 0x408	
Register 5:	GPIO Interrupt Event (GPIOIEV), offset 0x40C	
Register 6:	GPIO Interrupt Mask (GPIOIM), offset 0x410	
Register 7:	GPIO Raw Interrupt Status (GPIORIS), offset 0x414	
Register 8:	GPIO Masked Interrupt Status (GPIOMIS), offset 0x418	
Register 9:	GPIO Interrupt Clear (GPIOICR), offset 0x41C	
Register 10:	GPIO Alternate Function Select (GPIOAFSEL), offset 0x420	
Register 11:	GPIO 2-mA Drive Select (GPIODR2R), offset 0x500	
Register 12:	GPIO 4-mA Drive Select (GPIODR4R), offset 0x504	
Register 13:	GPIO 8-mA Drive Select (GPIODR8R), offset 0x508	
Register 14:	GPIO Open Drain Select (GPIOODR), offset 0x50C	
Register 15:	GPIO Pull-Up Select (GPIOPUR), offset 0x510	
Register 16:	GPIO Pull-Down Select (GPIOPDR), offset 0x514	
Register 17:	GPIO Slew Rate Control Select (GPIOSLR), offset 0x518	
Register 18:	GPIO Digital Enable (GPIODEN), offset 0x51C	
Register 19:	GPIO Lock (GPIOLOCK), offset 0x520	
Register 20:	GPIO Commit (GPIOCR), offset 0x524	
Register 21:	GPIO Peripheral Identification 4 (GPIOPeriphID4), offset 0xFD0	
Register 22:	GPIO Peripheral Identification 5 (GPIOPeriphID5), offset 0xFD4	
Register 23:	GPIO Peripheral Identification 6 (GPIOPeriphID6), offset 0xFD8	
Register 24:	GPIO Peripheral Identification 7 (GPIOPeriphID7), offset 0xFDC	202
Register 25:	GPIO Peripheral Identification 0 (GPIOPeriphID0), offset 0xFE0	203
Register 26:	GPIO Peripheral Identification 1 (GPIOPeriphID1), offset 0xFE4	
Register 27:	GPIO Peripheral Identification 2 (GPIOPeriphID2), offset 0xFE8	
Register 28:	GPIO Peripheral Identification 3 (GPIOPeriphID3), offset 0xFEC	
Register 29:	GPIO PrimeCell Identification 0 (GPIOPCellID0), offset 0xFF0	
Register 30:	GPIO PrimeCell Identification 1 (GPIOPCellID1), offset 0xFF4	
Register 31:	GPIO PrimeCell Identification 2 (GPIOPCellID2), offset 0xFF8	

Register 32:	GPIO PrimeCell Identification 3 (GPIOPCellID3), offset 0xFFC	210
General-Pu	rpose Timers	211
Register 1:	GPTM Configuration (GPTMCFG), offset 0x000	
Register 2:	GPTM TimerA Mode (GPTMTAMR), offset 0x004	224
Register 3:	GPTM TimerB Mode (GPTMTBMR), offset 0x008	226
Register 4:	GPTM Control (GPTMCTL), offset 0x00C	228
Register 5:	GPTM Interrupt Mask (GPTMIMR), offset 0x018	231
Register 6:	GPTM Raw Interrupt Status (GPTMRIS), offset 0x01C	
Register 7:	GPTM Masked Interrupt Status (GPTMMIS), offset 0x020	234
Register 8:	GPTM Interrupt Clear (GPTMICR), offset 0x024	235
Register 9:	GPTM TimerA Interval Load (GPTMTAILR), offset 0x028	237
Register 10:	GPTM TimerB Interval Load (GPTMTBILR), offset 0x02C	238
Register 11:	GPTM TimerA Match (GPTMTAMATCHR), offset 0x030	239
Register 12:	GPTM TimerB Match (GPTMTBMATCHR), offset 0x034	240
Register 13:	GPTM TimerA Prescale (GPTMTAPR), offset 0x038	241
Register 14:	GPTM TimerB Prescale (GPTMTBPR), offset 0x03C	242
Register 15:	GPTM TimerA Prescale Match (GPTMTAPMR), offset 0x040	243
Register 16:	GPTM TimerB Prescale Match (GPTMTBPMR), offset 0x044	244
Register 17:	GPTM TimerA (GPTMTAR), offset 0x048	245
Register 18:	GPTM TimerB (GPTMTBR), offset 0x04C	246
Watchdog 1	Timer	247
Register 1:	Watchdog Load (WDTLOAD), offset 0x000	
Register 2:	Watchdog Value (WDTVALUE), offset 0x004	
Register 3:	Watchdog Control (WDTCTL), offset 0x008	
Register 4:	Watchdog Interrupt Clear (WDTICR), offset 0x00C	
Register 5:	Watchdog Raw Interrupt Status (WDTRIS), offset 0x010	
Register 6:	Watchdog Masked Interrupt Status (WDTMIS), offset 0x014	
Register 7:	Watchdog Test (WDTTEST), offset 0x418	
Register 8:	Watchdog Lock (WDTLOCK), offset 0xC00	
Register 9:	Watchdog Peripheral Identification 4 (WDTPeriphID4), offset 0xFD0	258
Register 10:	Watchdog Peripheral Identification 5 (WDTPeriphID5), offset 0xFD4	259
Register 11:	Watchdog Peripheral Identification 6 (WDTPeriphID6), offset 0xFD8	260
Register 12:	Watchdog Peripheral Identification 7 (WDTPeriphID7), offset 0xFDC	261
Register 13:	Watchdog Peripheral Identification 0 (WDTPeriphID0), offset 0xFE0	262
Register 14:	Watchdog Peripheral Identification 1 (WDTPeriphID1), offset 0xFE4	263
Register 15:	Watchdog Peripheral Identification 2 (WDTPeriphID2), offset 0xFE8	264
Register 16:	Watchdog Peripheral Identification 3 (WDTPeriphID3), offset 0xFEC	265
Register 17:	Watchdog PrimeCell Identification 0 (WDTPCellID0), offset 0xFF0	266
Register 18:	Watchdog PrimeCell Identification 1 (WDTPCellID1), offset 0xFF4	267
Register 19:	Watchdog PrimeCell Identification 2 (WDTPCellID2), offset 0xFF8	268
Register 20:	Watchdog PrimeCell Identification 3 (WDTPCellID3), offset 0xFFC	269
Analog-to-D	Digital Converter (ADC)	270
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	286
Register 8:	ADC Sample Sequencer Priority (ADCSSPRI), offset 0x020	287
Register 9:	ADC Processor Sample Sequence Initiate (ADCPSSI), offset 0x028	288
Register 10:	ADC Sample Averaging Control (ADCSAC), offset 0x030	289
Register 11:	ADC Sample Sequence Input Multiplexer Select 0 (ADCSSMUX0), offset 0x040	290
Register 12:	ADC Sample Sequence Control 0 (ADCSSCTL0), offset 0x044	292
Register 13:	ADC Sample Sequence Result FIFO 0 (ADCSSFIFO0), offset 0x048	295
Register 14:	ADC Sample Sequence Result FIFO 1 (ADCSSFIFO1), offset 0x068	295
Register 15:	ADC Sample Sequence Result FIFO 2 (ADCSSFIFO2), offset 0x088	295
Register 16:	ADC Sample Sequence Result FIFO 3 (ADCSSFIFO3), offset 0x0A8	295
Register 17:	ADC Sample Sequence FIFO 0 Status (ADCSSFSTAT0), offset 0x04C	296
Register 18:	ADC Sample Sequence FIFO 1 Status (ADCSSFSTAT1), offset 0x06C	296
Register 19:	ADC Sample Sequence FIFO 2 Status (ADCSSFSTAT2), offset 0x08C	296
Register 20:	ADC Sample Sequence FIFO 3 Status (ADCSSFSTAT3), offset 0x0AC	296
Register 21:	ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1), offset 0x060	297
Register 22:	ADC Sample Sequence Input Multiplexer Select 2 (ADCSSMUX2), offset 0x080	297
Register 23:	ADC Sample Sequence Control 1 (ADCSSCTL1), offset 0x064	
Register 24:	ADC Sample Sequence Control 2 (ADCSSCTL2), offset 0x084	
Register 25:	ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3), offset 0x0A0	
Register 26:	ADC Sample Sequence Control 3 (ADCSSCTL3), offset 0x0A4	
Register 27:	ADC Test Mode Loopback (ADCTMLB), offset 0x100	302
Universal A	Asynchronous Receivers/Transmitters (UARTs)	303
Register 1:	UART Data (UARTDR), offset 0x000	
Register 2:	UART Receive Status/Error Clear (UARTRSR/UARTECR), offset 0x004	313
Register 3:	UART Flag (UARTFR), offset 0x018	315
Register 4:	UART IrDA Low-Power Register (UARTILPR), offset 0x020	317
Register 5:	UART Integer Baud-Rate Divisor (UARTIBRD), offset 0x024	318
Register 6:	UART Fractional Baud-Rate Divisor (UARTFBRD), offset 0x028	
Register 7:	UART Line Control (UARTLCRH), offset 0x02C	
Register 8:	UART Control (UARTCTL), offset 0x030	
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	
Register 13:	UART Interrupt Clear (UARTICR), offset 0x044	
Register 14:	UART Peripheral Identification 4 (UARTPeriphID4), offset 0xFD0	
Register 15:	UART Peripheral Identification 5 (UARTPeriphID5), offset 0xFD4	
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	343

Synchronou	us Serial Interface (SSI)	344
Register 1:	SSI Control 0 (SSICR0), offset 0x000	356
Register 2:	SSI Control 1 (SSICR1), offset 0x004	358
Register 3:	SSI Data (SSIDR), offset 0x008	360
Register 4:	SSI Status (SSISR), offset 0x00C	361
Register 5:	SSI Clock Prescale (SSICPSR), offset 0x010	363
Register 6:	SSI Interrupt Mask (SSIIM), offset 0x014	364
Register 7:	SSI Raw Interrupt Status (SSIRIS), offset 0x018	366
Register 8:	SSI Masked Interrupt Status (SSIMIS), offset 0x01C	367
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	370
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	373
Register 15:	SSI Peripheral Identification 1 (SSIPeriphID1), offset 0xFE4	374
Register 16:	SSI Peripheral Identification 2 (SSIPeriphID2), offset 0xFE8	375
Register 17:	SSI Peripheral Identification 3 (SSIPeriphID3), offset 0xFEC	376
Register 18:	SSI PrimeCell Identification 0 (SSIPCelIID0), offset 0xFF0	377
Register 19:	SSI PrimeCell Identification 1 (SSIPCelIID1), offset 0xFF4	
Register 20:	SSI PrimeCell Identification 2 (SSIPCelIID2), offset 0xFF8	
Register 21:	SSI PrimeCell Identification 3 (SSIPCelIID3), offset 0xFFC	380
Inter-Integra	ated Circuit (I ² C) Interface	381
Register 1:	I ² C Master Slave Address (I2CMSA), offset 0x000	395
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 0x000	
Register 11:	I ² C Slave Control/Status (I2CSCSR), offset 0x004	
Register 12:	I ² C Slave Data (I2CSDR), offset 0x008	
Register 13:	I ² C Slave Interrupt Mask (I2CSIMR), offset 0x00C	
Register 14:	I ² C Slave Raw Interrupt Status (I2CSRIS), offset 0x010	
Register 15:	I ² C Slave Masked Interrupt Status (I2CSMIS), offset 0x014	
•	1 ² C Slave Interrupt Clear (I2CSICR), offset 0x018	
Register 16:		
	Area Network (CAN) Module	
Register 1:	CAN Control (CANCTL), offset 0x000	
Register 2:	CAN Status (CANSTS), offset 0x004	
Register 3:	CAN Bit Timing (CANBIT) offset 0x008	
Register 4:	CAN later wat (CANINT), offset 0x00C	
Register 5:	CAN Took (CANTST), offset 0x010	
Register 6:	CAN Board Bata Prescalar Extension (CANBERE) affect 0x018	
Register 7:	CAN Baud Rate Prescalar Extension (CANBRPE), offset 0x018	440

Register 8:	CAN IF1 Command Request (CANIF1CRQ), offset 0x020	441
Register 9:	CAN IF2 Command Request (CANIF2CRQ), offset 0x080	441
Register 10:	CAN IF1 Command Mask (CANIF1CMSK), offset 0x024	442
Register 11:	CAN IF2 Command Mask (CANIF2CMSK), offset 0x084	442
Register 12:	CAN IF1 Mask 1 (CANIF1MSK1), offset 0x028	445
Register 13:	CAN IF2 Mask 1 (CANIF2MSK1), offset 0x088	445
Register 14:	CAN IF1 Mask 2 (CANIF1MSK2), offset 0x02C	446
Register 15:	CAN IF2 Mask 2 (CANIF2MSK2), offset 0x08C	446
Register 16:	CAN IF1 Arbitration 1 (CANIF1ARB1), offset 0x030	447
Register 17:	CAN IF2 Arbitration 1 (CANIF2ARB1), offset 0x090	447
Register 18:	CAN IF1 Arbitration 2 (CANIF1ARB2), offset 0x034	448
Register 19:	CAN IF2 Arbitration 2 (CANIF2ARB2), offset 0x094	448
Register 20:	CAN IF1 Message Control (CANIF1MCTL), offset 0x038	450
Register 21:	CAN IF2 Message Control (CANIF2MCTL), offset 0x098	450
Register 22:	CAN IF1 Data A1 (CANIF1DA1), offset 0x03C	452
Register 23:	CAN IF1 Data A2 (CANIF1DA2), offset 0x040	452
Register 24:	CAN IF1 Data B1 (CANIF1DB1), offset 0x044	452
Register 25:	CAN IF1 Data B2 (CANIF1DB2), offset 0x048	452
Register 26:	CAN IF2 Data A1 (CANIF2DA1), offset 0x09C	452
Register 27:	CAN IF2 Data A2 (CANIF2DA2), offset 0x0A0	452
Register 28:	CAN IF2 Data B1 (CANIF2DB1), offset 0x0A4	452
Register 29:	CAN IF2 Data B2 (CANIF2DB2), offset 0x0A8	452
Register 30:	CAN Transmission Request 1 (CANTXRQ1), offset 0x100	453
Register 31:	CAN Transmission Request 2 (CANTXRQ2), offset 0x104	453
Register 32:	CAN New Data 1 (CANNWDA1), offset 0x120	454
Register 33:	CAN New Data 2 (CANNWDA2), offset 0x124	454
Register 34:	CAN Message 1 Interrupt Pending (CANMSG1INT), offset 0x140	455
Register 35:	CAN Message 2 Interrupt Pending (CANMSG2INT), offset 0x144	455
Register 36:	CAN Message 1 Valid (CANMSG1VAL), offset 0x160	456
Register 37:	CAN Message 2 Valid (CANMSG2VAL), offset 0x164	456
Ethernet Co	ontroller	457
Register 1:	Ethernet MAC Raw Interrupt Status (MACRIS), offset 0x000	
Register 2:	Ethernet MAC Interrupt Acknowledge (MACIACK), offset 0x000	
Register 3:	Ethernet MAC Interrupt Mask (MACIM), offset 0x004	
Register 4:	Ethernet MAC Receive Control (MACRCTL), offset 0x008	
Register 5:	Ethernet MAC Transmit Control (MACTCTL), offset 0x00C	
Register 6:	Ethernet MAC Data (MACDATA), offset 0x010	
Register 7:	Ethernet MAC Individual Address 0 (MACIA0), offset 0x014	
Register 8:	Ethernet MAC Individual Address 1 (MACIA1), offset 0x018	
Register 9:	Ethernet MAC Threshold (MACTHR), offset 0x01C	
Register 10:	Ethernet MAC Management Control (MACMCTL), offset 0x020	
Register 11:	Ethernet MAC Management Divider (MACMDV), offset 0x024	
Register 12:	Ethernet MAC Management Transmit Data (MACMTXD), offset 0x02C	
Register 13:	Ethernet MAC Management Receive Data (MACMRXD), offset 0x030	
Register 14:	Ethernet MAC Number of Packets (MACNP), offset 0x034	
Register 15:	Ethernet MAC Transmission Request (MACTR), offset 0x038	
Register 16:	Ethernet MAC Timer Support (MACTS), offset 0x03C	
Register 17	Ethernet PHY Management Register 0 – Control (MR0), address 0x00	

Register 18:	Ethernet PHY Management Register 1 – Status (MR1), address 0x01	486
Register 19:	Ethernet PHY Management Register 2 – PHY Identifier 1 (MR2), address 0x02	488
Register 20:	Ethernet PHY Management Register 3 – PHY Identifier 2 (MR3), address 0x03	489
Register 21:	Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4), address 0x04	490
Register 22:	Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability (MR5), address 0x05	492
Register 23:	Ethernet PHY Management Register 6 – Auto-Negotiation Expansion (MR6), address 0x06	493
Register 24:	Ethernet PHY Management Register 16 – Vendor-Specific (MR16), address 0x10	494
Register 25:	Ethernet PHY Management Register 17 – Interrupt Control/Status (MR17), address 0x11	496
Register 26:	Ethernet PHY Management Register 18 – Diagnostic (MR18), address 0x12	498
Register 27:	Ethernet PHY Management Register 19 – Transceiver Control (MR19), address 0x13	
Register 28:	Ethernet PHY Management Register 23 – LED Configuration (MR23), address 0x17	
Register 29:		
Analog Cor	mparators	502
Register 1:	Analog Comparator Masked Interrupt Status (ACMIS), offset 0x00	
Register 2:	Analog Comparator Raw Interrupt Status (ACRIS), offset 0x04	
Register 3:	Analog Comparator Interrupt Enable (ACINTEN), offset 0x08	510
Register 4:	Analog Comparator Reference Voltage Control (ACREFCTL), offset 0x10	511
Register 5:	Analog Comparator Status 0 (ACSTAT0), offset 0x20	512
Register 6:	Analog Comparator Status 1 (ACSTAT1), offset 0x40	512
Register 7:	Analog Comparator Status 2 (ACSTAT2), offset 0x60	512
Register 8:	Analog Comparator Control 0 (ACCTL0), offset 0x24	513
Register 9:	Analog Comparator Control 1 (ACCTL1), offset 0x44	513
Register 10:	Analog Comparator Control 2 (ACCTL2), offset 0x64	513

Revision History

The revision history table notes changes made between the indicated revisions of the LM3S8933 data sheet.

Table 1. Revision History

Date	Revision	Description
March 2008	2550	Started tracking revision history.
April 2008	2881	■ The O _{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 datasheet. 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 T _{VDDRISE} parameter in the "Reset Characteristics" table in the "Electrical Characteristics" chapter was changed from a max of 100 to 250.
		■ The maximum value on Core supply voltage (V _{DD25}) in the "Maximum Ratings" table in the "Electrical Characteristics" chapter was changed from 4 to 3.
		■ The operational frequency of the internal 30-kHz oscillator clock source is 30 kHz ± 50% (prior datasheets 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 datasheets incorrectly noted 0x3 as a reserved value.
		■ The reset for bits 6:4 of the RCC2 register (OSCSRC2) is 0x1 (IOSC). Prior datasheets incorrectly noted the reset was 0x0 (MOSC).
		Two figures on clock source were added to the "Hibernation Module":
		Clock Source Using Crystal
		Clock Source Using Dedicated Oscillator
		The following notes on battery management were added to the "Hibernation Module" chapter:
		Battery voltage is not measured while in Hibernate mode.
		 System level factors may affect the accuracy of the low battery detect circuit. The designer should consider battery type, discharge characteristics, and a test load during battery voltage measurements.
		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.

Date	Revision	Description	
		The last revision of the datasheet (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 datasheet 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 PE5 to NC 	
		 Ball D1: Changed PE4 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 458).	
		Additional minor datasheet clarifications and corrections.	
July 2008	3108	Corrected resistor value in ERBIAS signal description.	
		Additional minor datasheet clarifications and corrections.	
August 2008	3447	Added note on clearing interrupts to Interrupts chapter.	
		Added Power Architecture diagram to System Control chapter.	
		Additional minor datasheet clarifications and corrections.	

About This Document

This data sheet provides reference information for the LM3S8933 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 documents are referenced by the data sheet, and available on the documentation CD or from the Luminary Micro web site at www.luminarymicro.com:

- ARM® Cortex™-M3 Technical Reference Manual
- ARM® CoreSight Technical Reference Manual
- ARM® v7-M Architecture Application Level Reference Manual
- Stellaris[®] Peripheral Driver Library User's Guide
- Stellaris[®] ROM User's Guide

The following related documents are also referenced:

■ IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture

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

Documentation Conventions

This document uses the conventions shown in Table 2 on page 22.

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 "Memory Map" on page 45.	

Notation	Meaning	
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/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.	
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		
X	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.	

Notation	Meaning
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 Luminary Micro 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[®] LM3S8000 series combines Bosch Controller Area Network technology with both a 10/100 Ethernet Media Access Control (MAC) and Physical (PHY) layer.

The LM3S8933 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.

For applications requiring extreme conservation of power, the LM3S8933 microcontroller features a Battery-backed Hibernation module to efficiently power down the LM3S8933 to a low-power state during extended periods of inactivity. With a power-up/power-down sequencer, a continuous time counter (RTC), a pair of match registers, an APB interface to the system bus, and dedicated non-volatile memory, the Hibernation module positions the LM3S8933 microcontroller perfectly for battery applications.

In addition, the LM3S8933 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 LM3S8933 microcontroller is code-compatible to all members of the extensive Stellaris® family; providing flexibility to fit our customers' precise needs.

Luminary Micro 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 587 for ordering information for Stellaris[®] family devices.

1.1 Product Features

The LM3S8933 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
- 32 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

Internal Memory

- 256 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
- 64 KB single-cycle SRAM

General-Purpose Timers

- Four General-Purpose Timer Modules (GPTM), each of which provides two 16-bit timers.
 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 in periodic and one-shot mode when the controller asserts the CPU Halt flag during debug
 - · ADC event trigger
- 16-bit Timer modes
 - General-purpose timer function with an 8-bit prescaler
 - · 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
- Controller Area Network (CAN)
 - Supports CAN protocol version 2.0 part A/B
 - Bit rates up to 1Mb/s
 - 32 message objects, each with its own identifier mask
 - Maskable interrupt
 - Disable automatic retransmission mode for TTCAN
 - Programmable loop-back mode for self-test operation
- 10/100 Ethernet Controller
 - Conforms to the IEEE 802.3-2002 Specification
 - Hardware assistance for IEEE 1588-2002 Precision Time Protocol (PTP)
 - Full- and half-duplex for both 100 Mbps and 10 Mbps operation
 - Integrated 10/100 Mbps Transceiver (PHY)
 - Automatic MDI/MDI-X cross-over correction
 - Programmable MAC address

- Power-saving and power-down modes
- 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

UART

- Two fully programmable 16C550-type UARTs with IrDA support
- Separate 16x8 transmit (TX) and 16x12 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
- False-start-bit detection
- Line-break generation and detection

ADC

- Single- and differential-input configurations
- Four 10-bit channels (inputs) when used as single-ended inputs
- Sample rate of one million samples/second
- Flexible, configurable analog-to-digital conversion
- Four programmable sample conversion sequences from one to eight entries long, with corresponding conversion result FIFOs
- Each sequence triggered by software or internal event (timers, analog comparators, or GPIO)
- On-chip temperature sensor
- Analog Comparators
 - Three 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

I²C

- Master and slave receive and transmit operation with transmission speed up to 100 Kbps in Standard mode and 400 Kbps in Fast mode
- Interrupt generation
- Master with arbitration and clock synchronization, multimaster support, and 7-bit addressing mode

GPIOs

- 6-36 GPIOs, depending on configuration
- 5-V-tolerant input/outputs
- Programmable interrupt generation as either edge-triggered or level-sensitive
- Low interrupt latency; as low as 6 cycles and never more than 12 cycles
- 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

Power

- On-chip Low Drop-Out (LDO) voltage regulator, with programmable output user-adjustable from 2.25 V to 2.75 V
- Hibernation module handles the power-up/down 3.3 V sequencing and control for the core digital logic and analog circuits
- Low-power options on controller: Sleep and Deep-sleep modes
- Low-power options for peripherals: software controls shutdown of individual peripherals

- User-enabled LDO unregulated voltage detection and automatic reset
- 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
- Additional Features
 - Six reset sources
 - Programmable clock source control
 - Clock gating to individual peripherals for power savings
 - IEEE 1149.1-1990 compliant Test Access Port (TAP) controller
 - Debug access via JTAG and Serial Wire interfaces
 - Full JTAG boundary scan
- Industrial and extended temperature 100-pin RoHS-compliant LQFP package
- Industrial-range 108-ball RoHS-compliant BGA package

1.2 Target Applications

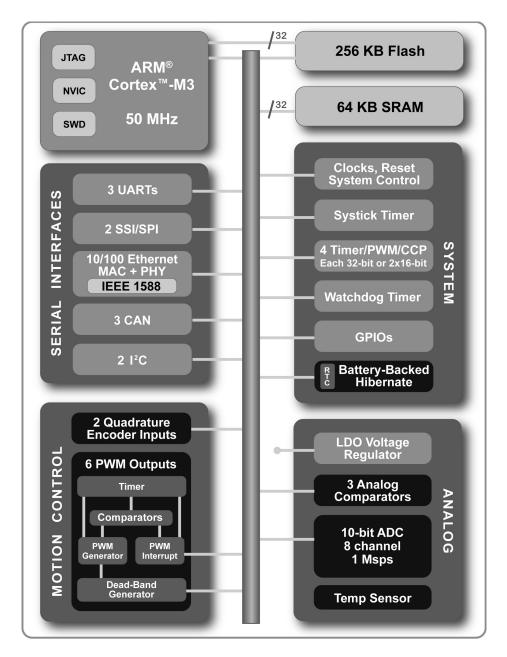
- Remote monitoring
- Electronic point-of-sale (POS) machines
- Test and measurement equipment
- Network appliances and switches
- Factory automation
- HVAC and building control
- Gaming equipment
- Motion control
- Medical instrumentation
- Fire and security

- Power and energy
- Transportation

1.3 High-Level Block Diagram

Figure 1-1 on page 31 represents the full set of features in the Stellaris[®] 8000 series of devices; not all features may be available on the LM3S8933 microcontroller.

Figure 1-1. Stellaris® 8000 Series High-Level Block Diagram



1.4 Functional Overview

The following sections provide an overview of the features of the LM3S8933 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 587.

1.4.1 ARM Cortex™-M3

1.4.1.1 Processor Core (see page 39)

All members of the Stellaris[®] product family, including the LM3S8933 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.

"ARM Cortex-M3 Processor Core" on page 39 provides an overview of the ARM core; the core is detailed in the ARM® Cortex™-M3 Technical Reference Manual.

1.4.1.2 System Timer (SysTick) (see page 42)

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.

1.4.1.3 Nested Vectored Interrupt Controller (NVIC) (see page 47)

The LM3S8933 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 32 interrupts.

"Interrupts" on page 47 provides an overview of the NVIC controller and the interrupt map. Exceptions and interrupts are detailed in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*.

1.4.2 Motor Control Peripherals

To enhance motor control, the LM3S8933 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 LM3S8933, PWM motion control functionality can be achieved through:

The motion control features of the general-purpose timers using the CCP pins

CCP Pins (see page 217)

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.

1.4.3 Analog Peripherals

To handle analog signals, the LM3S8933 microcontroller offers an Analog-to-Digital Converter (ADC).

For support of analog signals, the LM3S8933 microcontroller offers three analog comparators.

1.4.3.1 ADC (see page 270)

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

The LM3S8933 ADC module features 10-bit conversion resolution and supports four 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 502)

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

The LM3S8933 microcontroller provides three 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 LM3S8933 controller supports both asynchronous and synchronous serial communications with:

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

1.4.4.1 **UART** (see page 303)

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 LM3S8933 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 16x12 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 344)

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

The LM3S8933 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 381)

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 LM3S8933 controller includes one I²C module that provides the ability to communicate to other IC devices over an I²C bus. The I²C 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 Controller Area Network (see page 416)

Controller Area Network (CAN) is a multicast shared serial-bus standard for connecting electronic control units (ECUs). CAN was specifically designed to be robust in electromagnetically noisy environments and can utilize a differential balanced line like RS-485 or a more robust twisted-pair wire. Originally created for automotive purposes, now it is used in many embedded control applications (for example, industrial or medical). Bit rates up to 1Mb/s are possible at network lengths below 40 meters. Decreased bit rates allow longer network distances (for example, 125 Kb/s at 500m).

A transmitter sends a message to all CAN nodes (broadcasting). Each node decides on the basis of the identifier received whether it should process the message. The identifier also determines the priority that the message enjoys in competition for bus access. Each CAN message can transmit from 0 to 8 bytes of user information. The LM3S8933 includes one CAN units.

1.4.4.5 Ethernet Controller (see page 457)

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 169)

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 6-36 programmable input/output pins. The number of GPIOs available depends on the peripherals being used (see "Signal Tables" on page 517 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 Four Programmable Timers (see page 211)

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 four 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 247)

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 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 LM3S8933 controller offers both single-cycle SRAM and single-cycle Flash memory.

1.4.6.1 SRAM (see page 145)

The LM3S8933 static random access memory (SRAM) controller supports 64 KB SRAM. The internal SRAM of the Stellaris[®] devices is located at offset 0x0000.0000 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 146)

The LM3S8933 Flash controller supports 256 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 Memory Map (see page 45)

A memory map lists the location of instructions and data in memory. The memory map for the LM3S8933 controller can be found in "Memory Map" on page 45. Register addresses are given as a hexadecimal increment, relative to the module's base address as shown in the memory map.

The ARM® Cortex™-M3 Technical Reference Manual provides further information on the memory map.

1.4.7.2 JTAG TAP Controller (see page 50)

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

1.4.7.3 System Control and Clocks (see page 61)

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.

1.4.7.4 Hibernation Module (see page 125)

The Hibernation module provides logic to switch power off to the main processor and peripherals, and to wake on external or time-based events. The Hibernation module includes power-sequencing logic, a real-time clock with a pair of match registers, low-battery detection circuitry, and interrupt signalling to the processor. It also includes 64 32-bit words of non-volatile memory that can be used for saving state during hibernation.

1.4.8 Hardware Details

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

- "Pin Diagram" on page 515
- "Signal Tables" on page 517
- "Operating Characteristics" on page 543
- "Electrical Characteristics" on page 544

"Package Information" on page 559

2 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. Features include:

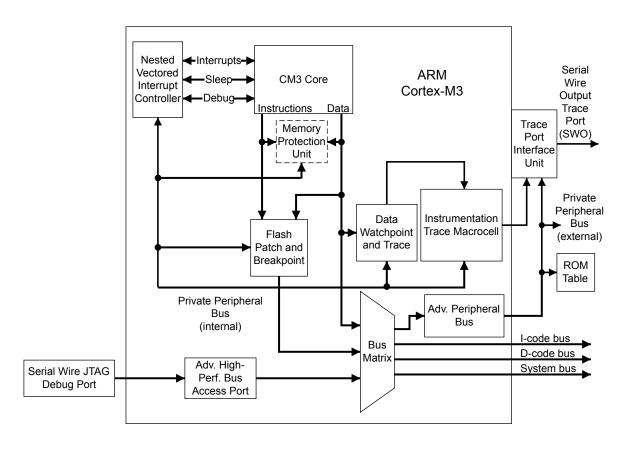
- Compact core.
- Thumb-2 instruction set, delivering the high-performance expected of an ARM core in the memory size usually associated with 8- and 16-bit devices; typically in the range of a few kilobytes of memory for microcontroller class applications.
- Rapid application execution through Harvard 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 ARM7TM processor family for better performance and power efficiency.
- Full-featured debug solution with a:
 - 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 motors.

For more information on the ARM Cortex-M3 processor core, see the *ARM*® *Cortex*™-*M3 Technical Reference Manual*. For information on SWJ-DP, see the *ARM*® *CoreSight Technical Reference Manual*.

2.1 Block Diagram

Figure 2-1. CPU Block Diagram



2.2 Functional Description

Important: The ARM® Cortex™-M3 Technical Reference Manual describes all the features of an ARM Cortex-M3 in detail. However, these features differ based on the implementation. This section describes the Stellaris® implementation.

Luminary Micro has implemented the ARM Cortex-M3 core as shown in Figure 2-1 on page 40. As noted in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*, several Cortex-M3 components are flexible in their implementation: SW/JTAG-DP, ETM, TPIU, the ROM table, the MPU, and the Nested Vectored Interrupt Controller (NVIC). Each of these is addressed in the sections that follow.

2.2.1 Serial Wire and JTAG Debug

Luminary Micro has replaced the ARM SW-DP and JTAG-DP with the ARM CoreSight[™]-compliant Serial Wire JTAG Debug Port (SWJ-DP) interface. This means Chapter 12, "Debug Port," of the *ARM® Cortex*[™]-*M3 Technical Reference Manual* does not apply to Stellaris[®] devices.

The SWJ-DP interface combines the SWD and JTAG debug ports into one module. See the CoreSight™ Design Kit Technical Reference Manual for details on SWJ-DP.

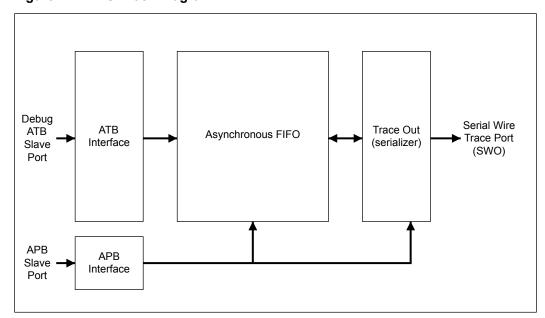
2.2.2 Embedded Trace Macrocell (ETM)

ETM was not implemented in the Stellaris[®] devices. This means Chapters 15 and 16 of the *ARM*® *Cortex*™-*M3 Technical Reference Manual* can be ignored.

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. The Stellaris[®] devices have implemented TPIU as shown in Figure 2-2 on page 41. This is similar to the non-ETM version described in the *ARM® Cortex™-M3 Technical Reference Manual*, however, SWJ-DP only provides SWV output for the TPIU.

Figure 2-2. TPIU Block Diagram



2.2.4 ROM Table

The default ROM table was implemented as described in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*.

2.2.5 Memory Protection Unit (MPU)

The Memory Protection Unit (MPU) is included on the LM3S8933 controller and 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.

2.2.6 Nested Vectored Interrupt Controller (NVIC)

The Nested Vectored Interrupt Controller (NVIC):

Facilitates low-latency exception and interrupt handling

- Controls power management
- Implements system control registers

The NVIC supports up to 240 dynamically reprioritizable interrupts each with up to 256 levels of priority. The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains knowledge of the stacked (nested) interrupts to enable tail-chaining of interrupts.

You can only fully access the NVIC from privileged mode, but you can pend interrupts in user-mode if you enable the Configuration Control Register (see the ARM® Cortex™-M3 Technical Reference Manual). Any other user-mode access causes a bus fault.

All NVIC registers are accessible using byte, halfword, and word unless otherwise stated.

2.2.6.1 Interrupts

The ARM® Cortex[™]-M3 Technical Reference Manual describes the maximum number of interrupts and interrupt priorities. The LM3S8933 microcontroller supports 32 interrupts with eight priority levels.

2.2.6.2 System Timer (SysTick)

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.

Functional Description

The timer consists of three registers:

- A control and status counter to configure its clock, enable the counter, enable the SysTick interrupt, and determine counter status.
- The reload value for the counter, used to provide the counter's wrap value.
- The current value of the counter.

A fourth register, the SysTick Calibration Value Register, is not implemented in the Stellaris[®] devices.

When enabled, the timer counts down from the reload value to zero, reloads (wraps) to the value in the SysTick Reload Value register on the next clock edge, then decrements on subsequent clocks. Writing a value of zero to the Reload Value register disables the counter on the next wrap. When the counter reaches zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

Writing to the Current Value register clears the register and the COUNTFLAG 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.

If the core is in debug state (halted), the counter will not decrement. The timer is clocked with respect to a reference clock. The reference clock can be the core clock or an external clock source.

SysTick Control and Status Register

Use the SysTick Control and Status Register to enable the SysTick features. The reset is 0x0000.0000.

Bit/Field	Name	Туре	Reset	Description
31:17	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	COUNTFLAG	R/W	0	Count Flag
				Returns 1 if timer counted to 0 since last time this was read. Clears on read by application. If read by the debugger using the DAP, this bit is cleared on read-only if the MasterType bit in the AHB-AP Control Register is set to 0. Otherwise, the COUNTFLAG bit is not changed by the debugger read.
15:3	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	CLKSOURCE	R/W	0	Clock Source
				Value Description
				0 External reference clock. (Not implemented for Stellaris microcontrollers.)
				1 Core clock
				If no reference clock is provided, it is held at 1 and so gives the same time as the core clock. The core clock must be at least 2.5 times faster than the reference clock. If it is not, the count values are unpredictable.
1	TICKINT	R/W	0	Tick Interrupt
				Value Description
				O Counting down to 0 does not generate the interrupt request to the NVIC. Software can use the COUNTFLAG to determine if ever counted to 0.
				1 Counting down to 0 pends the SysTick handler.
0	ENABLE	R/W	0	Enable
				Value Description
				0 Counter disabled.
				Counter operates in a multi-shot way. That is, counter loads with the Reload value and then begins counting down. On reaching 0, it sets the COUNTFLAG to 1 and optionally pends the SysTick handler, based on TICKINT. It then loads the Reload value again, and begins counting.

SysTick Reload Value Register

Use the SysTick Reload Value Register to specify the start value to load into the current value register when the counter reaches 0. It can be any value between 1 and 0x00FF.FFFF. A start value

of 0 is possible, but has no effect because the SysTick interrupt and COUNTFLAG are activated when counting from 1 to 0.

Therefore, as a multi-shot timer, repeated over and over, it fires every N+1 clock pulse, where N is any value from 1 to 0x00FF.FFFF. So, if the tick interrupt is required every 100 clock pulses, 99 must be written into the RELOAD. If a new value is written on each tick interrupt, so treated as single shot, then the actual count down must be written. For example, if a tick is next required after 400 clock pulses, 400 must be written into the RELOAD.

Bit/Field	Name	Туре	Reset	Description
31:24	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.
23:0	RELOAD	W1C	-	Reload Value to load into the SysTick Current Value Register when the counter reaches 0.

SysTick Current Value Register

Use the SysTick Current Value Register to find the current value in the register.

Bit/Field	Name	Туре	Reset	Description
31:24	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.
23:0	CURRENT	W1C	-	Current Value
				Current value at the time the register is accessed. 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 to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

SysTick Calibration Value Register

The SysTick Calibration Value register is not implemented.

3 Memory Map

The memory map for the LM3S8933 controller is provided in Table 3-1 on page 45.

In this manual, register addresses are given as a hexadecimal increment, relative to the module's base address as shown in the memory map. See also Chapter 4, "Memory Map" in the ARM® CortexTM-M3 Technical Reference Manual.

Table 3-1. Memory Map^a

Start	End	Description	For details on registers, see page
Memory			
0x0000.0000	0x0003.FFFF	On-chip flash ^b	149
0x0004.0000	0x1FFF.FFFF	Reserved	-
0x2000.0000	0x2000.FFFF	Bit-banded on-chip SRAM ^c	149
0x2001.0000	0x21FF.FFFF	Reserved	-
0x2200.0000	0x221F.FFFF	Bit-band alias of 0x2000.0000 through 0x200F.FFFF	145
0x2220.0000	0x3FFF.FFFF	Reserved	-
FiRM Peripherals	<u>'</u>		'
0x4000.0000	0x4000.0FFF	Watchdog timer	249
0x4000.1000	0x4000.3FFF	Reserved	-
0x4000.4000	0x4000.4FFF	GPIO Port A	176
0x4000.5000	0x4000.5FFF	GPIO Port B	176
0x4000.6000	0x4000.6FFF	GPIO Port C	176
0x4000.7000	0x4000.7FFF	GPIO Port D	176
0x4000.8000	0x4000.8FFF	SSI0	355
0x4000.9000	0x4000.BFFF	Reserved	-
0x4000.C000	0x4000.CFFF	UART0	310
0x4000.D000	0x4000.DFFF	UART1	310
0x4000.E000	0x4001.FFFF	Reserved	-
Peripherals	<u>.</u>		
0x4002.0000	0x4002.07FF	I2C Master 0	394
0x4002.0800	0x4002.0FFF	I2C Slave 0	407
0x4002.1000	0x4002.3FFF	Reserved	-
0x4002.4000	0x4002.4FFF	GPIO Port E	176
0x4002.5000	0x4002.5FFF	GPIO Port F	176
0x4002.6000	0x4002.6FFF	GPIO Port G	176
0x4002.7000	0x4002.FFFF	Reserved	-
0x4003.0000	0x4003.0FFF	Timer0	222
0x4003.1000	0x4003.1FFF	Timer1	222
0x4003.2000	0x4003.2FFF	Timer2	222
0x4003.3000	0x4003.3FFF	Timer3	222
0x4003.4000	0x4003.7FFF	Reserved	-
0x4003.8000	0x4003.8FFF	ADC	277

Start	End	Description	For details on registers, see page
0x4003.9000	0x4003.BFFF	Reserved	-
0x4003.C000	0x4003.CFFF	Analog Comparators	502
0x4003.D000	0x4003.FFFF	Reserved	-
0x4004.0000	0x4004.0FFF	CAN0 Controller	428
0x4004.1000	0x4004.7FFF	Reserved	-
0x4004.8000	0x4004.8FFF	Ethernet Controller	465
0x4004.9000	0x400F.BFFF	Reserved	-
0x400F.C000	0x400F.CFFF	Hibernation Module	132
0x400F.D000	0x400F.DFFF	Flash control	149
0x400F.E000	0x400F.EFFF	System control	71
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 B	us	'	
0xE000.0000	0xE000.0FFF	Instrumentation Trace Macrocell (ITM)	ARM® Cortex™-M3 Technical Reference Manual
0xE000.1000	0xE000.1FFF	Data Watchpoint and Trace (DWT)	ARM® Cortex™-M3 Technical Reference Manual
0xE000.2000	0xE000.2FFF	Flash Patch and Breakpoint (FPB)	ARM® Cortex™-M3 Technical Reference Manual
0xE000.3000	0xE000.DFFF	Reserved	-
0xE000.E000	0xE000.EFFF	Nested Vectored Interrupt Controller (NVIC)	ARM® Cortex™-M3 Technical Reference Manual
0xE000.F000	0xE003.FFFF	Reserved	-
0xE004.0000	0xE004.0FFF	Trace Port Interface Unit (TPIU)	ARM® Cortex™-M3 Technical Reference Manual
0xE004.1000	0xFFFF.FFFF	Reserved	-
L		1	

- a. All reserved space returns a bus fault when read or written.
- b. The unavailable flash will bus fault throughout this range.
- c. The unavailable SRAM will bus fault throughout this range.

4 Interrupts

The ARM Cortex-M3 processor and the Nested Vectored Interrupt Controller (NVIC) 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.

Table 4-1 on page 47 lists all exception types. Software can set eight priority levels on seven of these exceptions (system handlers) as well as on 32 interrupts (listed in Table 4-2 on page 48).

Priorities on the system handlers are set with the NVIC System Handler Priority registers. Interrupts are enabled through the NVIC Interrupt Set Enable register and prioritized with the NVIC Interrupt Priority registers. You also can group priorities by splitting priority levels into pre-emption priorities and subpriorities. All of the interrupt registers are described in Chapter 8, "Nested Vectored Interrupt Controller" in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*.

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

If you assign the same priority level to two or more interrupts, their hardware priority (the lower position number) determines the order in which the processor activates them. For example, if both GPIO Port A and GPIO Port B are priority level 1, then GPIO Port A has higher priority.

Important: It may take several processor cycles after a write to clear an interrupt source in order for NVIC to see the interrupt source de-assert. This means if the interrupt clear is done as the last action in an interrupt handler, it is possible for the interrupt handler to complete while NVIC sees the interrupt as still asserted, causing the interrupt handler to be re-entered errantly. This 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 Chapter 5, "Exceptions" and Chapter 8, "Nested Vectored Interrupt Controller" in the *ARM*® *Cortex™-M3 Technical Reference Manual* for more information on exceptions and interrupts.

Table 4-1. Exception Types

Exception Type	Vector Number	Priority ^a	Description
-	0	-	Stack top is loaded from first entry of vector table on reset.
Reset	1	-3 (highest)	Invoked on power up and warm reset. On first instruction, drops to lowest priority (and then is called the base level of activation). This is asynchronous.
Non-Maskable Interrupt (NMI)	2	-2	Cannot be stopped or preempted by any exception but reset. This is asynchronous. An NMI is only producible by software, using the NVIC Interrupt Control State register.
Hard Fault	3	-1	All classes of Fault, when the fault cannot activate due to priority or the configurable fault handler has been disabled. This is synchronous.
Memory Management	4	settable	MPU mismatch, including access violation and no match. This is synchronous. The priority of this exception can be changed.

Exception Type	Vector Number	Priority ^a	Description
Bus Fault	5	settable	Pre-fetch fault, memory access fault, and other address/memory related faults. This is synchronous when precise and asynchronous when imprecise.
			You can enable or disable this fault.
Usage Fault	6	settable	Usage fault, such as undefined instruction executed or illegal state transition attempt. This is synchronous.
-	7-10	-	Reserved.
SVCall	11	settable	System service call with SVC instruction. This is synchronous.
Debug Monitor	12	settable	Debug monitor (when not halting). This is synchronous, but only active when enabled. It does not activate if lower priority than the current activation.
-	13	-	Reserved.
PendSV	14	settable	Pendable request for system service. This is asynchronous and only pended by software.
SysTick	15	settable	System tick timer has fired. This is asynchronous.
Interrupts	16 and above	settable	Asserted from outside the ARM Cortex-M3 core and fed through the NVIC (prioritized). These are all asynchronous. Table 4-2 on page 48 lists the interrupts on the LM3S8933 controller.

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

Table 4-2. Interrupts

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Description
0-15	-	Processor exceptions
16	0	GPIO Port A
17	1	GPIO Port B
18	2	GPIO Port C
19	3	GPIO Port D
20	4	GPIO Port E
21	5	UART0
22	6	UART1
23	7	SSI0
24	8	I2C0
25-29	9-13	Reserved
30	14	ADC Sequence 0
31	15	ADC Sequence 1
32	16	ADC Sequence 2
33	17	ADC Sequence 3
34	18	Watchdog timer
35	19	Timer0 A
36	20	Timer0 B
37	21	Timer1 A
38	22	Timer1 B
39	23	Timer2 A
40	24	Timer2 B

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Description
41	25	Analog Comparator 0
42	26	Analog Comparator 1
43	27	Analog Comparator 2
44	28	System Control
45	29	Flash Control
46	30	GPIO Port F
47	31	GPIO Port G
48-50	32-34	Reserved
51	35	Timer3 A
52	36	Timer3 B
53-54	37-38	Reserved
55	39	CAN0
56-57	40-41	Reserved
58	42	Ethernet Controller
59	43	Hibernation Module
60-63	44-47	Reserved

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

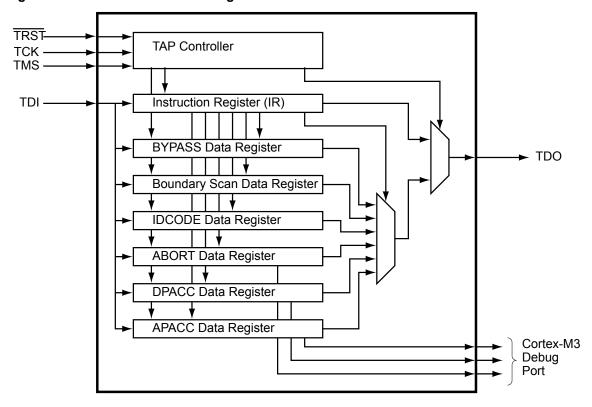
The JTAG module has the following features:

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

See the ARM® Cortex™-M3 Technical Reference Manual for more information on the ARM JTAG controller.

5.1 Block Diagram

Figure 5-1. JTAG Module Block Diagram



5.2 Functional Description

A high-level conceptual drawing of the JTAG module is shown in Figure 5-1 on page 51. 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 5-2 on page 57 for a list of implemented instructions).

See "JTAG and Boundary Scan" on page 555 for JTAG timing diagrams.

5.2.1 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 5-1 on page 52. Detailed information on each pin follows.

Table 5-1. 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

5.2.1.1 Test Reset Input (TRST)

The TRST pin is an asynchronous active Low input signal for initializing and resetting the JTAG TAP controller and associated JTAG circuitry. When TRST is asserted, the TAP controller resets to the Test-Logic-Reset state and remains there while 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.

5.2.1.2 Test Clock Input (TCK)

The ${ t TCK}$ 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, ${ t TCK}$ is driven by a free-running clock with a nominal 50% duty cycle. When necessary, ${ t TCK}$ can be stopped at 0 or 1 for extended periods of time. While ${ t TCK}$ 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.

5.2.1.3 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 5-2 on page 54.

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.

5.2.1.4 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.

5.2.1.5 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 TDO 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.

5.2.2 JTAG TAP Controller

The JTAG TAP controller state machine is shown in Figure 5-2 on page 54. 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*.

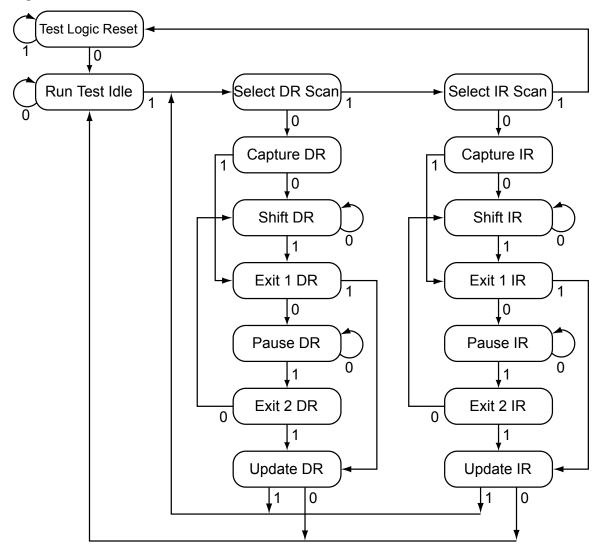


Figure 5-2. Test Access Port State Machine

5.2.3 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 57.

5.2.4 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.

5.2.4.1 GPIO Functionality

When the controller is reset with either a POR or \overline{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 commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 186) are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register (see page 196) has been unlocked and the appropriate bits of the **GPIO Commit (GPIOCR)** register (see page 197) have been set to 1.

Recovering a "Locked" Device

Note: Performing the below sequence will cause the nonvolatile registers discussed in "Nonvolatile Register Programming" on page 148 to be restored to their factory default values. The mass erase of the flash memory caused by the below sequence occurs prior to the nonvolatile registers being restored.

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 RST signal.
- Perform the JTAG-to-SWD switch sequence.
- 3. Perform the SWD-to-JTAG switch sequence.
- Perform the JTAG-to-SWD switch sequence.
- Perform the SWD-to-JTAG switch sequence.
- Perform the JTAG-to-SWD switch sequence.
- Perform the SWD-to-JTAG switch sequence.
- Perform the JTAG-to-SWD switch sequence.
- Perform the SWD-to-JTAG switch sequence.
- 10. Perform the JTAG-to-SWD switch sequence.
- 11. Perform the SWD-to-JTAG switch sequence.

- 12. Release the RST signal.
- 13. Wait 400 ms.
- 14. Power-cycle the device.

The JTAG-to-SWD and SWD-to-JTAG switch sequences are described in "ARM Serial Wire Debug (SWD)" on page 56. When performing switch sequences for the purpose of recovering the debug capabilities of the device, only steps 1 and 2 of the switch sequence need to be performed.

5.2.4.2 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 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, 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*® *Cortex™-M3 Technical Reference Manual* and the *ARM*® *CoreSight Technical Reference Manual*.

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 a switch sequence to the device. The 16-bit switch sequence for switching to SWD mode is defined as b1110011110011110, transmitted LSB first. This can also be represented as 16'hE79E 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 set to 1. This ensures that both JTAG and SWD are in their reset/idle states.
- 2. Send the 16-bit JTAG-to-SWD switch sequence, 16'hE79E.
- Send at least 50 TCK/SWCLK cycles with TMS/SWDIO set to 1. This ensures that if SWJ-DP was already in SWD mode, before sending the switch sequence, the SWD goes into the line reset state.

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 sequence to the device. The 16-bit switch sequence for switching to JTAG mode is defined as b1110011100111100, transmitted LSB first. This can also be represented as 16'hE73C 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 set to 1. This ensures that both JTAG and SWD are in their reset/idle states.
- 2. Send the 16-bit SWD-to-JTAG switch sequence, 16'hE73C.
- Send at least 5 TCK/SWCLK cycles with TMS/SWDIO set to 1. This ensures that if SWJ-DP was already in JTAG mode, before sending the switch sequence, the JTAG goes into the Test Logic Reset state.

5.3 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.

5.4 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.

5.4.1 Instruction Register (IR)

The JTAG TAP Instruction Register (IR) is a four-bit serial scan chain with a parallel load register connected between the JTAG TDI and TDO pins. 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 5-2 on page 57. A detailed explanation of each instruction, along with its associated Data Register, follows.

Table 5-2. 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 TDI is always connected to TDO.

5.4.1.1 EXTEST Instruction

The EXTEST instruction does not have an associated 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.

5.4.1.2 INTEST Instruction

The INTEST instruction does not have an associated 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.

5.4.1.3 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 60 for more information.

5.4.1.4 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 60 for more information.

5.4.1.5 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 60 for more information.

5.4.1.6 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 60 for more information.

5.4.1.7 IDCODE Instruction

The IDCODE instruction connects the associated IDCODE Data Register chain between \mathtt{TDI} and \mathtt{TDO} . 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, $\overline{\mathtt{TRST}}$ is asserted, or the Test-Logic-Reset state is entered. Please see "IDCODE Data Register" on page 59 for more information.

5.4.1.8 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 59 for more information.

5.4.2 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.

5.4.2.1 IDCODE Data Register

The format for the 32-bit IDCODE Data Register defined by the *IEEE Standard 1149.1* is shown in Figure 5-3 on page 59. 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 0x3BA00477. This value indicates an ARM Cortex-M3, Version 1 processor. This allows the debuggers to automatically configure themselves to work correctly with the Cortex-M3 during debug.

Figure 5-3. IDCODE Register Format



5.4.2.2 BYPASS Data Register

The format for the 1-bit BYPASS Data Register defined by the *IEEE Standard 1149.1* is shown in Figure 5-4 on page 60. 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 5-4. BYPASS Register Format

$$-TDI \longrightarrow 0$$

5.4.2.3 Boundary Scan Data Register

The format of the Boundary Scan Data Register is shown in Figure 5-5 on page 60. Each GPIO pin, in a counter-clockwise direction from 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. In addition to the GPIO pins, the controller reset pin, \overline{RST} , is included in the chain. Because the reset pin is always an input, only the input signal is included in the Data Register chain.

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 <code>TCK</code> 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 5-5. Boundary Scan Register Format

For detailed information on the order of the input, output, and output enable bits for each of the GPIO ports, please refer to the Stellaris[®] Family Boundary Scan Description Language (BSDL) files, downloadable from www.luminarymicro.com.

5.4.2.4 APACC Data Register

The format for the 35-bit APACC Data Register defined by ARM is described in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*.

5.4.2.5 DPACC Data Register

The format for the 35-bit DPACC Data Register defined by ARM is described in the *ARM*® Cortex™-M3 Technical Reference Manual.

5.4.2.6 ABORT Data Register

The format for the 35-bit ABORT Data Register defined by ARM is described in the *ARM*® *Cortex*™-*M3 Technical Reference Manual*.

6 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.

6.1 Functional Description

The System Control module provides the following capabilities:

- Device identification, see "Device Identification" on page 61
- Local control, such as reset (see "Reset Control" on page 61), power (see "Power Control" on page 64) and clock control (see "Clock Control" on page 65)
- System control (Run, Sleep, and Deep-Sleep modes), see "System Control" on page 68

6.1.1 Device Identification

Seven 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.

6.1.2 Reset Control

This section discusses aspects of hardware functions during reset as well as system software requirements following the reset sequence.

6.1.2.1 CMOD0 and CMOD1 Test-Mode Control Pins

Two pins, CMOD0 and CMOD1, are defined for use by Luminary Micro for testing the devices during manufacture. They have no end-user function and should not be used. The CMOD pins should be connected to ground.

6.1.2.2 Reset Sources

The controller has five sources of reset:

- 1. External reset input pin (RST) assertion, see "RST Pin Assertion" on page 61.
- 2. Power-on reset (POR), see "Power-On Reset (POR)" on page 62.
- 3. Internal brown-out (BOR) detector, see "Brown-Out Reset (BOR)" on page 62.
- 4. Software-initiated reset (with the software reset registers), see "Software Reset" on page 63.
- 5. A watchdog timer reset condition violation, see "Watchdog Timer Reset" on page 63.

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 is the cause, and then all the other bits in the **RESC** register are cleared except for the POR indicator.

6.1.2.3 RST Pin Assertion

The external reset pin (RST) resets the controller. This resets the core and all the peripherals except the JTAG TAP controller (see "JTAG Interface" on page 50). The external reset sequence is as follows:

- 1. The external reset pin (\overline{RST}) is asserted and then de-asserted.
- 2. The internal reset is released and the core loads from memory the initial stack pointer, the initial program counter, the first instruction designated by the program counter, and begins execution. A few clocks cycles from RST de-assertion to the start of the reset sequence is necessary for synchronization.

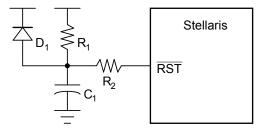
The external reset timing is shown in Figure 22-11 on page 557.

6.1.2.4 Power-On Reset (POR)

The Power-On Reset (POR) circuit monitors the power supply voltage (V_{DD}). The POR circuit generates a reset signal to the internal logic when the power supply ramp reaches a threshold value (V_{TH}). If the application only uses the POR circuit, the $\overline{\tt RST}$ input needs to be connected to the power supply (V_{DD}) through a pull-up resistor (1K to 10K Ω).

The device must be operating within the specified operating parameters at the point when the on-chip power-on reset pulse is complete. The 3.3-V power supply to the device must reach 3.0 V within 10 msec of it crossing 2.0 V to guarantee proper operation. For applications that require the use of an external reset to hold the device in reset longer than the internal POR, the RST input may be used with the circuit as shown in Figure 6-1 on page 62.

Figure 6-1. External Circuitry to Extend Reset



The R_1 and C_1 components define the power-on delay. The R_2 resistor mitigates any leakage from the \overline{RST} input. The diode (D₁) discharges C_1 rapidly when the power supply is turned off.

The Power-On Reset sequence is as follows:

- 1. The controller waits for the later of external reset (RST) or internal POR to go inactive.
- 2. The internal reset is released and the core loads from memory the initial stack pointer, the initial program counter, the first instruction designated by the program counter, and begins execution.

The internal POR is only active on the initial power-up of the controller. The Power-On Reset timing is shown in Figure 22-12 on page 558.

Note: The power-on reset also resets the JTAG controller. An external reset does not.

6.1.2.5 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 equivelent 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 22-13 on page 558.

6.1.2.6 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 68). 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 22-14 on page 558.

6.1.2.7 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 22-15 on page 558.

6.1.3 Power Control

The Stellaris microcontroller provides an integrated LDO regulator that may be used to provide power to the majority of the controller's internal logic. 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 6-2 on page 65 shows the power architecture.

Note: On the printed circuit board, use the LDO output as the source of VDD25 input. In addition, the LDO requires decoupling capacitors. See "On-Chip Low Drop-Out (LDO) Regulator Characteristics" on page 545.

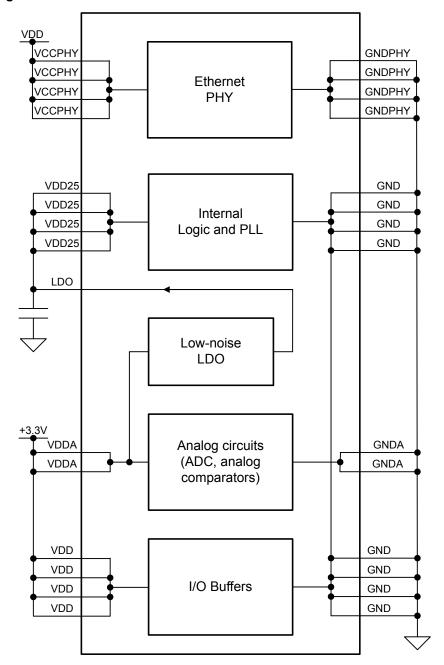


Figure 6-2. Power Architecture

6.1.4 Clock Control

System control determines the control of clocks in this part.

6.1.4.1 Fundamental Clock Sources

There are four 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.

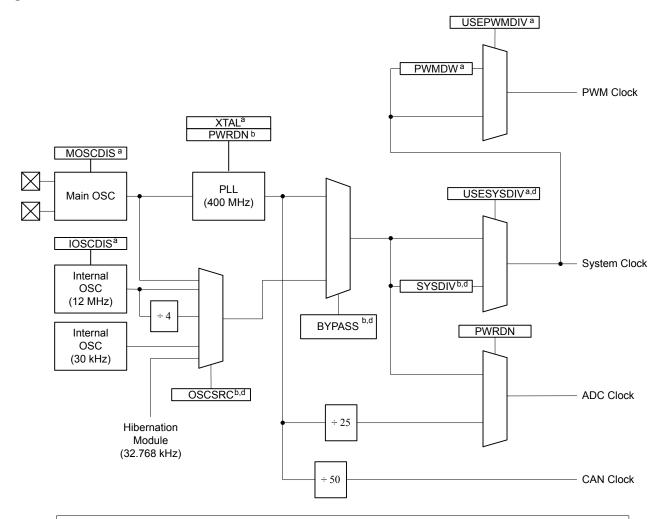
- Main Oscillator (MOSC): The main oscillator provides a frequency-accurate clock source by one of two means: an external single-ended clock source is connected to the OSC0 input pin, or an external crystal is connected across the OSC0 input and OSC1 output pins. If the PLL is being used, the crystal 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 80).
- 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.
- External Real-Time Oscillator: The external real-time oscillator provides a low-frequency, accurate clock reference. It is intended to provide the system with a real-time clock source. The real-time oscillator is part of the Hibernation Module ("Hibernation Module" on page 125) and may also provide an accurate source of Deep-Sleep or Hibernate mode power savings.

The internal system clock (SysClk), is derived from any of the four sources plus two others: the output of the main internal PLL, and the internal oscillator divided by four (3 MHz ± 30%). The frequency of the PLL clock reference must be in the range of 3.579545 MHz to 8.192 MHz (inclusive).

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.

Figure 6-3 on page 67 shows the logic for the main clock tree. The peripheral blocks are driven by the system clock signal and can be programmatically enabled/disabled. The ADC clock signal is automatically divided down to 16 MHz for proper ADC operation.

Figure 6-3. 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.

6.1.4.2 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 \mathtt{XTAL} bit in the **RCC** register (see page 80) 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.

6.1.4.3 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.

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 84). The internal translation provides a translation within \pm 1% of the targeted PLL VCO frequency.

The Crystal Value field (XTAL) on page 80 describes the available crystal choices and default programming of the **PLLCFG** register. The crystal number is written into the XTAL field of the **Run-Mode Clock Configuration (RCC)** register. Any time the XTAL field changes, the new settings are translated and the internal PLL settings are updated.

6.1.4.4 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 80 and page 85).

6.1.4.5 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 22-7 on page 547). During the relock time, the affected PLL is not usable as a clock reference.

The 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.

6.1.5 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.

In Run mode, the processor executes code. In Sleep mode, the clock frequency of the active peripherals is unchanged, but the processor is not clocked and therefore no longer executes code. 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. Each mode is described in more detail below.

There are four levels of operation for the device defined as:

- Run Mode. 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. 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 the system control NVIC section of the ARM® Cortex™-M3 Technical Reference Manual for more details.
 - In Sleep mode, the Cortex-M3 processor core and the memory subsystem are not clocked. 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. 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 the system control NVIC section of the ARM® Cortex™-M3 Technical Reference Manual 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 /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.
- Hibernate Mode. In this mode, the power supplies are turned off to the main part of the device and only the Hibernation module's circuitry is active. An external wake event or RTC event is required to bring the device back to Run mode. The Cortex-M3 processor and peripherals outside of the Hibernation module see a normal "power on" sequence and the processor starts running code. It can determine that it has been restarted from Hibernate mode by inspecting the Hibernation module registers.

6.2 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:

- 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 (using the
 main oscillator or internal oscillator) 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.
- 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.

6.3 Register Map

Table 6-1 on page 70 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 by Luminary Micro, Inc. Software should not modify any reserved memory address.

Table 6-1. System Control Register Map

Offset	Name	Type	Reset	Description	See page
0x000	DID0	RO	-	Device Identification 0	72
0x004	DID1	RO	-	Device Identification 1	88
0x008	DC0	RO	0x00FF.007F	Device Capabilities 0	90
0x010	DC1	RO	0x0101.33FF	Device Capabilities 1	91
0x014	DC2	RO	0x070F.1013	Device Capabilities 2	93
0x018	DC3	RO	0x8F0F.3FC0	Device Capabilities 3	95
0x01C	DC4	RO	0x5100.007F	Device Capabilities 4	97
0x030	PBORCTL	R/W	0x0000.7FFD	Brown-Out Reset Control	74
0x034	LDOPCTL	R/W	0x0000.0000	LDO Power Control	75
0x040	SRCR0	R/W	0x00000000	Software Reset Control 0	120
0x044	SRCR1	R/W	0x00000000	Software Reset Control 1	121
0x048	SRCR2	R/W	0x00000000	Software Reset Control 2	123
0x050	RIS	RO	0x0000.0000	Raw Interrupt Status	76
0x054	IMC	R/W	0x0000.0000	Interrupt Mask Control	77
0x058	MISC	R/W1C	0x0000.0000	Masked Interrupt Status and Clear	78
0x05C	RESC	R/W	-	Reset Cause	79

Offset	Name	Туре	Reset	Description	See page
0x060	RCC	R/W	0x0780.3AD1	Run-Mode Clock Configuration	80
0x064	PLLCFG	RO	-	XTAL to PLL Translation	84
0x070	RCC2	R/W	0x0780.2810	Run-Mode Clock Configuration 2	85
0x100	RCGC0	R/W	0x00000040	Run Mode Clock Gating Control Register 0	99
0x104	RCGC1	R/W	0x00000000	Run Mode Clock Gating Control Register 1	105
0x108	RCGC2	R/W	0x00000000	Run Mode Clock Gating Control Register 2	114
0x110	SCGC0	R/W	0x00000040	Sleep Mode Clock Gating Control Register 0	101
0x114	SCGC1	R/W	0x00000000	Sleep Mode Clock Gating Control Register 1	108
0x118	SCGC2	R/W	0x00000000	Sleep Mode Clock Gating Control Register 2	116
0x120	DCGC0	R/W	0x00000040	Deep Sleep Mode Clock Gating Control Register 0	103
0x124	DCGC1	R/W	0x00000000	Deep Sleep Mode Clock Gating Control Register 1	111
0x128	DCGC2	R/W	0x00000000	Deep Sleep Mode Clock Gating Control Register 2	118
0x144	DSLPCLKCFG	R/W	0x0780.0000	Deep Sleep Clock Configuration	87

6.4 Register Descriptions

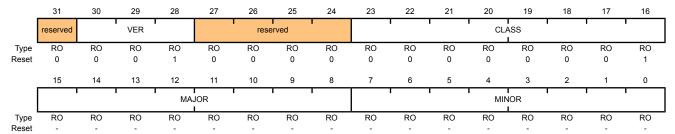
All addresses given are relative to the System Control base address of 0x400F.E000.

Register 1: Device Identification 0 (DID0), offset 0x000

This register identifies the version of the device.

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 $\textbf{DID0}$ register format version. The version number is numeric. The value of the \mathtt{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.

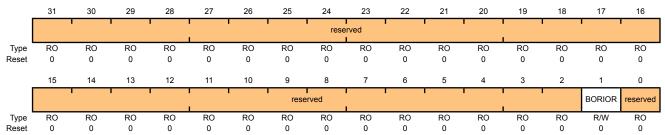
Register 2: Brown-Out Reset Control (PBORCTL), offset 0x030

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	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	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}).

Reset

LDO Power Control (LDOPCTL)

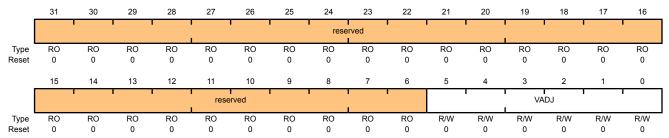
Name

Type

Base 0x400F.E000 Offset 0x034

Bit/Field

Type R/W, reset 0x0000.0000



		,,		'
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

Description

This field sets the on-chip output voltage. The programming values for the \mathtt{VADJ} field are provided below.

Value	$V_{OUT}\left(V\right)$
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

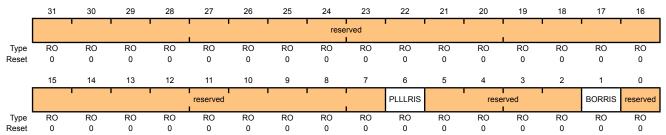
Register 4: Raw Interrupt Status (RIS), offset 0x050

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.

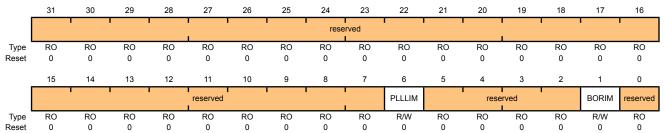
Register 5: Interrupt Mask Control (IMC), offset 0x054

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 current limit detection 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.

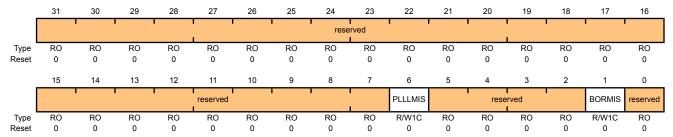
Register 6: Masked Interrupt Status and Clear (MISC), offset 0x058

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 76).

Masked Interrupt Status and Clear (MISC)

Base 0x400F.E000

Offset 0x058
Type R/W1C, reset 0x0000.0000



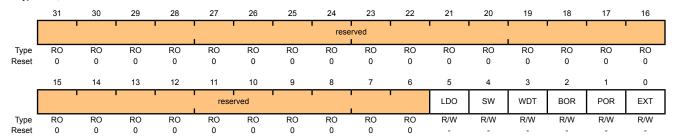
Bit/Field	Name	Туре	Reset	Description
31:7	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	PLLLMIS	R/W1C	0	PLL Lock Masked Interrupt Status
				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 ${\tt BORMIS}$ is simply the ${\tt BORRIS}$ ANDed with the mask value, ${\tt BORIM}.$
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 an external reset is the cause, and then all the other bits in the **RESC** register are cleared.

Reset Cause (RESC)

Base 0x400F.E000 Offset 0x05C Type R/W, reset -



Bit/Field	Name	Туре	Reset	Description
31:6	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	LDO	R/W	-	LDO Reset
				When set, indicates the LDO circuit has lost regulation and has generated a reset event.
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

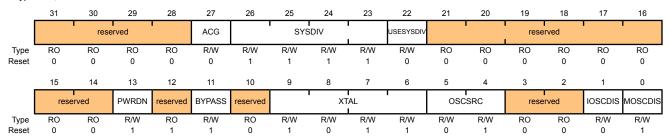
This register is defined to provide source control and frequency speed.

Run-Mode Clock Configuration (RCC)

Base 0x400F.E000 Offset 0x060

D:4/E: -1-4

Type R/W, reset 0x0780.3AD1



Bivrieid	ivame	туре	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

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

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.

Bit/Field	Name	Type	Reset	Description	
26:23	SYSDIV	R/W	0xF	System Clock Divisor	
				Specifies which divisor is us PLL output.	sed to generate the system clock from the
				The PLL VCO frequency is	400 MHz.
				Value Divisor (BYPASS=1) Frequency (RVPASS=0)
				0x0 reserved	reserved
				0x1 /2	reserved
				0x2 /3	reserved
				0x3 /4	50 MHz
				0x4 /5	40 MHz
				0x5 /6	33.33 MHz
				0x6 /7	28.57 MHz
				0x7 /8	25 MHz
				0x8 /9	22.22 MHz
				0x9 /10	20 MHz
				0xA /11	18.18 MHz
				0xB /12	16.67 MHz
				0xC /13	15.38 MHz
				0xD /14	14.29 MHz
				0xE /15	13.33 MHz
				0xF /16	12.5 MHz (default)
				page 80), the SYSDIV value	le Clock Configuration (RCC) register (see e is MINSYSDIV if a lower divider was eing used. This lower value is allowed to
22	USESYSDIV	R/W	0	Enable System Clock Divid	er
				-	er as the source for the system clock. The ed to be used when the PLL is selected as
21:14	reserved	RO	0		the value of a reserved bit. To provide ducts, the value of a reserved bit should be odify-write operation.
13	PWRDN	R/W	1	PLL Power Down	
				This bit connects to the PLL down the PLL.	PWRDN input. The reset value of 1 powers
12	reserved	RO	1		the value of a reserved bit. To provide ducts, the value of a reserved bit should be odify-write operation.

July 25, 2008 Preliminary

Bit/Field	Name	Туре	Reset	Description		
11	BYPASS	R/W	1	PLL Bypas	S	
				the OSC so source. Oth	hether the system clock is der ource. If set, the clock that driv nerwise, the clock that drives the ed by the system divider.	es the system is the OSC
				1 tl s	The ADC must be clocked from 4-MHz to 18-MHz clock source he ADC works in a 14-18 MHz cample/second rate, the ADC relock source.	e to operate properly. While range, to maintain a 1 M
10	reserved	RO	0	compatibilit	nould not rely on the value of a ry with future products, the valu across a read-modify-write ope	ue of a reserved bit should be
9:6	XTAL	R/W	0xB	Crystal Val	ue	
					pecifies the crystal value attach or this field is provided below.	ned to the main oscillator. The
				Value	Crystal Frequency (MHz) Not Using the PLL	Crystal Frequency (MHz) Using the PLL
				0x0	1.000	reserved
				0x1	1.8432	reserved
				0x2	2.000	reserved
				0x3	2.4576	reserved
				0x4	3.5795	545 MHz
				0x5	3.686	64 MHz
				0x6	4	MHz
				0x7	4.09	6 MHz
				0x8	4.915	52 MHz
				0x9	5	MHz
				0xA	5.12	2 MHz
				0xB	6 MHz (r	eset value)
				0xC	6.14	4 MHz
				0xD		28 MHz
				0xE		MHz
				0xF	8.19	2 MHz
5:4	OSCSRC	R/W	0x1	Oscillator S	Source	
				Picks amor	ng the four input sources for th	e OSC. The values are:
				Value Inpo		
					n oscillator	
					rnal oscillator (default)	
					rnal oscillator / 4 (this is neces	ssary if used as input to PLL)
				0x3 30 I	KHz internal oscillator	

Bit/Field	Name	Type	Reset	Description
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	IOSCDIS	R/W	0	Internal Oscillator Disable
				0: Internal oscillator (IOSC) is enabled.
				1: Internal oscillator is disabled.
0	MOSCDIS	R/W	1	Main Oscillator Disable
				0: Main oscillator is enabled .
				1: Main oscillator is disabled (default).

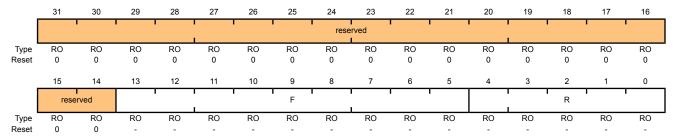
Register 9: XTAL to PLL Translation (PLLCFG), offset 0x064

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 80).

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

24

This register overrides the **RCC** equivalent register fields when the USERCC2 bit is set. This allows RCC2 to be used to extend the capabilities, while also providing a means to be backward-compatible to previous parts. The fields within the **RCC2** register occupy the same bit positions as they do within the **RCC** register as LSB-justified.

The SYSDIV2 field is wider so that additional larger divisors are possible. This allows a lower system clock frequency for improved Deep Sleep power consumption.

23

22

21

preserved across a read-modify-write operation.

preserved across a read-modify-write operation.

When set, bypasses the PLL for the clock source.

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

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

19

reserved

18

16

Run-Mode Clock Configuration 2 (RCC2)

reserved

28

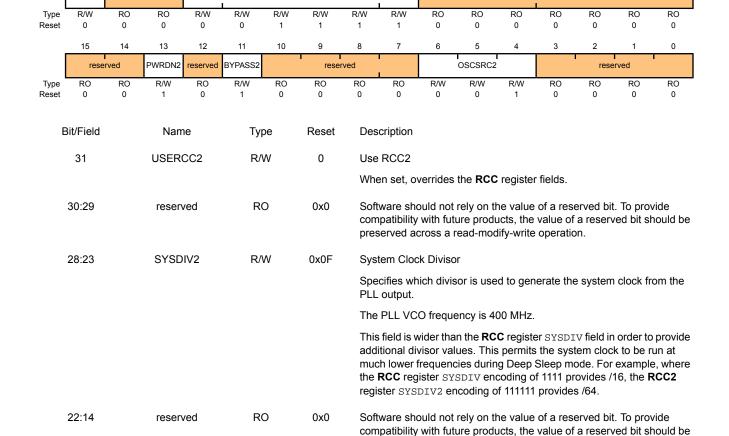
26

SYSDIV2

Base 0x400F.E000 Offset 0x070

JSERCO

Type R/W, reset 0x0780.2810



Power-Down PLL

Bypass PLL

When set, powers down the PLL.

13

12

11

PWRDN2

reserved

BYPASS2

R/W

RO

R/W

1

0

1

Bit/Field	Name	Туре	Reset	Description
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
				Picks among the input sources for the OSC. The values are:
				Value Description
				0x0 Main oscillator (MOSC)
				0x1 Internal oscillator (IOSC)
				0x2 Internal oscillator / 4
				0x3 30 kHz internal oscillator
				0x7 32 kHz external oscillator
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.

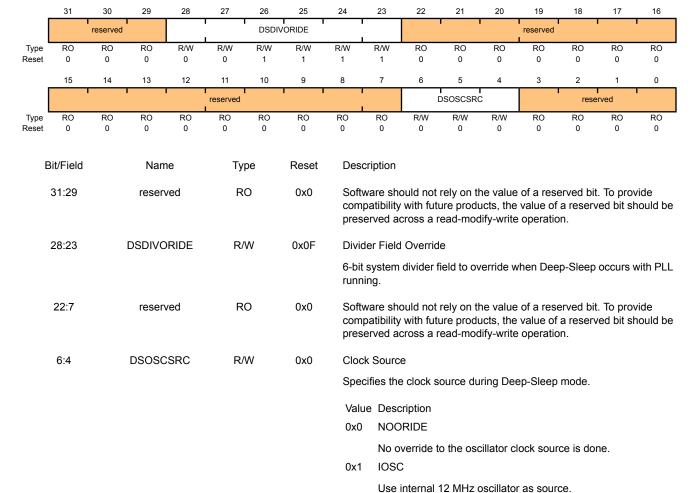
Register 11: Deep Sleep Clock Configuration (DSLPCLKCFG), offset 0x144

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



30kHz

32kHz

Use 30 kHz internal oscillator.

Use 32 kHz external oscillator.

0x3

0x7

RO

0x0

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.

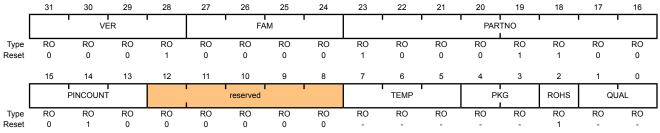
3.0

Register 12: Device Identification 1 (DID1), offset 0x004

This register identifies the device family, part number, temperature range, pin count, and package type.

Device Identification 1 (DID1)

Base 0x400F.E000 Offset 0x004 Type RO, reset -



			-	
Bit/Field	Name	Туре	Reset	Description
31:28	VER	RO	0x1	DID1 Version
				This field defines the DID1 register format version. The version number is numeric. The value of the VER field is encoded as follows (all other encodings are reserved):
				Value Description
				0x1 Second version of the DID1 register format.
27:24	FAM	RO	0x0	Family
				This field provides the family identification of the device within the Luminary Micro product portfolio. The value is encoded as follows (all other encodings are reserved):
				Value Description
				0x0 Stellaris family of microcontollers, that is, all devices with external part numbers starting with LM3S.
23:16	PARTNO	RO	0x8C	Part Number
				This field provides the part number of the device within the family. The value is encoded as follows (all other encodings are reserved):
				Value Description
				0x8C LM3S8933

PINCOUNT RO 15:13 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):

Value Description

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

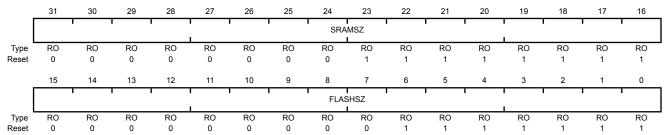
Register 13: Device Capabilities 0 (DC0), offset 0x008

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 0x00FF.007F



Bit/Field	Name	Type	Reset	Description
31:16	SRAMSZ	RO	0x00FF	SRAM Size Indicates the size of the on-chip SRAM memory. Value Description 0x00FF 64 KB of SRAM
15:0	FLASHSZ	RO	0x007F	Flash Size

Indicates the size of the on-chip flash memory.

Value Description 0x007F 256 KB of Flash

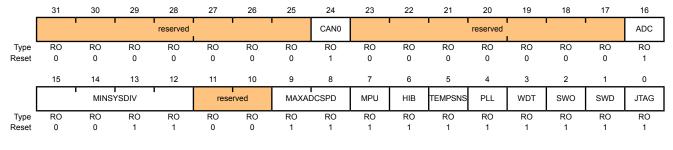
Register 14: Device Capabilities 1 (DC1), offset 0x010

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)

Base 0x400F.E000 Offset 0x010

Type RO, reset 0x0101.33FF



Bit/Field	Name	Туре	Reset	Description
31:25	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.
24	CAN0	RO	1	CAN Module 0 Present
				When set, indicates that CAN unit 0 is present.
23:17	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	RO	1	ADC Module Present
				When set, indicates that the ADC module is present.
15:12	MINSYSDIV	RO	0x3	System Clock Divider
				Minimum 4-bit divider value for system clock. The reset value is hardware-dependent. See the RCC register for how to change the system clock divisor using the SYSDIV bit.
				Value Description
				0x3 Specifies a 50-MHz CPU clock with a PLL divider of 4.
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.

Bit/Field	Name	Туре	Reset	Description
9:8	MAXADCSPD	RO	0x3	Max ADC Speed
				Indicates the maximum rate at which the ADC samples data.
				Value Description
				0x3 1M samples/second
7	MPU	RO	1	MPU Present
				When set, indicates that the Cortex-M3 Memory Protection Unit (MPU) module is present. See the ARM Cortex-M3 Technical Reference Manual for details on the MPU.
6	HIB	RO	1	Hibernation Module Present
				When set, indicates that the Hibernation module is present.
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

Type RO, reset 0x070F.1013

			reserved			COMP2	COMP1	COMP0		reser	ved		TIMER3	TIMER2	TIMER1	TIMER0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO 0	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	1	1	1	0	0	-	0	1	1	1	1
ı	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		reserved		I2C0				reserved				SSI0	rese	rved	UART1	UART0
Type Reset	RO 0	RO 0	RO 0	RO 1	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 0	RO 0	RO 1	RO 1
110001	Ü	Ü	Ü		Ü	Ü	Ü	Ů	Ü	Ü	Ü		Ü	Ü		•
_	Bit/Field		Nam	۵	Ту	na	Reset	Des	cription							
-	olu i i cia		· · · · · ·	·	. y	pc	recoct	D00	onpuon							
	31:27		reserv	ed	R	0	0			ould not r						
										with futu cross a re					ea bit sr	iouia be
						_		·				•				
	26		COMI	P2	R	0	1	Ana	log Com	parator 2	Presen	t				
								Whe	en set, in	dicates th	nat anal	og comp	arator 2	is prese	nt.	
	25		COMI	P1	R	0	1	Ana	log Com	parator 1	Presen	t				
								Whe	en set. in	dicates th	nat anal	oa comp	arator 1	is prese	nt.	
	24		COMI	P0	R	0	1	Ana	log Com	parator 0	Presen	t				
								Whe	en set, in	dicates th	nat anal	og comp	arator 0	is prese	nt.	
	23:20		reserv	ed	R	0	0	Soft	ware sho	ould not r	ely on th	ne value	of a rese	erved bit	. To prov	/ide
										with futu					ed bit sh	nould be
								pres	erved a	cross a re	ad-mod	lify-write	operation	n.		
	19		TIME	R3	R	0	1	Time	er 3 Pres	sent						
								Whe	en set, in	dicates th	nat Gen	eral-Pur _l	pose Tim	ner modu	ıle 3 is p	resent.
	18		TIMEI	R2	R	0	1	Time	er 2 Pres	ent						
	10		1 11VIL	ν			•			dicates th	act Con	oral Dur	noon Tim	or madi	ıla 2 ia n	rocont
								VVIIE	en set, in	uicales li	iai Gen	erai-rui	pose IIII	iei illout	ile z is p	resent.
	17		TIME	R1	R	0	1	Time	er 1 Pres	sent						
								Whe	en set, in	dicates th	nat Gen	eral-Pur _l	pose Tim	ner modu	ıle 1 is p	resent.
	16		TIMEI	R0	R	0	1	Time	er 0 Pres	sent						
								Whe	en set in	dicates th	nat Gen	eral-Puri	oose Tim	ner modi	ıle () is n	resent
															·	
	15:13		reserv	red	R	0	0	com	patibility	ould not rough with futu cross a re	re produ	ucts, the	value of	a reserv		

Bit/Field	Name	Туре	Reset	Description
12	I2C0	RO	1	I2C Module 0 Present
				When set, indicates that I2C module 0 is present.
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	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.

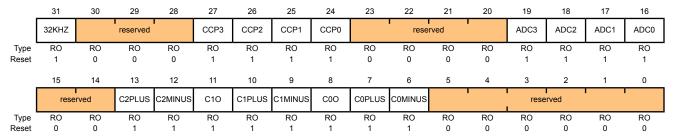
Register 16: Device Capabilities 3 (DC3), offset 0x018

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

Offset 0x018 Type RO, reset 0x8F0F.3FC0



Bit/Field	Name	Туре	Reset	Description
31	32KHZ	RO	1	32KHz Input Clock Available
				When set, indicates an even CCP pin is present and can be used as a 32-KHz input clock.
30:28	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.
27	CCP3	RO	1	CCP3 Pin Present
				When set, indicates that Capture/Compare/PWM pin 3 is present.
26	CCP2	RO	1	CCP2 Pin Present
				When set, indicates that Capture/Compare/PWM pin 2 is present.
25	CCP1	RO	1	CCP1 Pin Present
				When set, indicates that Capture/Compare/PWM pin 1 is present.
24	CCP0	RO	1	CCP0 Pin Present
				When set, indicates that Capture/Compare/PWM pin 0 is present.
23:20	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	ADC3	RO	1	ADC3 Pin Present
				When set, indicates that ADC pin 3 is present.
18	ADC2	RO	1	ADC2 Pin Present
				When set, indicates that ADC pin 2 is present.
17	ADC1	RO	1	ADC1 Pin Present
				When set, indicates that ADC pin 1 is present.

Bit/Field	Name	Туре	Reset	Description
16	ADC0	RO	1	ADC0 Pin Present
				When set, indicates that ADC pin 0 is present.
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	C2PLUS	RO	1	C2+ Pin Present
				When set, indicates that the analog comparator 2 (+) input pin is present.
12	C2MINUS	RO	1	C2- Pin Present
				When set, indicates that the analog comparator 2 (-) input pin is present.
11	C10	RO	1	C1o Pin Present
				When set, indicates that the analog comparator 1 output pin is present.
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	C0PLUS	RO	1	C0+ Pin Present
				When set, indicates that the analog comparator 0 (+) input pin is present.
6	C0MINUS	RO	1	C0- Pin Present
				When set, indicates that the analog comparator 0 (-) input pin is present.
5:0	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.

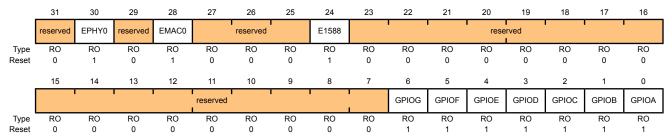
Register 17: Device Capabilities 4 (DC4), offset 0x01C

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 0x5100.007F



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	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.
27:25	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.
24	E1588	RO	1	1588 Capable
				When set, indicates that that EMAC0 is 1588-capable.
23:7	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.

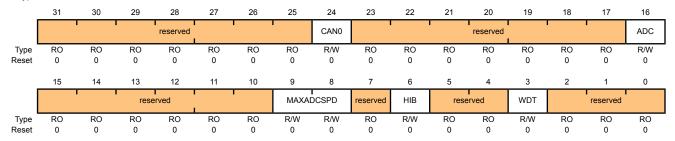
Bit/Field	Name	Туре	Reset	Description
3	GPIOD	RO	1	GPIO Port D Present When set, indicates that GPIO Port D is present.
2	GPIOC	RO	1	GPIO Port C Present When set, indicates that GPIO Port C is present.
1	GPIOB	RO	1	GPIO Port B Present When set, indicates that GPIO Port B is present.
0	GPIOA	RO	1	GPIO 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



Bit/Field	Name	Type	Reset	Description
31:25	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.
24	CAN0	R/W	0	CAN0 Clock Gating Control
				This bit controls the clock gating for CAN unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
23:17	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
				0x3 1M samples/second
				0x2 500K samples/second
				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	HIB	R/W	0	HIB Clock Gating Control
				This bit controls the clock gating for the Hibernation module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
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.

Sleep Mode Clock Gating Control Register 0 (SCGC0)

Base 0x400F.E000 Offset 0x110 Type R/W, reset 0x00000040

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		'	'	reserved	1		'	CAN0				reserved				ADC
Type	RO	RO	RO	RO	RO	RO	RO	R/W	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
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		'	rese	erved	1		MAXA	DCSPD	reserved	HIB	rese	erved	WDT		reserved	
Туре	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	R/W	RO	RO	R/W	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
31:25	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.
24	CAN0	R/W	0	CAN0 Clock Gating Control
				This bit controls the clock gating for CAN unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
23:17	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
				0x3 1M samples/second
				0x2 500K samples/second
				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	HIB	R/W	0	HIB Clock Gating Control
				This bit controls the clock gating for the Hibernation module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
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

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	'	reserved		1	<u>'</u>	CAN0			<u> </u>	reserved				ADC
Type	RO	RO	RO	RO	RO	RO	RO	R/W	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
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		1	rese	erved		1	MAXAI	I DCSPD	reserved	HIB	rese	rved	WDT		reserved	
Type	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	R/W	RO	RO	R/W	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
31:25	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.
24	CAN0	R/W	0	CAN0 Clock Gating Control
				This bit controls the clock gating for CAN unit 0. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
23:17	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.

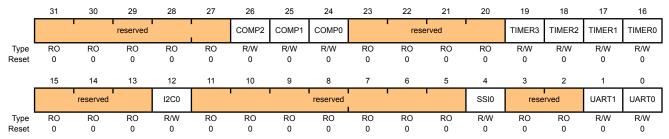
Bit/Field	Name	Туре	Reset	Description
9:8	MAXADCSPD	R/W	0	ADC Sample Speed
				This field sets the rate at which the ADC samples data. You cannot set the rate higher than the maximum rate. You can set the sample rate by setting the MAXADCSPD bit as follows:
				Value Description
				0x3 1M samples/second
				0x2 500K samples/second
				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	HIB	R/W	0	HIB Clock Gating Control
				This bit controls the clock gating for the Hibernation module. If set, the unit receives a clock and functions. Otherwise, the unit is unclocked and disabled.
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



Bit/Field	Name	Type	Reset	Description
31:27	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	COMP2	R/W	0	Analog Comparator 2 Clock Gating
				This bit controls the clock gating for analog comparator 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 unit will generate a bus fault.
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.
23:20	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
19	TIMER3	R/W	0	Timer 3 Clock Gating Control
				This bit controls the clock gating for General-Purpose Timer module 3. 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.
18	TIMER2	R/W	0	Timer 2 Clock Gating Control
				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 unit will generate a bus fault.
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	SSI0	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.

Bit/Field	Name	Type	Reset	Description
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.

Sleep Mode Clock Gating Control Register 1 (SCGC1)

Base 0x400F.E000 Offset 0x114

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		' '	reserved			COMP2	COMP1	COMP0		rese	rved		TIMER3	TIMER2	TIMER1	TIMER0
Type	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	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
		reserved		I2C0		reserved						SSI0	rese	rved	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

Bit/Field	Name	Туре	Reset	Description
31:27	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	COMP2	R/W	0	Analog Comparator 2 Clock Gating
				This bit controls the clock gating for analog comparator 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 unit will generate a bus fault.
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.
23:20	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
19	TIMER3	R/W	0	Timer 3 Clock Gating Control
				This bit controls the clock gating for General-Purpose Timer module 3. 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.
18	TIMER2	R/W	0	Timer 2 Clock Gating Control
				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 unit will generate a bus fault.
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	SSI0	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.

July 25, 2008 Preliminary

Bit/Field	Name	Type	Reset	Description
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 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

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	reserved			COMP2	COMP1	COMP0		resei	ved		TIMER3	TIMER2	TIMER1	TIMER0
Type	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	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
		reserved I2C0 reserved						1		SSI0	rese	rved	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

Bit/Field	Name	Туре	Reset	Description
31:27	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	COMP2	R/W	0	Analog Comparator 2 Clock Gating
				This bit controls the clock gating for analog comparator 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 unit will generate a bus fault.
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.
23:20	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
19	TIMER3	R/W	0	Timer 3 Clock Gating Control
				This bit controls the clock gating for General-Purpose Timer module 3. 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.
18	TIMER2	R/W	0	Timer 2 Clock Gating Control
				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 unit will generate a bus fault.
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	SSI0	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.

Bit/Field	Name	Type	Reset	Description
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.

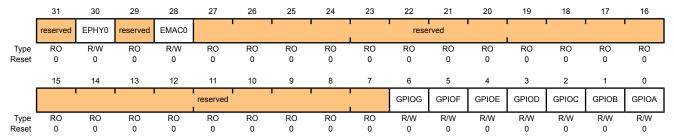
Register 24: Run Mode Clock Gating Control Register 2 (RCGC2), offset 0x108

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			<u>'</u>			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
	<u> </u>		•		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.

Bit/Field	Name	Туре	Reset	Description
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.
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						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					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.

Bit/Field	Name	Туре	Reset	Description
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.
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.

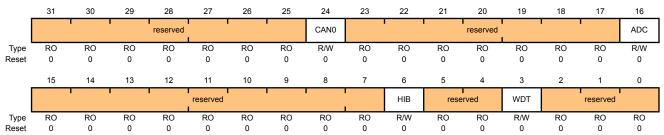
Register 27: Software Reset Control 0 (SRCR0), offset 0x040

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 Type R/W, reset 0x00000000



Bit/Field	Name	Туре	Reset	Description
31:25	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.
24	CAN0	R/W	0	CAN0 Reset Control
				Reset control for CAN unit 0.
23:17	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.
15:7	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	HIB	R/W	0	HIB Reset Control
				Reset control for the Hibernation module.
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 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

Offset 0x044
Type R/W, reset 0x00000000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		'	reserved			COMP2	COMP1	COMP0		rese	rved	'	TIMER3	TIMER2	TIMER1	TIMER0
Type	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	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
		reserved		I2C0		'	1	reserved		'		SSI0	rese	rved	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

Bit/Field	Name	Type	Reset	Description
31:27	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	COMP2	R/W	0	Analog Comp 2 Reset Control
				Reset control for analog comparator 2.
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.
23:20	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	TIMER3	R/W	0	Timer 3 Reset Control
				Reset control for General-Purpose Timer module 3.
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.

Bit/Field	Name	Туре	Reset	Description
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.
				Reset control for 331 unit 0.
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			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	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 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.

Bit/Field	Name	Туре	Reset	Description
0	GPIOA	R/W	0	Port A Reset Control
				Reset control for GPIO Port A.

7 Hibernation Module

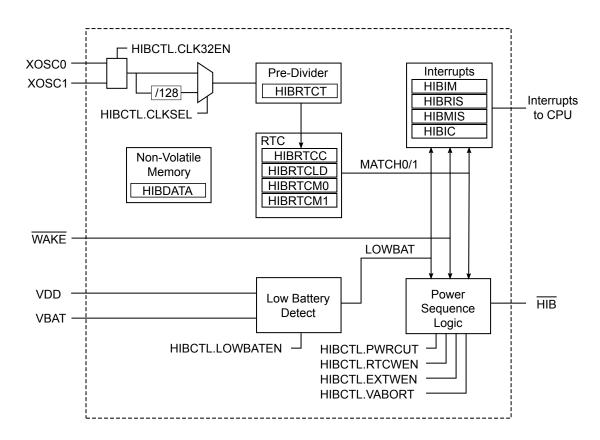
The Hibernation Module manages removal and restoration of power to the rest of the microcontroller to provide a means for reducing power consumption. When the processor and peripherals are idle, power can be completely removed with only the Hibernation Module remaining powered. Power can be restored based on an external signal, or at a certain time using the built-in real-time clock (RTC). The Hibernation module can be independently supplied from a battery or an auxiliary power supply.

The Hibernation module has the following features:

- Power-switching logic to discrete external regulator
- Dedicated pin for waking from an external signal
- Low-battery detection, signaling, and interrupt generation
- 32-bit real-time counter (RTC)
- Two 32-bit RTC match registers for timed wake-up and interrupt generation
- Clock source from a 32.768-kHz external oscillator or a 4.194304-MHz crystal
- RTC predivider trim for making fine adjustments to the clock rate
- 64 32-bit words of non-volatile memory
- Programmable interrupts for RTC match, external wake, and low battery events

7.1 Block Diagram

Figure 7-1. Hibernation Module Block Diagram



7.2 Functional Description

The Hibernation module controls the power to the processor with an enable signal ($\overline{\texttt{HIB}}$) that signals an external voltage regulator to turn off. The Hibernation module power is determined dynamically. The supply voltage of the Hibernation module is the larger of the main voltage source (VDD) or the battery/auxilliary voltage source (VBAT). A voting circuit indicates the larger and an internal power switch selects the appropriate voltage source. The Hibernation module also has a separate clock source to maintain a real-time clock (RTC). Once in hibernation, the module signals an external voltage regulator to turn back on the power when an external pin ($\overline{\texttt{WAKE}}$) is asserted, or when the internal RTC reaches a certain value. The Hibernation module can also detect when the battery voltage is low, and optionally prevent hibernation when this occurs.

Power-up from a power cut to code execution is defined as the regulator turn-on time (specified at $t_{HIB\ TO\ VDD}$ maximum) plus the normal chip POR (see "Hibernation Module" on page 553).

7.2.1 Register Access Timing

Because the Hibernation module has an independent clocking domain, certain registers must be written only with a timing gap between accesses. The delay time is $t_{HIB_REG_WRITE}$, therefore software must guarantee that a delay of $t_{HIB_REG_WRITE}$ is inserted between back-to-back writes to certain

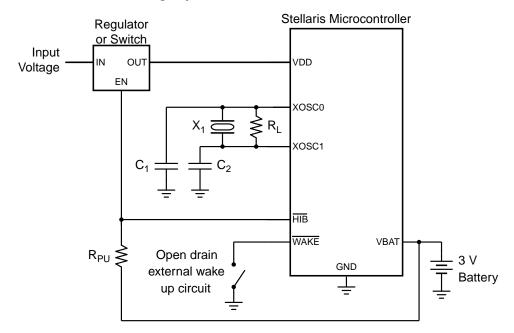
Hibernation registers, or between a write followed by a read to those same registers. There is no restriction on timing for back-to-back reads from the Hibernation module.

7.2.2 Clock Source

The Hibernation module must be clocked by an external source, even if the RTC feature will not be used. An external oscillator or crystal can be used for this purpose. To use a crystal, a 4.194304-MHz crystal is connected to the xosco and xosco pins. This clock signal is divided by 128 internally to produce the 32.768-kHz clock reference. To use a more precise clock source, a 32.768-kHz oscillator can be connected to the xosco pin. See Figure 7-2 on page 127 and Figure 7-3 on page 128. Note that these diagrams only show the connection to the Hibernation pins and not to the full system. See "Hibernation Module" on page 553 for specific values.

The clock source is enabled by setting the CLK32EN bit of the **HIBCTL** register. The type of clock source is selected by setting the CLKSEL bit to 0 for a 4.194304-MHz clock source, and to 1 for a 32.768-kHz clock source. If the bit is set to 0, the input clock is divided by 128, resulting in a 32.768-kHz clock source. If a crystal is used for the clock source, the software must leave a delay of $t_{\text{XOSC_SETTLE}}$ after setting the CLK32EN bit and before any other accesses to the Hibernation module registers. The delay allows the crystal to power up and stabilize. If an oscillator is used for the clock source, no delay is needed.

Figure 7-2. Clock Source Using Crystal



Note: R_{TERM} = Optional series termination resistor.

 R_{PU} = Pull-up resistor (1 M½).

See "Hibernation Module" on page 553 for specific parameter values.

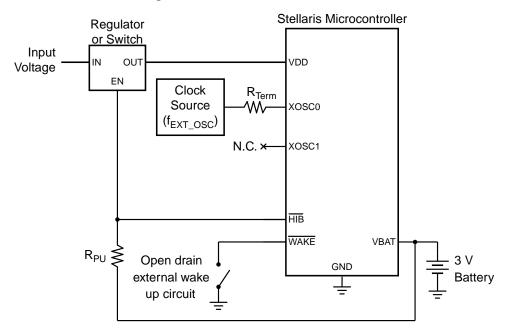


Figure 7-3. Clock Source Using Dedicated Oscillator

Note: X_1 = Crystal frequency is f_{XOSC_XTAL} .

 R_L = Load resistor is R_{XOSC_LOAD} .

 $C_{1,2}$ = Capacitor value derived from crystal vendor load capacitance specifications.

 R_{PU} = Pull-up resistor (1 M½).

See "Hibernation Module" on page 553 for specific parameter values.

7.2.3 Battery Management

The Hibernation module can be independently powered by a battery or an auxiliary power source. The module can monitor the voltage level of the battery and detect when the voltage drops below 2.35 V. When this happens, an interrupt can be generated. The module also can be configured so that it will not go into Hibernate mode if the battery voltage drops below this threshold. Battery voltage is not measured while in Hibernate mode.

Important: System level factors may affect the accuracy of the low battery detect circuit. The designer should consider battery type, discharge characteristics, and a test load during battery voltage measurements.

Note that the Hibernation module draws power from whichever source (VBAT or VDD) has the higher voltage. Therefore, it is important to design the circuit to ensure that VDD is higher that VBAT under nominal conditions or else the Hibernation module draws power from the battery even when VDD is available.

The Hibernation module can be configured to detect a low battery condition by setting the LOWBATEN bit of the **HIBCTL** register. In this configuration, the LOWBAT bit of the **HIBRIS** register will be set when the battery level is low. If the VABORT bit is also set, then the module is prevented from entering Hibernation mode when a low battery is detected. The module can also be configured to generate an interrupt for the low-battery condition (see "Interrupts and Status" on page 130).

7.2.4 **Real-Time Clock**

The Hibernation module includes a 32-bit counter that increments once per second with a proper clock source and configuration (see "Clock Source" on page 127). The 32.768-kHz clock signal is fed into a predivider register which counts down the 32.768-kHz clock ticks to achieve a once per second clock rate for the RTC. The rate can be adjusted to compensate for inaccuracies in the clock source by using the predivider trim register. **HIBRTCT**. This register has a nominal value of 0x7FFF. and is used for one second out of every 64 seconds to divide the input clock. This allows the software to make fine corrections to the clock rate by adjusting the predivider trim register up or down from 0x7FFF. The predivider trim should be adjusted up from 0x7FFF in order to slow down the RTC rate, and down from 0x7FFF in order to speed up the RTC rate.

The Hibernation module includes two 32-bit match registers that are compared to the value of the RTC counter. The match registers can be used to wake the processor from hibernation mode, or to generate an interrupt to the processor if it is not in hibernation.

The RTC must be enabled with the RTCEN bit of the HIBCTL register. The value of the RTC can be set at any time by writing to the **HIBRTCLD** register. The predivider trim can be adjusted by reading and writing the HIBRTCT register. The predivider uses this register once every 64 seconds to adjust the clock rate. The two match registers can be set by writing to the HIBRTCM0 and HIBRTCM1 registers. The RTC can be configured to generate interrupts by using the interrupt registers (see "Interrupts and Status" on page 130).

7.2.5 **Non-Volatile Memory**

The Hibernation module contains 64 32-bit words of memory which are retained during hibernation. This memory is powered from the battery or auxiliary power supply during hibernation. The processor software can save state information in this memory prior to hibernation, and can then recover the state upon waking. The non-volatile memory can be accessed through the HIBDATA registers.

7.2.6 Power Control

Important: The Hibernation Module requires special system implementation considerations since it is intended to power-down all other sections of its host device. The system power-supply distribution and interfaces of the system must be driven to 0 V_{DC} or powered down with the same regulator controlled by HIB. See "Hibernation Module" on page 553 for more details.

The Hibernation module controls power to the processor through the use of the HIB pin, which is intended to be connected to the enable signal of the external regulator(s) providing 3.3 V and/or 2.5 V to the microcontroller. When the HIB signal is asserted by the Hibernation module, the external regulator is turned off and no longer powers the microcontroller. The Hibernation module remains powered from the VBAT supply, which could be a battery or an auxiliary power source. Hibernation mode is initiated by the microcontroller setting the HIBREO bit of the **HIBCTL** register. Prior to doing this, a wake-up condition must be configured, either from the external WAKE pin, or by using an RTC match.

The Hibernation module is configured to wake from the external WAKE pin by setting the PINWEN bit of the HIBCTL register. It is configured to wake from RTC match by setting the RTCWEN bit. Either one or both of these bits can be set prior to going into hibernation. The WAKE pin includes a weak internal pull-up. Note that both the HIB and WAKE pins use the Hibernation module's internal power supply as the logic 1 reference.

When the Hibernation module wakes, the microcontroller will see a normal power-on reset. It can detect that the power-on was due to a wake from hibernation by examining the raw interrupt status register (see "Interrupts and Status" on page 130) and by looking for state data in the non-volatile memory (see "Non-Volatile Memory" on page 129).

When the $\overline{\mathtt{HIB}}$ signal deasserts, enabling the external regulator, the external regulator must reach the operating voltage within t_{HIB} TO VDD.

7.2.7 Interrupts and Status

The Hibernation module can generate interrupts when the following conditions occur:

- Assertion of WAKE pin
- RTC match
- Low battery detected

All of the interrupts are ORed together before being sent to the interrupt controller, so the Hibernate module can only generate a single interrupt request to the controller at any given time. The software interrupt handler can service multiple interrupt events by reading the **HIBMIS** register. Software can also read the status of the Hibernation module at any time by reading the **HIBRIS** register which shows all of the pending events. This register can be used at power-on to see if a wake condition is pending, which indicates to the software that a hibernation wake occurred.

The events that can trigger an interrupt are configured by setting the appropriate bits in the **HIBIM** register. Pending interrupts can be cleared by writing the corresponding bit in the **HIBIC** register.

7.3 Initialization and Configuration

The Hibernation module can be set in several different configurations. The following sections show the recommended programming sequence for various scenarios. The examples below assume that a 32.768-kHz oscillator is used, and thus always show bit 2 (CLKSEL) of the **HIBCTL** register set to 1. If a 4.194304-MHz crystal is used instead, then the CLKSEL bit remains cleared. Because the Hibernation module runs at 32 kHz and is asynchronous to the rest of the system, software must allow a delay of $t_{\text{HIB_REG_WRITE}}$ after writes to certain registers (see "Register Access Timing" on page 126). The registers that require a delay are listed in a note in "Register Map" on page 131 as well as in each register description.

7.3.1 Initialization

The clock source must be enabled first, even if the RTC will not be used. If a 4.194304-MHz crystal is used, perform the following steps:

- Write 0x40 to the HIBCTL register at offset 0x10 to enable the crystal and select the divide-by-128 input path.
- 2. Wait for a time of t_{XOSC_SETTLE} for the crystal to power up and stabilize before performing any other operations with the Hibernation module.

If a 32.678-kHz oscillator is used, then perform the following steps:

- 1. Write 0x44 to the **HIBCTL** register at offset 0x10 to enable the oscillator input.
- No delay is necessary.

The above is only necessary when the entire system is initialized for the first time. If the processor is powered due to a wake from hibernation, then the Hibernation module has already been powered

up and the above steps are not necessary. The software can detect that the Hibernation module and clock are already powered by examining the CLK32EN bit of the **HIBCTL** register.

7.3.2 RTC Match Functionality (No Hibernation)

Use the following steps to implement the RTC match functionality of the Hibernation module:

- Write the required RTC match value to one of the HIBRTCMn registers at offset 0x004 or 0x008.
- 2. Write the required RTC load value to the **HIBRTCLD** register at offset 0x00C.
- 3. Set the required RTC match interrupt mask in the RTCALTO and RTCALT1 bits (bits 1:0) in the HIBIM register at offset 0x014.
- 4. Write 0x0000.0041 to the **HIBCTL** register at offset 0x010 to enable the RTC to begin counting.

7.3.3 RTC Match/Wake-Up from Hibernation

Use the following steps to implement the RTC match and wake-up functionality of the Hibernation module:

- 1. Write the required RTC match value to the **HIBRTCMn** registers at offset 0x004 or 0x008.
- 2. Write the required RTC load value to the **HIBRTCLD** register at offset 0x00C.
- Write any data to be retained during power cut to the HIBDATA register at offsets 0x030-0x12C.
- Set the RTC Match Wake-Up and start the hibernation sequence by writing 0x0000.004F to the HIBCTL register at offset 0x010.

7.3.4 External Wake-Up from Hibernation

Use the following steps to implement the Hibernation module with the external $\overline{\mathtt{WAKE}}$ pin as the wake-up source for the microcontroller:

- 1. Write any data to be retained during power cut to the **HIBDATA** register at offsets 0x030-0x12C.
- 2. Enable the external wake and start the hibernation sequence by writing 0x0000.0056 to the **HIBCTL** register at offset 0x010.

7.3.5 RTC/External Wake-Up from Hibernation

- Write the required RTC match value to the HIBRTCMn registers at offset 0x004 or 0x008.
- 2. Write the required RTC load value to the **HIBRTCLD** register at offset 0x00C.
- Write any data to be retained during power cut to the HIBDATA register at offsets 0x030-0x12C.
- 4. Set the RTC Match/External Wake-Up and start the hibernation sequence by writing 0x0000.005F to the **HIBCTL** register at offset 0x010.

7.4 Register Map

Table 7-1 on page 132 lists the Hibernation registers. All addresses given are relative to the Hibernation Module base address at 0x400F.C000.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Table 7-1. Hibernation Module Register Map

Offset	Name	Type	Reset	Description	See page
0x000	HIBRTCC	RO	0x0000.0000	Hibernation RTC Counter	133
0x004	HIBRTCM0	R/W	0xFFFF.FFFF	Hibernation RTC Match 0	134
0x008	HIBRTCM1	R/W	0xFFFF.FFFF	Hibernation RTC Match 1	135
0x00C	HIBRTCLD	R/W	0xFFFF.FFFF	Hibernation RTC Load	136
0x010	HIBCTL	R/W	0x0000.0000	Hibernation Control	137
0x014	HIBIM	R/W	0x0000.0000	Hibernation Interrupt Mask	139
0x018	HIBRIS	RO	0x0000.0000	Hibernation Raw Interrupt Status	140
0x01C	HIBMIS	RO	0x0000.0000	Hibernation Masked Interrupt Status	141
0x020	HIBIC	R/W1C	0x0000.0000	Hibernation Interrupt Clear	142
0x024	HIBRTCT	R/W	0x0000.7FFF	Hibernation RTC Trim	143
0x030- 0x12C	HIBDATA	R/W	0x0000.0000	Hibernation Data	144

7.5 Register Descriptions

The remainder of this section lists and describes the Hibernation module registers, in numerical order by address offset.

Register 1: Hibernation RTC Counter (HIBRTCC), offset 0x000

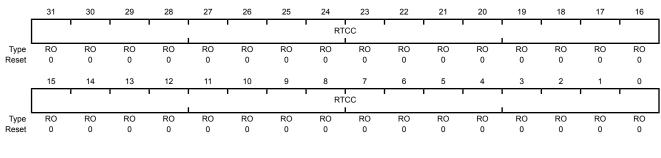
This register is the current 32-bit value of the RTC counter.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation RTC Counter (HIBRTCC)

Base 0x400F.C000 Offset 0x000

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:0	RTCC	RO	0x0000.0000	RTC Counter

A read returns the 32-bit counter value. This register is read-only. To change the value, use the ${\bf HIBRTCLD}$ register.

Register 2: Hibernation RTC Match 0 (HIBRTCM0), offset 0x004

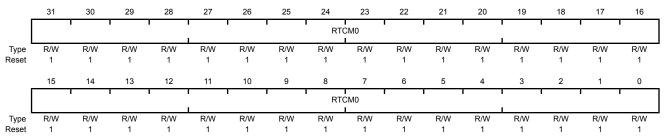
This register is the 32-bit match 0 register for the RTC counter.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation RTC Match 0 (HIBRTCM0)

Base 0x400F.C000 Offset 0x004

Type R/W, reset 0xFFFF.FFF



Bit/Field	Name	Туре	Reset	Description
31:0	RTCM0	R/W	0xFFFF.FFFF	RTC Match 0

A write loads the value into the RTC match register.

A read returns the current match value.

Register 3: Hibernation RTC Match 1 (HIBRTCM1), offset 0x008

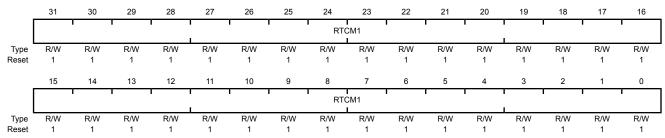
This register is the 32-bit match 1 register for the RTC counter.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation RTC Match 1 (HIBRTCM1)

Base 0x400F.C000 Offset 0x008

Type R/W, reset 0xFFFF.FFF



Bit/Field	Name	Туре	Reset	Description	
31:0	RTCM1	R/W	0xFFFF.FFFF	RTC Match 1	

A write loads the value into the RTC match register.

A read returns the current match value.

Register 4: Hibernation RTC Load (HIBRTCLD), offset 0x00C

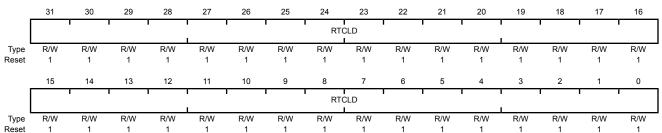
This register is the 32-bit value loaded into the RTC counter.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation RTC Load (HIBRTCLD)

Base 0x400F.C000 Offset 0x00C

Type R/W, reset 0xFFFF.FFF



Bit/Field	Name	Type	Reset	Description	
31:0	RTCLD	R/W	0xFFFF.FFFF	RTC Load	

A write loads the current value into the RTC counter (RTCC).

A read returns the 32-bit load value.

Register 5: Hibernation Control (HIBCTL), offset 0x010

This register is the control register for the Hibernation module.

Hibernation Control (HIBCTL)

Base 0x400F.C000 Offset 0x010 Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	1	1	1	1	1	rese	rved I	1	1) 			
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
		'	1	rese	erved I	'	1	1	VABORT	CLK32EN	LOWBATEN	PINWEN	RTCWEN	CLKSEL	HIBREQ	RTCEN
Туре	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	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	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	VABORT	R/W	0	Power Cut Abort Enable
				Value Description 0 Power cut occurs during a low-battery alert. 1 Power cut is aborted.
6	CLK32EN	R/W	0	32-kHz Oscillator Enable
				Value Description 0 Disabled 1 Enabled This bit must be enabled to use the Hibernation module. If a crystal is used, then software should wait 20 ms after setting this bit to allow the
				crystal to power up and stabilize.
5	LOWBATEN	R/W	0	Low Battery Monitoring Enable Value Description 0 Disabled 1 Enabled
				When set, low battery voltage detection is enabled (VBAT < 2.35 V).
4	PINWEN	R/W	0	External WAKE Pin Enable
				Value Description 0 Disabled 1 Enabled

When set, an external event on the $\overline{\mathtt{WAKE}}$ pin will re-power the device.

Bit/Field	Name	Туре	Reset	Description
3	RTCWEN	R/W	0	RTC Wake-up Enable
				Value Description 0 Disabled 1 Enabled When set, an RTC match event (RTCM0 or RTCM1) will re-power the
				device based on the RTC counter value matching the corresponding match register 0 or 1.
2	CLKSEL	R/W	0	Hibernation Module Clock Select
				Value Description 0 Use Divide by 128 output. Use this value for a 4-MHz crystal. 1 Use raw output. Use this value for a 32-kHz oscillator.
1	HIBREQ	R/W	0	Hibernation Request
				Value Description 0 Disabled 1 Hibernation initiated
				After a wake-up event, this bit is cleared by hardware.
0	RTCEN	R/W	0	RTC Timer Enable
				Value Description 0 Disabled 1 Enabled

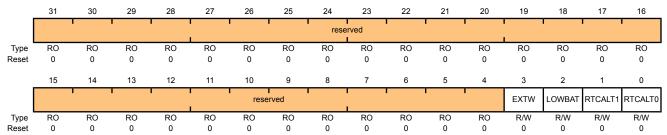
Register 6: Hibernation Interrupt Mask (HIBIM), offset 0x014

This register is the interrupt mask register for the Hibernation module interrupt sources.

Hibernation Interrupt Mask (HIBIM)

Base 0x400F.C000

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



Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x000.0000	Software should not rely on the value of 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	EXTW	R/W	0	External Wake-Up Interrupt Mask
				Value Description
				0 Masked
				1 Unmasked
2	LOWBAT	R/W	0	Low Battery Voltage Interrupt Mask
				Value Description
				0 Masked
				1 Unmasked
1	RTCALT1	R/W	0	RTC Alert1 Interrupt Mask
				Value Description
				0 Masked
				1 Unmasked
0	RTCALT0	R/W	0	RTC Alert0 Interrupt Mask
				Value Description
				0 Masked
				1 Unmasked

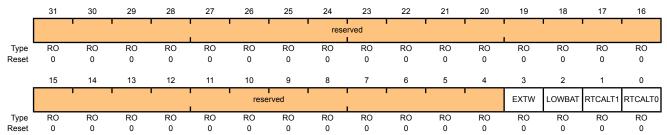
Register 7: Hibernation Raw Interrupt Status (HIBRIS), offset 0x018

This register is the raw interrupt status for the Hibernation module interrupt sources.

Hibernation Raw Interrupt Status (HIBRIS)

Base 0x400F.C000 Offset 0x018

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:4	reserved	RO	0x000.0000	Software should not rely on the value of 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	EXTW	RO	0	External Wake-Up Raw Interrupt Status
2	LOWBAT	RO	0	Low Battery Voltage Raw Interrupt Status
1	RTCALT1	RO	0	RTC Alert1 Raw Interrupt Status
0	RTCALT0	RO	0	RTC Alert0 Raw Interrupt Status

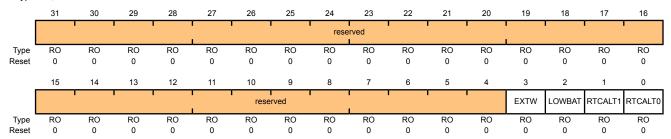
Register 8: Hibernation Masked Interrupt Status (HIBMIS), offset 0x01C

This register is the masked interrupt status for the Hibernation module interrupt sources.

Hibernation Masked Interrupt Status (HIBMIS)

Base 0x400F.C000 Offset 0x01C

Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:4	reserved	RO	0x000.0000	Software should not rely on the value of 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	EXTW	RO	0	External Wake-Up Masked Interrupt Status
2	LOWBAT	RO	0	Low Battery Voltage Masked Interrupt Status
1	RTCALT1	RO	0	RTC Alert1 Masked Interrupt Status
0	RTCALT0	RO	0	RTC Alert0 Masked Interrupt Status

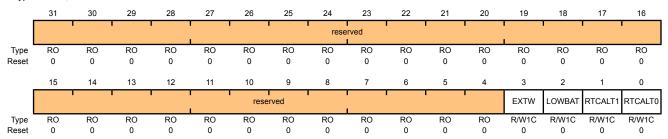
Register 9: Hibernation Interrupt Clear (HIBIC), offset 0x020

This register is the interrupt write-one-to-clear register for the Hibernation module interrupt sources.

Hibernation Interrupt Clear (HIBIC)

Base 0x400F.C000

Offset 0x020 Type R/W1C, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:4	reserved	RO	0x000.0000	Software should not rely on the value of 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	EXTW	R/W1C	0	External Wake-Up Masked Interrupt Clear Reads return an indeterminate value.
2	LOWBAT	R/W1C	0	Low Battery Voltage Masked Interrupt Clear Reads return an indeterminate value.
1	RTCALT1	R/W1C	0	RTC Alert1 Masked Interrupt Clear Reads return an indeterminate value.
0	RTCALT0	R/W1C	0	RTC Alert0 Masked Interrupt Clear Reads return an indeterminate value

Register 10: Hibernation RTC Trim (HIBRTCT), offset 0x024

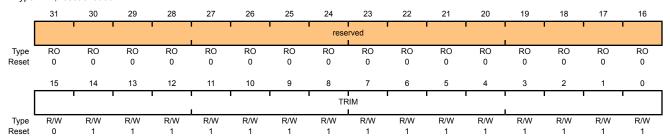
This register contains the value that is used to trim the RTC clock predivider. It represents the computed underflow value that is used during the trim cycle. It is represented as $0x7FFF \pm N$ clock cycles.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation RTC Trim (HIBRTCT)

Base 0x400F.C000 Offset 0x024

Type R/W, reset 0x0000.7FFF



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	TRIM	R/W	0x7FFF	RTC Trim Value

This value is loaded into the RTC predivider every 64 seconds. It is used to adjust the RTC rate to account for drift and inaccuracy in the clock source. The compensation is made by software by adjusting the default value of 0x7FFF up or down.

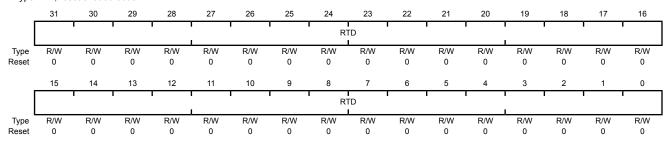
Register 11: Hibernation Data (HIBDATA), offset 0x030-0x12C

This address space is implemented as a 64x32-bit memory (256 bytes). It can be loaded by the system processor in order to store any non-volatile state data and will not lose power during a power cut operation.

Note: HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT, and HIBDATA are on the Hibernation module clock domain and require a delay of t_{HIB_REG_WRITE} between write accesses. See "Register Access Timing" on page 126.

Hibernation Data (HIBDATA)

Base 0x400F.C000 Offset 0x030-0x12C Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:0	RTD	R/W	0x0000.0000	Hibernation Module NV Registers[63:0]

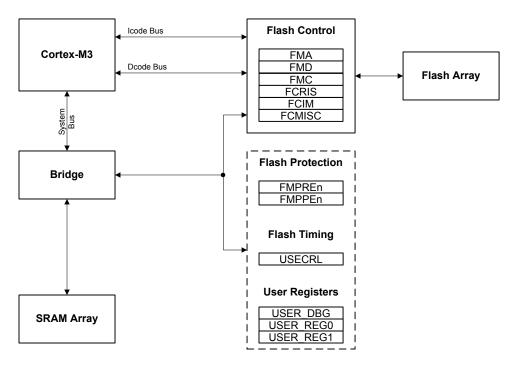
8 Internal Memory

The LM3S8933 microcontroller comes with 64 KB of bit-banded SRAM and 256 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.

8.1 Block Diagram

Figure 8-1 on page 145 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 8-1. Flash Block Diagram



8.2 Functional Description

This section describes the functionality of the SRAM and Flash memories.

8.2.1 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, please refer to Chapter 4, "Memory Map" in the ARM® Cortex™-M3 Technical Reference Manual.

8.2.2 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 563 for a preprogrammed flash-resident utility used to download code to the flash memory of a device without the use of a debug interface.

8.2.2.1 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.

8.2.2.2 Flash Memory Protection

The user is provided two forms of flash protection per 2-KB flash blocks in four 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 set, the block may be executed or read by software or debuggers. If cleared, the block may only be executed and contents of the memory block are prohibited from being accessed as data.

The policies may be combined as shown in Table 8-1 on page 146.

Table 8-1. Flash Protection Policy Combinations

FMPPE	FMPREn	Protection
0		Execute-only protection. The block may only be executed and may not be written or erased. This mode is used to protect code.

FMPPEn	FMPREn	Protection
1	0	The block may be written, erased or executed, but not read. This combination is unlikely to be used.
0		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.

An access that attempts to program or erase a PE-protected block is prohibited. A controller interrupt may be optionally generated (by setting the AMASK bit in the **FIM** register) to alert software developers of poorly behaving software during the development and debug phases.

An access that attempts to read an RE-protected block is prohibited. Such accesses return data filled with all 0s. A controller interrupt may be optionally generated 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. This implements a policy of open access and programmability. The register bits may be changed by writing 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. Details on programming these bits are discussed in "Nonvolatile Register Programming" on page 148.

8.3 Flash Memory Initialization and Configuration

8.3.1 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**.

8.3.1.1 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.

8.3.1.2 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.

8.3.1.3 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.

8.3.2 Nonvolatile Register Programming

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 array and are not affected by an ERASE or MASS ERASE operation. These nonvolatile registers are updated by using the COMT bit in the **FMC** register to activate a write operation. For the **USER_DBG** register, the data to be written must be loaded into the **FMD** register before it is "committed". All other registers are R/W and can have their operation tried before committing them to nonvolatile memory.

Important: These registers can only have bits changed from 1 to 0 by user programming, but can be restored to their factory default values by performing the sequence described in the section called "Recovering a "Locked" Device" on page 55. The mass erase of the main flash array caused by the sequence is performed prior to restoring these registers.

In addition, the **USER_REG0**, **USER_REG1**, and **USER_DBG** use bit 31 (NW) of their respective registers to indicate that they are available for user write. These three registers can only be written once whereas the flash protection registers may be written multiple times. Table 8-2 on page 148 provides the FMA address required for commitment of each of the registers and the source of the data to be written when the COMT bit of 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 8-2. Flash Resident Registers^a

Register to be Committed	FMA Value	Data Source
FMPRE0	0x0000.0000	FMPRE0
FMPRE1	0x0000.0002	FMPRE1
FMPRE2	0x0000.0004	FMPRE2
FMPRE3	0x0000.0008	FMPRE3
FMPPE0	0x0000.0001	FMPPE0
FMPPE1	0x0000.0003	FMPPE1
FMPPE2	0x0000.0005	FMPPE2
FMPPE3	0x0000.0007	FMPPE3
USER_REG0	0x8000.0000	USER_REG0
USER_REG1	0x8000.0001	USER_REG1
USER_DBG	0x7510.0000	FMD

a. Which FMPREn and FMPPEn registers are available depend on the flash size of your particular Stellaris® device.

8.4 Register Map

Table 8-3 on page 149 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** registers are relative to the Flash control base address of 0x400F.D000. The **FMPREn**, **FMPPEn**, **USECRL**, **USER_DBG**, and **USER_REGn** registers are relative to the System Control base address of 0x400F.E000.

Table 8-3. Flash Register Map

Offset	Name	Туре	Reset	Description	See page
Flash Re	gisters (Flash Control Of	fset)			
0x000	FMA	R/W	0x0000.0000	Flash Memory Address	150
0x004	FMD	R/W	0x0000.0000	Flash Memory Data	151
0x008	FMC	R/W	0x0000.0000	Flash Memory Control	152
0x00C	FCRIS	RO	0x0000.0000	Flash Controller Raw Interrupt Status	154
0x010	FCIM	R/W	0x0000.0000	Flash Controller Interrupt Mask	155
0x014	FCMISC	R/W1C	0x0000.0000	Flash Controller Masked Interrupt Status and Clear	156
Flash Reg	gisters (System Control (Offset)			
0x130	FMPRE0	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 0	158
0x200	FMPRE0	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 0	158
0x134	FMPPE0	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 0	159
0x400	FMPPE0	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 0	159
0x140	USECRL	R/W	0x31	USec Reload	157
0x1D0	USER_DBG	R/W	0xFFFF.FFFE	User Debug	160
0x1E0	USER_REG0	R/W	0xFFFF.FFFF	User Register 0	161
0x1E4	USER_REG1	R/W	0xFFFF.FFFF	User Register 1	162
0x204	FMPRE1	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 1	163
0x208	FMPRE2	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 2	164
0x20C	FMPRE3	R/W	0xFFFF.FFFF	Flash Memory Protection Read Enable 3	165
0x404	FMPPE1	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 1	166
0x408	FMPPE2	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 2	167
0x40C	FMPPE3	R/W	0xFFFF.FFFF	Flash Memory Protection Program Enable 3	168

8.5 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.

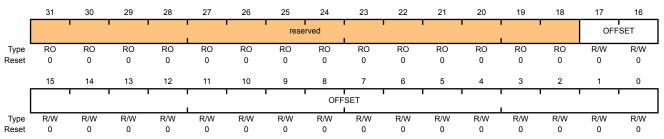
Register 1: Flash Memory Address (FMA), offset 0x000

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
31:18	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.
17:0	OFFSET	R/W	0x0	Address Offset

Address offset in flash where operation is performed, except for nonvolatile registers (see "Nonvolatile Register Programming" on page 148 for details on values for this field).

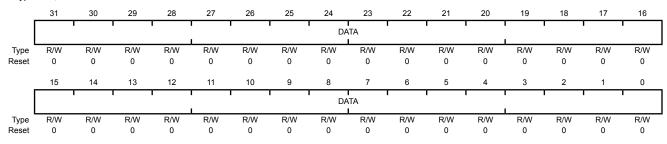
Register 2: Flash Memory Data (FMD), offset 0x004

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 150). If the access is a write access, the data contained in the **Flash Memory Data (FMD)** register (see page 151) 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)

Name

COMT

MERASE

Type

R/W

R/W

Reset

0

0

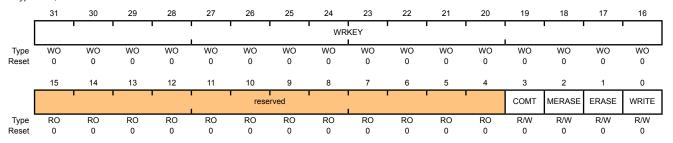
Base 0x400F.D000 Offset 0x008

Bit/Field

3

2

Type R/W, reset 0x0000.0000



31:16	WRKEY	WO	0x0	Flash Write Key
				This field contains a write key, which is used to minimize the incidence of accidental flash writes. The value 0xA442 must be written into this field for a write to occur. Writes to the FMC register without this WRKEY value are ignored. A read of this field returns the value 0.
15:4	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

Description

Commit Register Value

preserved across a read-modify-write operation.

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 μ s.

Mass Erase Flash Memory

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	Type	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.

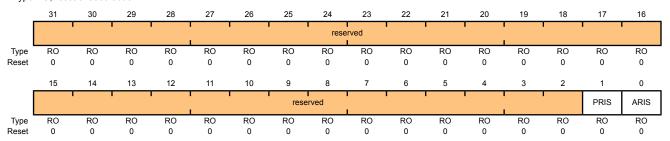
Register 4: Flash Controller Raw Interrupt Status (FCRIS), offset 0x00C

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

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 indicates the current state of the programming cycle. If set, the programming cycle completed; if cleared, the programming cycle has not completed. Programming cycles are either write or erase actions generated through the Flash Memory Control (FMC) register bits (see page 152).
0	ARIS	RO	0	Access Raw Interrupt Status

This bit indicates if the flash was improperly accessed. If set, the program tried to access the flash counter to the policy as set in the Flash Memory Protection Read Enable (FMPREn) and Flash Memory Protection Program Enable (FMPPEn) registers. Otherwise, no access has tried to improperly access the flash.

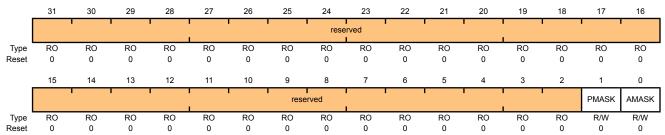
Register 5: Flash Controller Interrupt Mask (FCIM), offset 0x010

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 controller. If set, a programming-generated interrupt is promoted to the controller. Otherwise, interrupts are recorded but suppressed from the controller.
0	AMASK	R/W	0	Access Interrupt Mask

This bit controls the reporting of the access raw interrupt status to the controller. If set, an access-generated interrupt is promoted to the controller. Otherwise, interrupts are recorded but suppressed from the 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)

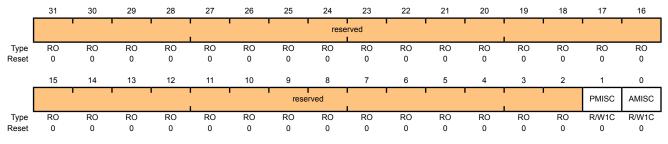
Name

Type

Reset

Base 0x400F.D000 Offset 0x014
Type R/W1C, reset 0x0000.0000

Bit/Field



		• •		·
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
				This bit indicates whether an interrupt was signaled because a programming cycle completed and was not masked. This bit is cleared by writing a 1. The PRIS bit in the FCRIS register (see page 154) is also cleared when the PMISC bit is cleared.
0	AMISC	R/W1C	0	Access Masked Interrupt Status and Clear

Description

This bit indicates whether an interrupt was signaled because an improper access was attempted and was not masked. This bit is cleared by writing a 1. The ARIS bit in the FCRIS register is also cleared when the AMISC bit is cleared.

8.6 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.

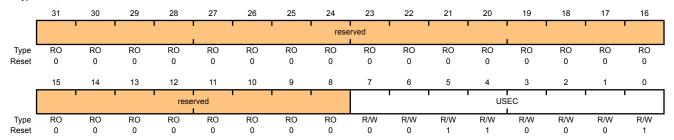
Register 7: USec Reload (USECRL), offset 0x140

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	LISEC	DAM	0v21	Microsecond Peload Value

MHz -1 of the controller clock when the flash is being erased or programmed.

If the maximum system frequency is being used, ${\tt USEC}$ 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

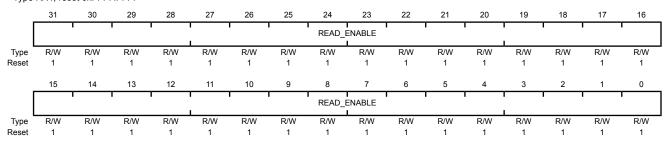
Note: This register is aliased for backwards compatability.

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). 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 0 (FMPRE0)

Base 0x400F.E000 Offset 0x130 and 0x200 Type R/W, reset 0xFFFF.FFFF



Bit/Field	Name	Type	Reset	Description
31:0	READ_ENABLE	R/W	0xFFFFFFF	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

0xFFFFFFF Enables 256 KB of flash.

Register 9: Flash Memory Protection Program Enable 0 (FMPPE0), offset 0x134 and 0x400

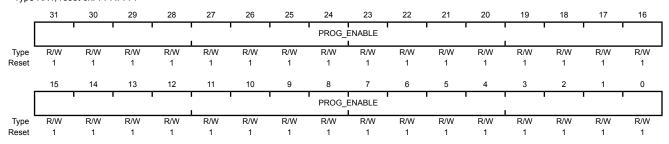
Note: This register is aliased for backwards compatability.

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). 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 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 Enables 256 KB of flash.

Register 10: User Debug (USER DBG), offset 0x1D0

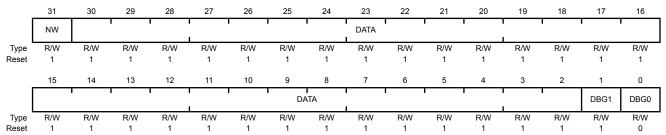
Note: Offset is relative to System Control base address of 0x400FE000.

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 NOTWRITTEN bit (bit 31) indicates that the register is available to be written and is controlled through hardware to ensure that the register is only written once.

User Debug (USER DBG)

Base 0x400F.E000 Offset 0x1D0

Type R/W, reset 0xFFFF.FFFE



Bit/Field	Name	Type	Reset	Description
31	NW	R/W	1	User Debug Not Written Specifies that this 32-bit dword has not been written.
30:2	DATA	R/W	0x1FFFFFF	User Data
				Contains the user data value. This field is initialized to all 1s and can only be written once.
1	DBG1	R/W	1	Debug Control 1
				The $\mathtt{DBG1}$ bit must be 1 and $\mathtt{DBG0}$ 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

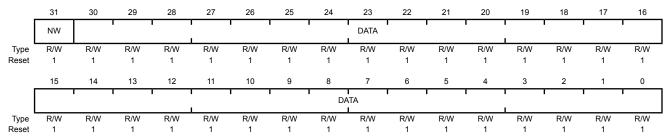
Note: Offset is relative to System Control base address of 0x400FE000.

This register provides 31 bits of user-defined data that is non-volatile and can only be written once. Bit 31 indicates that the register is available to be written and is controlled through hardware to ensure that the register is only written once. 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.

User Register 0 (USER_REG0)

Base 0x400F.E000 Offset 0x1E0

Type R/W, reset 0xFFFF.FFF



Bit/Field	Name	Туре	Reset	Description
31	NW	R/W	1	Not Written
				Specifies that this 32-bit dword has not been written.
30:0	DATA	R/W (x7FFFFFF	User Data

Contains the user data value. This field is initialized to all 1s and can only be written once.

Register 12: User Register 1 (USER_REG1), offset 0x1E4

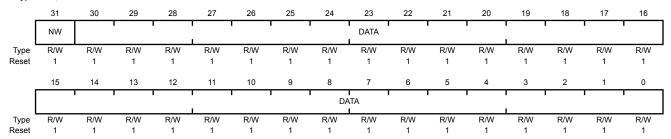
Note: Offset is relative to System Control base address of 0x400FE000.

This register provides 31 bits of user-defined data that is non-volatile and can only be written once. Bit 31 indicates that the register is available to be written and is controlled through hardware to ensure that the register is only written once. 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.

User Register 1 (USER_REG1)

Base 0x400F.E000 Offset 0x1E4

Type R/W, reset 0xFFFF.FFF



Bit/Field	Name	Type	Reset	Description
31	NW	R/W	1	Not Written
				Specifies that this 32-bit dword has not been written.
30:0	DATA	R/W	0x7FFFFFF	User Data

Contains the user data value. This field is initialized to all 1s and can only be written once.

Register 13: Flash Memory Protection Read Enable 1 (FMPRE1), offset 0x204

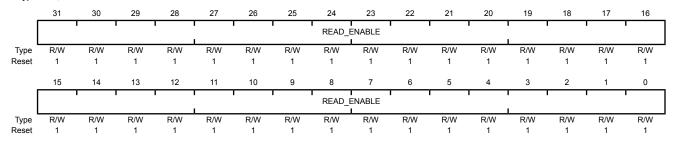
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the read-only protection bits for each 2-KB flash block (**FMPPEn** stores the execute-only bits). 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 1 (FMPRE1)

Base 0x400F.E000 Offset 0x204

Type R/W, reset 0xFFFF.FFFF



Bit/Field Name Type Reset Description

31:0 READ_ENABLE R/W 0xFFFFFFF 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

Register 14: Flash Memory Protection Read Enable 2 (FMPRE2), offset 0x208

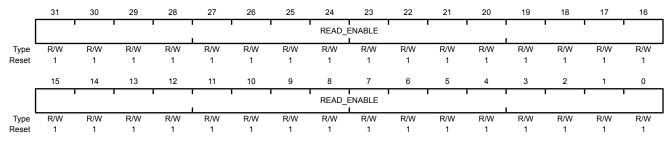
Note: Offset is relative to System Control base address of 0x400FE000.

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). 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 2 (FMPRE2)

Base 0x400F.E000 Offset 0x208

Type R/W, reset 0xFFFF.FFFF



Bit/Field Name Type Reset Description

31:0 READ_ENABLE R/W 0xFFFFFFF 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

Register 15: Flash Memory Protection Read Enable 3 (FMPRE3), offset 0x20C

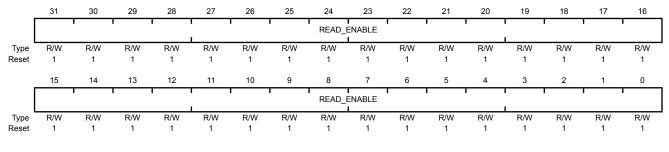
Note: Offset is relative to System Control base address of 0x400FE000.

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 0xFFFF.FFF



Bit/Field Name Type Reset Description

31:0 READ_ENABLE R/W 0xFFFFFFF 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

Register 16: Flash Memory Protection Program Enable 1 (FMPPE1), offset 0x404

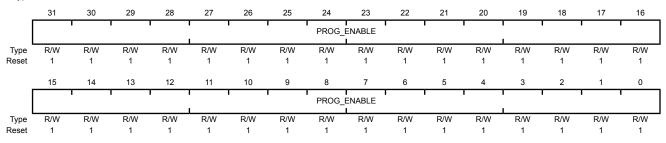
Note: Offset is relative to System Control base address of 0x400FE000.

This register stores the execute-only protection bits for each 2-KB flash block (**FMPREn** stores the execute-only bits). 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 1 (FMPPE1)

Base 0x400F.E000 Offset 0x404

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

Register 17: Flash Memory Protection Program Enable 2 (FMPPE2), offset 0x408

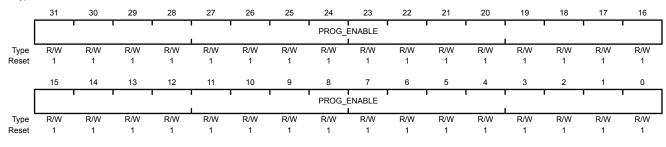
Note: Offset is relative to System Control base address of 0x400FE000.

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 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

Register 18: Flash Memory Protection Program Enable 3 (FMPPE3), offset 0x40C

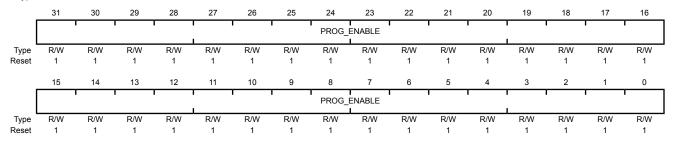
Note: Offset is relative to System Control base address of 0x400FE000.

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 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

9 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, and Port G,). The GPIO module supports 6-36 programmable input/output pins, depending on the peripherals being used.

The GPIO module has the following features:

- Programmable control for GPIO interrupts
 - Interrupt generation masking
 - Edge-triggered on rising, falling, or both
 - Level-sensitive on High or Low values
- 5-V-tolerant input/outputs
- Bit masking in both read and write operations through address lines
- 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

9.1 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.

Each GPIO port is a separate hardware instantiation of the same physical block (see Figure 9-1 on page 170). The LM3S8933 microcontroller contains seven ports and thus seven of these physical GPIO blocks.

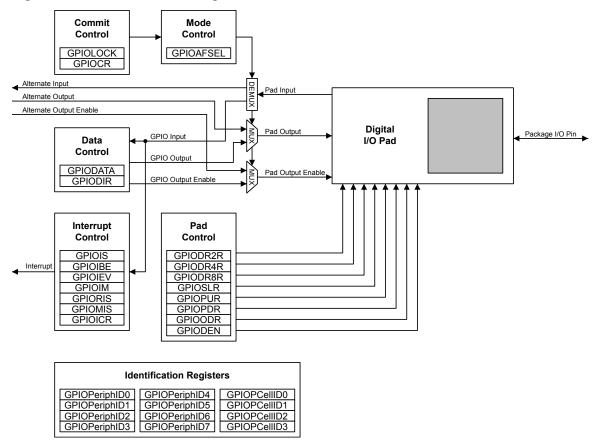


Figure 9-1. GPIO Port Block Diagram

9.1.1 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.

9.1.1.1 Data Direction Operation

The **GPIO Direction (GPIODIR)** register (see page 178) 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.

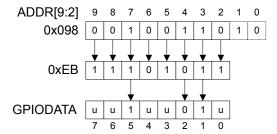
9.1.1.2 Data Register Operation

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 177) 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.

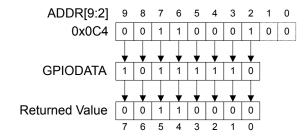
For example, writing a value of 0xEB to the address GPIODATA + 0x098 would yield as shown in Figure 9-2 on page 171, where u is data unchanged by the write.

Figure 9-2. 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 9-3 on page 171.

Figure 9-3. GPIODATA Read Example



9.1.2 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 179)
- GPIO Interrupt Both Edges (GPIOIBE) register (see page 180)
- GPIO Interrupt Event (GPIOIEV) register (see page 181)

Interrupts are enabled/disabled via the GPIO Interrupt Mask (GPIOIM) register (see page 182).

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 183 and page 184). 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 ARM Integrated Nested Vectored Interrupt Controller (NVIC) Interrupt Set Enable (SETNA) 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 B4, and wait for the ADC interrupt or the ADC interrupt needs to be disabled in the SETNA register and the PortB interrupt handler polls the ADC registers until the conversion is completed.

Interrupts are cleared by writing a 1 to the appropriate bit of the **GPIO Interrupt Clear (GPIOICR)** register (see page 185).

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.

9.1.3 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 186), 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.

9.1.4 Commit Control

The commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Writes to protected bits of the **GPIO Alternate Function Select (GPIOAFSEL)** register (see page 186) are not committed to storage unless the **GPIO Lock (GPIOLOCK)** register (see page 196) has been unlocked and the appropriate bits of the **GPIO Commit (GPIOCR)** register (see page 197) have been set to 1.

9.1.5 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**, **GPIOPUR**, **GPIOPDR**, **GPIOPDR**, and **GPIODEN** registers. These registers control drive strength, open-drain configuration, pull-up and pull-down resistors, slew-rate control and digital input 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.

9.1.6 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.

9.2 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 9-1 on page 173 shows all possible configurations of the GPIO pads and the control register settings required to achieve them. Table 9-2 on page 173 shows how a rising edge interrupt would be configured for pin 2 of a GPIO port.

Table 9-1. GPIO Pad Configuration Examples

Configuration	GPIO Register Bit Value ^a											
	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 Input (GPIO)	0	0	1	1	Х	Х	Х	Х	Х	X		
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 (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 9-2. GPIO Interrupt Configuration Example

Register		Pin 2 Bit Value ^a								
	Interrupt Event Trigger	7	6	5	4	3	2	1	0	
GPIOIS	0=edge 1=level	Х	Х	Х	Х	Х	0	Х	Х	
GPIOIBE	0=single edge 1=both edges	Х	Х	Х	Х	Х	0	Х	Х	

^{?=}Can be either 0 or 1, depending on the configuration

Register		Pin 2 Bit Value ^a								
Event	Interrupt Event Trigger	7	6	5	4	3	2	1	0	
GPIOIEV	0=Low level, or negative edge 1=High level, or positive edge		Х	Х	Х	X	1	Х	Х	
GPIOIM	0=masked 1=not masked	0	0	0	0	0	1	0	0	

a. X=Ignored (don't care bit)

9.3 Register Map

Table 9-3 on page 175 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: 0x4000.4000

GPIO Port B: 0x4000.5000

GPIO Port C: 0x4000.6000

GPIO Port D: 0x4000.7000

GPIO Port E: 0x4002.4000

GPIO Port F: 0x4002.5000

GPIO Port G: 0x4002.6000

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 9-3. GPIO Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	GPIODATA	R/W	0x0000.0000	GPIO Data	177
0x400	GPIODIR	R/W	0x0000.0000	GPIO Direction	178
0x404	GPIOIS	R/W	0x0000.0000	GPIO Interrupt Sense	179
0x408	GPIOIBE	R/W	0x0000.0000	GPIO Interrupt Both Edges	180
0x40C	GPIOIEV	R/W	0x0000.0000	GPIO Interrupt Event	181
0x410	GPIOIM	R/W	0x0000.0000	GPIO Interrupt Mask	182
0x414	GPIORIS	RO	0x0000.0000	GPIO Raw Interrupt Status	183
0x418	GPIOMIS	RO	0x0000.0000	GPIO Masked Interrupt Status	184
0x41C	GPIOICR	W1C	0x0000.0000	GPIO Interrupt Clear	185
0x420	GPIOAFSEL	R/W	-	GPIO Alternate Function Select	186
0x500	GPIODR2R	R/W	0x0000.00FF	GPIO 2-mA Drive Select	188
0x504	GPIODR4R	R/W	0x0000.0000	GPIO 4-mA Drive Select	189
0x508	GPIODR8R	R/W	0x0000.0000	GPIO 8-mA Drive Select	190
0x50C	GPIOODR	R/W	0x0000.0000	GPIO Open Drain Select	191
0x510	GPIOPUR	R/W	-	GPIO Pull-Up Select	192
0x514	GPIOPDR	R/W	0x0000.0000	GPIO Pull-Down Select	193
0x518	GPIOSLR	R/W	0x0000.0000	GPIO Slew Rate Control Select	194
0x51C	GPIODEN	R/W	-	GPIO Digital Enable	195
0x520	GPIOLOCK	R/W	0x0000.0001	GPIO Lock	196
0x524	GPIOCR	-	-	GPIO Commit	197
0xFD0	GPIOPeriphID4	RO	0x0000.0000	GPIO Peripheral Identification 4	199
0xFD4	GPIOPeriphID5	RO	0x0000.0000	GPIO Peripheral Identification 5	200
0xFD8	GPIOPeriphID6	RO	0x0000.0000	GPIO Peripheral Identification 6	201
0xFDC	GPIOPeriphID7	RO	0x0000.0000	GPIO Peripheral Identification 7	202
0xFE0	GPIOPeriphID0	RO	0x0000.0061	GPIO Peripheral Identification 0	203
0xFE4	GPIOPeriphID1	RO	0x0000.0000	GPIO Peripheral Identification 1	204
0xFE8	GPIOPeriphID2	RO	0x0000.0018	GPIO Peripheral Identification 2	205
0xFEC	GPIOPeriphID3	RO	0x0000.0001	GPIO Peripheral Identification 3	206
0xFF0	GPIOPCellID0	RO	0x0000.000D	GPIO PrimeCell Identification 0	207
0xFF4	GPIOPCellID1	RO	0x0000.00F0	GPIO PrimeCell Identification 1	208
0xFF8	GPIOPCellID2	RO	0x0000.0005	GPIO PrimeCell Identification 2	209
0xFFC	GPIOPCellID3	RO	0x0000.00B1	GPIO PrimeCell Identification 3	210

9.4 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 178).

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.

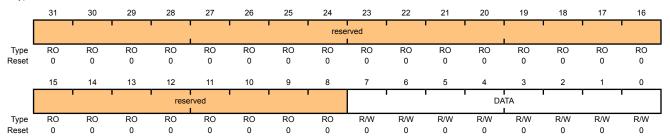
Similarly, the values read from this register are determined for each bit by the mask bit derived from the address used to access the data register, bits [9:2]. Bits that are 1 in the address mask cause the corresponding bits in **GPIODATA** to be read, and bits that are 0 in the address mask cause the 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	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	0x00	GPIO Data

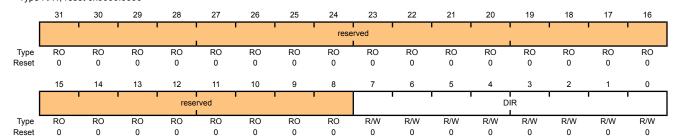
This register is virtually mapped to 256 locations in the address space. To facilitate the reading and writing of data to these registers by independent drivers, the data read from and the data written to the registers are masked by the eight address lines $\mathtt{ipaddr}[9:2]$. Reads from this register 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 170 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 GPIO Port G base: 0x4002.6000 Offset 0x400 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	DIR	R/W	0x00	GPIO Data Direction

The DIR values are defined as follows:

Value Description

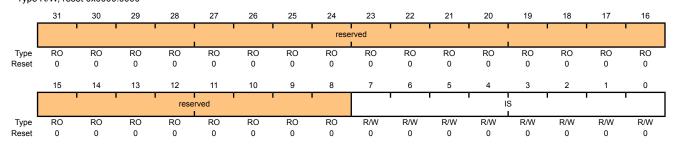
- 0 Pins are inputs.
- 1 Pins are outputs.

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



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 IS values are defined as follows:

Value Description

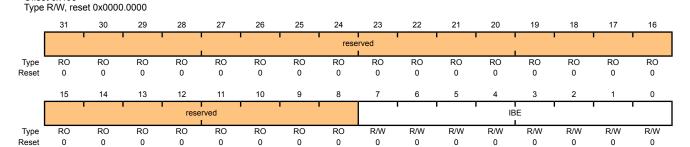
- 0 Edge on corresponding pin is detected (edge-sensitive).
- 1 Level on corresponding pin is detected (level-sensitive).

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 179) 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 181). 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



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	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 181).
- 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

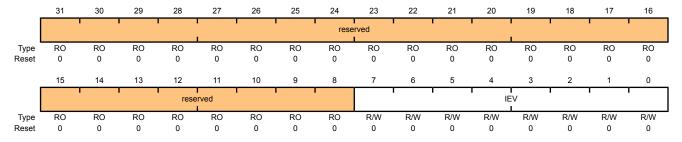
The **GPIOIEV** register is the interrupt event register. Bits set to High in **GPIOIEV** configure the corresponding pin to detect rising edges or high levels, depending on the corresponding bit value in the GPIO Interrupt Sense (GPIOIS) register (see page 179). Clearing a bit configures the pin to 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	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	IEV	R/W	0x00	GPIO Interrupt Event

The IEV values are defined as follows:

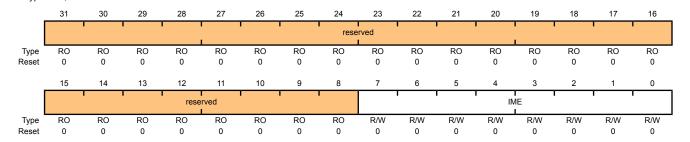
- Falling edge or Low levels on corresponding pins trigger interrupts.
- Rising edge or High levels on corresponding pins trigger interrupts.

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 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	IME	R/W	0x00	GPIO Interrupt Mask Enable

The 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 182). Bits read as zero indicate that corresponding input pins have not initiated an interrupt. All bits are cleared by a reset.

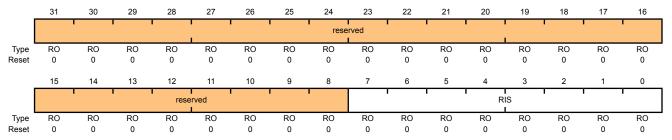
GPIO Raw Interrupt Status (GPIORIS)

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

D:4/E:-14

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 ARM Integrated Nested Vectored Interrupt Controller (NVIC) Interrupt Set Enable (SETNA) 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 B4, and wait for the ADC interrupt or the ADC interrupt needs to be disabled in the SETNA register and the PortB interrupt handler polls the ADC registers until the conversion is completed.

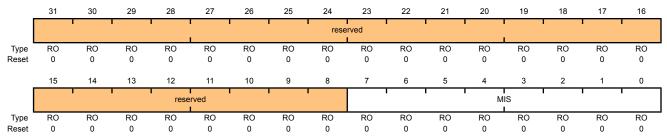
GPIOMIS is the state of the interrupt after masking.

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.

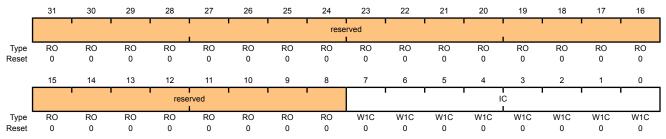
Register 9: GPIO Interrupt Clear (GPIOICR), offset 0x41C

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	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	IC	W1C	0x00	GPIO Interrupt Clear

The IC values are defined as follows:

- Corresponding interrupt is unaffected.
- 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 commit control registers provide a layer of protection against accidental programming of critical hardware peripherals. Writes to protected bits of the GPIO Alternate Function Select (GPIOAFSEL) register (see page 186) are not committed to storage unless the GPIO Lock (GPIOLOCK) register (see page 196) has been unlocked and the appropriate bits of the GPIO Commit (GPIOCR) register (see page 197) have been set to 1.

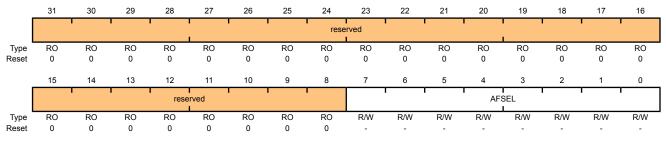
Important: All GPIO pins are tri-stated by default (GPIOAFSEL=0, GPIODEN=0, GPIOPDR=0, and GPIOPUR=0), with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). The JTAG/SWD pins default to their JTAG/SWD functionality (GPIOAFSEL=1. GPIODEN=1 and GPIOPUR=1). A Power-On-Reset (FOR) or asserting RST puts both groups of pins back to their default state.

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)

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

Type R/W, reset



Description

Bit/Field Name Reset Type 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	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:

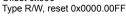
The default reset value for the **GPIOAFSEL**, **GPIOPUR**, and **GPIODEN** registers are 0x0000.0000 for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins default to JTAG/SWD functionality. Because of this, the default reset value of these registers for GPIO Port B is 0x0000.0080 while the default reset value for Port C is 0x0000.000F.

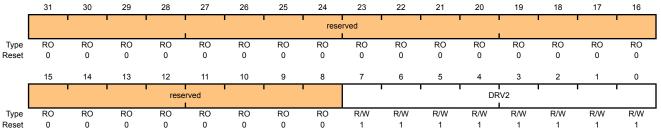
Register 11: GPIO 2-mA Drive Select (GPIODR2R), offset 0x500

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





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.

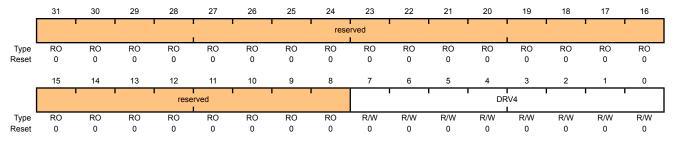
Register 12: GPIO 4-mA Drive Select (GPIODR4R), offset 0x504

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 Offset 0x504

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

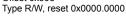
A write of 1 to either **GPIODR2[n]** or **GPIODR8[n]** clears the corresponding 4-mA enable bit. The change is effective on the second clock cycle after the write.

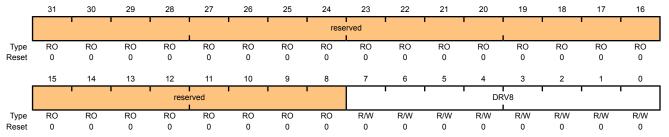
Register 13: GPIO 8-mA Drive Select (GPIODR8R), offset 0x508

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

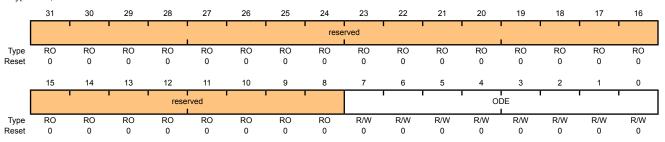
The **GPIOODR** register is the open drain control register. Setting a bit in this register enables the open drain configuration of the corresponding GPIO pad. When open drain mode is enabled, the corresponding bit should also be set in the **GPIO Digital Input Enable (GPIODEN)** register (see page 195). 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 set to 0; and as an open drain output when set to 1.

When using the I²C module, in addition to configuring the pin to open drain, the **GPIO Alternate Function Select (GPIOAFSEL)** register bit for the I²C clock and data pins should be set to 1 (see examples in "Initialization and Configuration" on page 173).

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

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	ODE	R/W	0x00	Output Pad Open Drain Enable

The ODE values are defined as follows:

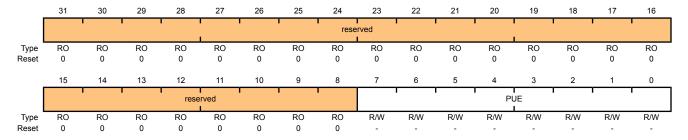
- 0 Open drain configuration is disabled.
- 1 Open drain configuration is enabled.

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 193).

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	PUF	R/W	_	Pad Weak Pull-Up Enable

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:

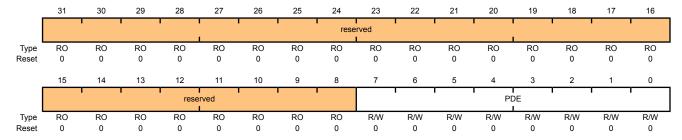
The default reset value for the **GPIOAFSEL**, **GPIOPUR**, and **GPIODEN** registers are 0x0000.0000 for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). These five pins default to JTAG/SWD functionality. Because of this, the default reset value of these registers for GPIO 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 192).

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

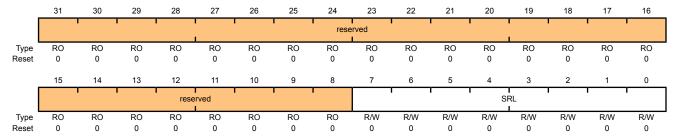
A write of 1 to GPIOPUR[n] clears the corresponding GPIOPDR[n] enables. The change is effective on the second clock cycle after the write.

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 190).

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	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	SRL	R/W	0x00	Slew Rate Limit Enable (8-mA drive only)

The SRL values are defined as follows:

- Slew rate control disabled.
- Slew rate control enabled.

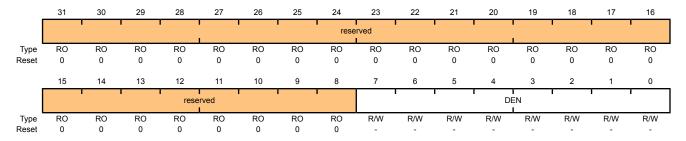
Register 18: GPIO Digital Enable (GPIODEN), offset 0x51C

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.
- 1 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

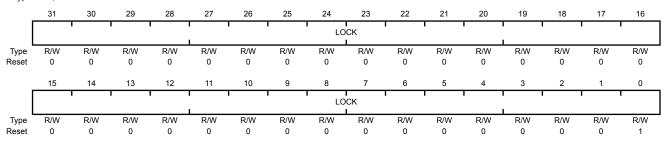
The **GPIOLOCK** register enables write access to the **GPIOCR** register (see page 197). Writing 0x1ACC.E551 to the **GPIOLOCK** register will unlock the **GPIOCR** register. Writing any other value to the **GPIOLOCK** register re-enables the locked state. Reading the **GPIOLOCK** register returns the 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

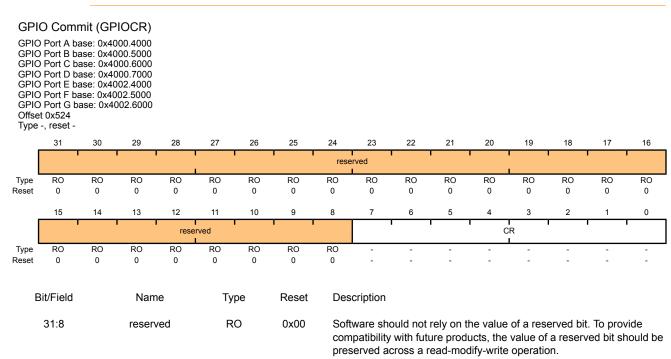
Register 20: GPIO Commit (GPIOCR), offset 0x524

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.



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.

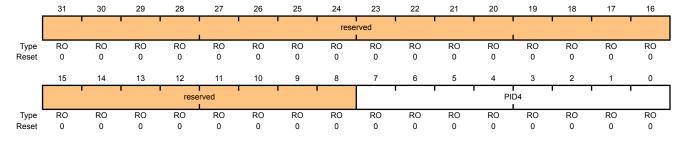
The default reset value for the **GPIOCR** register is 0x0000.00FF for all GPIO pins, with the exception of the five JTAG/SWD pins (PB7 and PC[3:0]). To ensure that the JTAG port is not accidentally programmed as a GPIO, these five pins default to non-committable. Because of this, the default reset value of **GPIOCR** for GPIO Port B is 0x0000.007F while the default reset value of GPIOCR for Port C is 0x0000.00FO.

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



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]

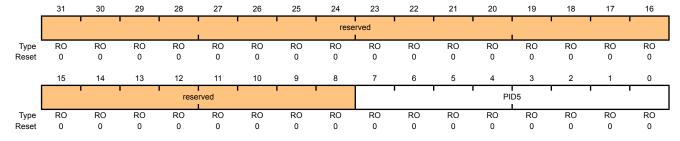
Register 22: GPIO Peripheral Identification 5 (GPIOPeriphID5), offset 0xFD4

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 F base: 0x4002.5000 GPIO Port G base: 0x4002.6000 Offset 0xFD4

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	GPIO Peripheral ID Register[15:8]

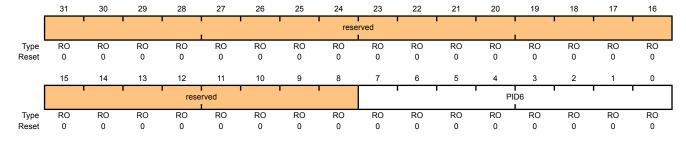
Register 23: GPIO Peripheral Identification 6 (GPIOPeriphID6), offset 0xFD8

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]

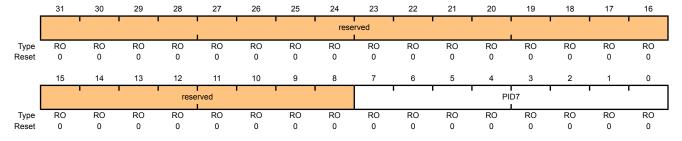
Register 24: GPIO Peripheral Identification 7 (GPIOPeriphID7), offset 0xFDC

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 G base: 0x4002.5000 GPIO Port G base: 0x4002.6000

Offset 0xFDC Type RO, reset 0x0000.0000



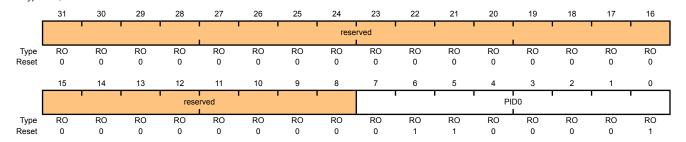
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID7	RO	0x00	GPIO Peripheral ID Register[31:24]

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 GPIO Port G base: 0x4002.6000 Offset 0xFEO Type RO, reset 0x0000.0061



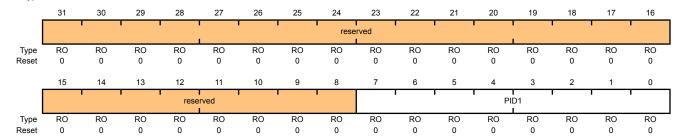
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID0	RO	0x61	GPIO Peripheral ID Register[7:0]

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



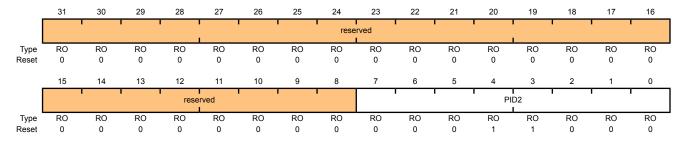
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID1	RO	0x00	GPIO Peripheral ID Register[15:8]

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 GPIO Port G base: 0x4002.6000 Offset 0xFE8 Type RO, reset 0x0000.0018



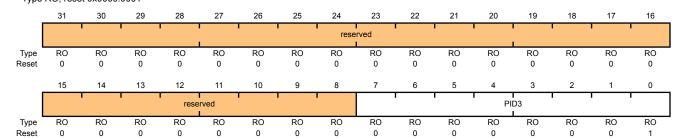
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID2	RO	0x18	GPIO Peripheral ID Register[23:16]

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 Offset 0xFEC Type RO, reset 0x0000.0001



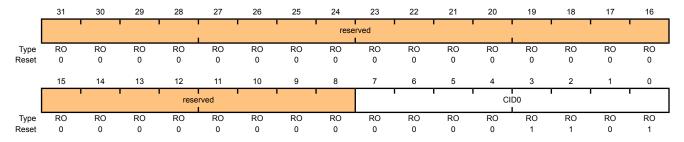
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	GPIO Peripheral ID Register[31:24]

Register 29: GPIO PrimeCell Identification 0 (GPIOPCellID0), offset 0xFF0

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 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



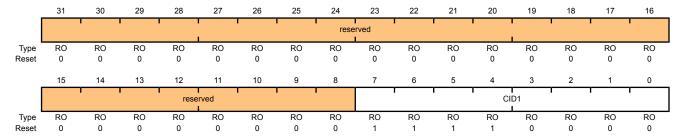
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	GPIO PrimeCell ID Register[7:0]

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 Offset 0xFF4 Type RO, reset 0x0000.00F0



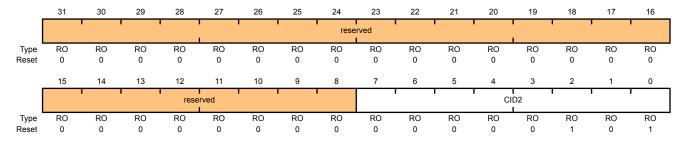
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	GPIO PrimeCell ID Register[15:8]

Register 31: GPIO PrimeCell Identification 2 (GPIOPCellID2), offset 0xFF8

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 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



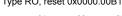
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	GPIO PrimeCell ID Register[23:16]

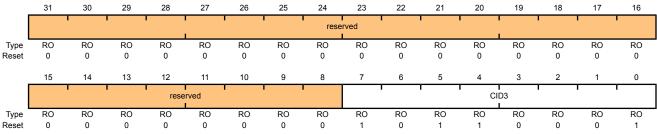
Register 32: GPIO PrimeCell Identification 3 (GPIOPCellID3), offset 0xFFC

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 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	GPIO PrimeCell ID Register[31:24]

10 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 four GPTM blocks (Timer0, Timer1, Timer 2, and Timer 3). 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). Timers can also be used to trigger analog-to-digital (ADC) conversions. The 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 General-Purpose Timer Module is one timing resource available on the Stellaris[®] microcontrollers. Other timer resources include the System Timer (SysTick) (see "System Timer (SysTick)" on page 42).

The following modes are supported:

- 32-bit Timer modes
 - Programmable one-shot timer
 - Programmable periodic timer
 - Real-Time Clock using 32.768-KHz input clock
 - Software-controlled event stalling (excluding RTC mode)
- 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
 - Software-controlled event stalling
- 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

10.1 Block Diagram

Note: In Figure 10-1 on page 212, the specific CCP pins available depend on the Stellaris[®] device. See Table 10-1 on page 212 for the available CCPs.

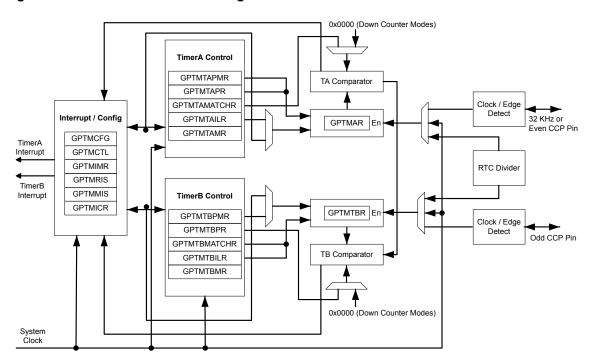


Figure 10-1. GPTM Module Block Diagram

Table 10-1. 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	-	-
Timer 3	TimerA	-	-
	TimerB	-	-

10.2 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 223), the **GPTM TimerA Mode (GPTMTAMR)** register (see page 224), and the **GPTM TimerB Mode (GPTMTBMR)** register (see page 226). 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.

10.2.1 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 (GPTMTAILR) register (see page 237) and the GPTM TimerB Interval Load (GPTMTBILR) register (see page 238). The prescale counters are initialized to 0x00: the GPTM TimerA Prescale (GPTMTAPR) register (see page 241) and the GPTM TimerB Prescale (GPTMTBPR) register (see page 242).

10.2.2 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 237
- GPTM TimerB Interval Load (GPTMTBILR) register [15:0], see page 238
- GPTM TimerA (GPTMTAR) register [15:0], see page 245
- GPTM TimerB (GPTMTBR) register [15:0], see page 246

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]

10.2.2.1 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 224), 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 228), 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 233), and holds it until it is cleared by writing the GPTM Interrupt Clear (GPTMICR) register (see page 235). If the time-out interrupt is enabled in the GPTM Interrupt Mask (GPTIMR) register (see page 231), the GPTM also sets the TATOMIS bit in the GPTM Masked Interrupt Status (GPTMMIS) register (see page 234). The trigger is enabled by setting the TAOTE bit in GPTMCTL, and can trigger SoC-level events such as ADC conversions.

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 asserted, the timer freezes counting until the signal is deasserted.

10.2.2.2 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 239) by the controller.

The input clock on the CCP0, CCP2, or CCP4 pins 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 inthe **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 **GPTIMR**, the GPTM also sets the RTCMIS bit in **GPTMISR** 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**.

10.2.3 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 223). 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.

10.2.3.1 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 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 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 **GPTIMR**, the GPTM also sets the TnTOMIS bit in **GPTMISR** and generates a controller interrupt. The trigger is enabled by setting the TnOTE bit in the **GPTMCTL** register, and can trigger SoC-level events such as ADC conversions.

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 enabled, the timer freezes counting until the signal is deasserted.

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).

Table 10-2. 16-Bit Timer With Prescaler Configurations

Prescale	#Clock (T c) ^a	Max Time	Units
00000000	1	1.3107	mS
00000001	2	2.6214	mS
00000010	3	3.9321	mS
11111100	254	332.9229	mS
11111110	255	334.2336	mS
11111111	256	335.5443	mS

a. Tc is the clock period.

10.2.3.2 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 \mathtt{TnEN} bit in the **GPTM Control (GPTMCTL)** register, the timer is enabled for event capture. Each input event on the \mathtt{CCP} pin decrements the counter by 1 until the event count matches **GPTMTnMATCHR**. When the counts match, the GPTM asserts the \mathtt{CnMRIS} bit in the **GPTMRIS** register (and the \mathtt{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 \mathtt{TnEN} bit in the **GPTMCTL** register. Once the event count has been reached, all further events are ignored until \mathtt{TnEN} is re-enabled by software.

Figure 10-2 on page 216 shows how input edge count mode works. In this case, the timer start value is set to **GPTMnILR** =0x000A and the match value is set to **GPTMnMATCHR** =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 \mathtt{TnEN} bit after the current count matches the value in the **GPTMnMR** register.

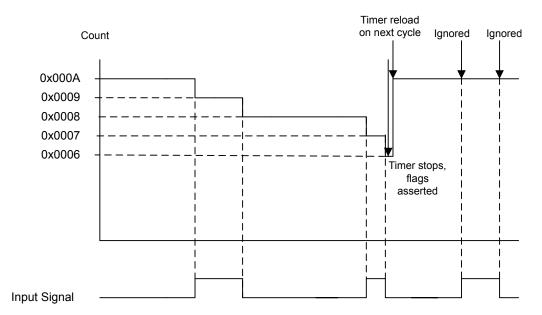


Figure 10-2. 16-Bit Input Edge Count Mode Example

10.2.3.3 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). This mode allows for event capture of either rising or falling edges, but not both. The timer is placed into Edge Time mode by setting the \mathtt{TnCMR} bit in the **GPTMTnMR** register, and the type of event that the timer captures is determined by the $\mathtt{TnEVENT}$ fields of the **GPTMCnTL** 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 **GPTMnILR** register.

Figure 10-3 on page 217 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**).

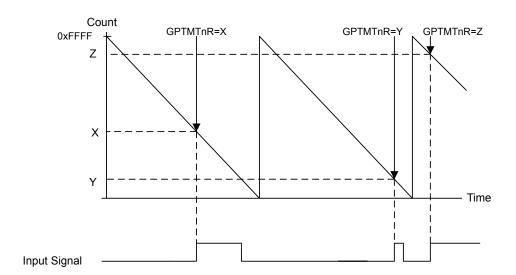


Figure 10-3. 16-Bit Input Edge Time Mode Example

10.2.3.4 16-Bit PWM Mode

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**. 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 (GPTMnMATCHR)**. Software has the capability of inverting the output PWM signal by setting the TnPWML bit in the **GPTMCTL** register.

Figure 10-4 on page 218 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 **GPTMnIRL**=0xC350 and the match value is **GPTMnMR**=0x411A.

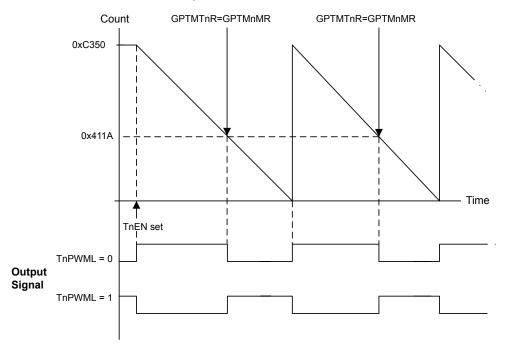


Figure 10-4. 16-Bit PWM Mode Example

10.3 Initialization and Configuration

To use the general-purpose timers, the peripheral clock must be enabled by setting the TIMERO, TIMER1, TIMER2, and TIMER3 bits in the **RCGC1** register.

This section shows module initialization and configuration examples for each of the supported timer modes.

10.3.1 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).
- 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 219. To re-enable the timer, repeat the sequence. A timer configured in Periodic mode does not stop counting after it times out.

10.3.2 32-Bit Real-Time Clock (RTC) Mode

To use the RTC mode, the timer must have a 32.768-KHz input signal on its CCP0, CCP2, or CCP4 pins. To enable the RTC feature, follow these steps:

- Ensure the timer is disabled (the TAEN bit is cleared) before making any changes.
- Write the GPTM Configuration Register (GPTMCFG) with a value of 0x1.
- Write the desired match value to the GPTM TimerA Match Register (GPTMTAMATCHR).
- Set/clear the RTCEN bit in the GPTM Control Register (GPTMCTL) as desired.
- 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 counter is re-loaded with 0x0000.0000 and begins counting. If an interrupt is enabled, it does not have to be cleared.

10.3.3 16-Bit One-Shot/Periodic Timer Mode

A timer is configured for 16-bit One-Shot and Periodic modes by the following sequence:

- 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.
- If a prescaler is to be used, write the prescale value to the GPTM Timern Prescale Register (GPTMTnPR).
- Load the start value into the GPTM Timer Interval Load Register (GPTMTnILR).
- If interrupts are required, set the TnTOIM bit in the GPTM Interrupt Mask Register (GPTMIMR).
- 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 219. To re-enable the timer, repeat the sequence. A timer configured in Periodic mode does not stop counting after it times out.

10.3.4 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.
- 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.
- 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 220 through step 9 on page 220.

10.3.5 16-Bit Input Edge Timing Mode

A timer is configured to Input Edge Timing mode by the following sequence:

- Ensure the timer is disabled (the Then bit is cleared) before making any changes.
- 2. Write the **GPTM Configuration (GPTMCFG)** register with a value of 0x4.
- 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.

10.3.6 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 TREVENT field of the GPTM Control (GPTMCTL) register.
- 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.

10.4 Register Map

Table 10-3 on page 221 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: 0x4003.1000

Timer2: 0x4003.2000

Timer3: 0x4003.3000

Table 10-3. Timers Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	GPTMCFG	R/W	0x0000.0000	GPTM Configuration	223
0x004	GPTMTAMR	R/W	0x0000.0000	GPTM TimerA Mode	224
800x0	GPTMTBMR	R/W	0x0000.0000	GPTM TimerB Mode	226
0x00C	GPTMCTL	R/W	0x0000.0000	GPTM Control	228

Offset	Name	Туре	Reset	Description	See page
0x018	GPTMIMR	R/W	0x0000.0000	GPTM Interrupt Mask	231
0x01C	GPTMRIS	RO	0x0000.0000	GPTM Raw Interrupt Status	233
0x020	GPTMMIS	RO	0x0000.0000	GPTM Masked Interrupt Status	234
0x024	GPTMICR	W1C	0x0000.0000	GPTM Interrupt Clear	235
0x028	GPTMTAILR	R/W	0x0000.FFFF (16-bit mode) 0xFFFF.FFF (32-bit mode)	GPTM TimerA Interval Load	237
0x02C	GPTMTBILR	R/W	0x0000.FFFF	GPTM TimerB Interval Load	238
0x030	GPTMTAMATCHR	R/W	0x0000.FFFF (16-bit mode) 0xFFFF.FFF (32-bit mode)	GPTM TimerA Match	239
0x034	GPTMTBMATCHR	R/W	0x0000.FFFF	GPTM TimerB Match	240
0x038	GPTMTAPR	R/W	0x0000.0000	GPTM TimerA Prescale	241
0x03C	GPTMTBPR	R/W	0x0000.0000	GPTM TimerB Prescale	242
0x040	GPTMTAPMR	R/W	0x0000.0000	GPTM TimerA Prescale Match	243
0x044	GPTMTBPMR	R/W	0x0000.0000	GPTM TimerB Prescale Match	244
0x048	GPTMTAR	RO	0x0000.FFFF (16-bit mode) 0xFFFF.FFF (32-bit mode)	GPTM TimerA	245
0x04C	GPTMTBR	RO	0x0000.FFFF	GPTM TimerB	246

10.5 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

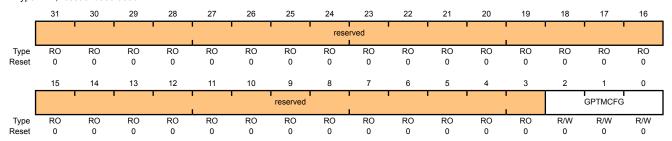
This register configures the global operation of the GPTM module. The value written to this register determines whether the GPTM is in 32- or 16-bit mode.

GPTM Configuration (GPTMCFG)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

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

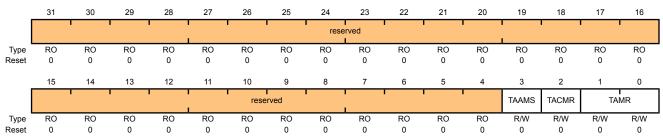
This register configures the GPTM based on the configuration selected in the **GPTMCFG** register. When in 16-bit PWM mode, set the TAAMS bit to 0x1, the TACMR bit to 0x0, and the TAMR field to 0x2.

GPTM TimerA Mode (GPTMTAMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x004

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	TAAMS	R/W	0	GPTM TimerA Alternate Mode Select

The TAAMS values are defined as follows:

Value Description

Capture mode is enabled.

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

The TACMR values are defined as follows:

Value Description

0 Edge-Count mode

1 Edge-Time mode

Bit/Field	Name	Туре	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

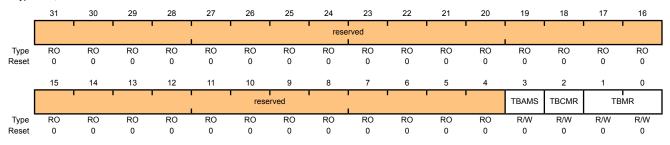
This register configures the GPTM based on the configuration selected in the **GPTMCFG** register. When in 16-bit PWM mode, set the TBAMS bit to 0x1, the TBCMR bit to 0x0, and the TBMR field to 0x2.

GPTM TimerB Mode (GPTMTBMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x008

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	TBAMS	R/W	0	GPTM TimerB Alternate Mode Select

The TBAMS values are defined as follows:

Value Description

Capture mode is enabled.

PWM mode is enabled.

Note: To enable PWM mode, you must also clear the TBCMR bit and set the TBMR field to 0x2.

2 TBCMR R/W 0 GPTM TimerB Capture Mode

The TBCMR values are defined as follows:

Value Description

0 Edge-Count mode

1 Edge-Time mode

Bit/Field	Name	Туре	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 $\ensuremath{\mathbf{GPTMTAMR}}$ is used.

Register 4: GPTM Control (GPTMCTL), offset 0x00C

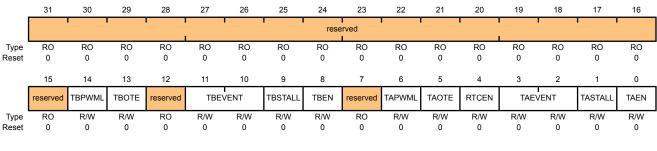
This register is used alongside the **GPTMCFG** and **GMTMTnMR** registers to fine-tune the timer configuration, and to enable other features such as timer stall and the output trigger. The output trigger can be used to initiate transfers on the ADC module.

GPTM Control (GPTMCTL)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x00C

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:15	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.
14	TBPWML	R/W	0	GPTM TimerB PWM Output Level The TBPWML values are defined as follows: Value Description 0 Output is unaffected. 1 Output is inverted.
13	ТВОТЕ	R/W	0	GPTM TimerB Output Trigger Enable The TBOTE values are defined as follows: Value Description 0 The output TimerB trigger is disabled. 1 The output TimerB trigger 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.

RO

0

reserved

12

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 TimerB Stall Enable
				The TBSTALL values are defined as follows:
				Value Description
				0 TimerB stalling is disabled.
				1 TimerB stalling is enabled.
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 trigger is disabled.
				1 The output TimerA trigger is enabled.

July 25, 2008 229

Bit/Field	Name	Type	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 TimerA Stall Enable		
				The TASTALL values are defined as follows:		
				Value Description		
				0 TimerA stalling is disabled.		
				1 TimerA stalling is enabled.		
0	TAEN	R/W	0	GPTM TimerA Enable		
				The TAEN values are defined as follows:		
				Value Description		
				0 TimerA is disabled.		
				1 TimerA is enabled and begins counting or the capture logic is enabled based on the GPTMCFG register.		

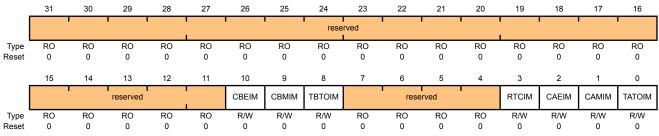
Register 5: GPTM Interrupt Mask (GPTMIMR), offset 0x018

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
Timer3 base: 0x4003.3000
Offset 0x018

Type R/W, 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	CBEIM	R/W	0	GPTM CaptureB Event Interrupt Mask
				The CBEIM values are defined as follows:
				Value Description
				0 Interrupt is disabled.
				1 Interrupt is enabled.
9	СВМІМ	R/W	0	GPTM CaptureB Match Interrupt Mask
				The CBMIM values are defined as follows:
				Value Description
				0 Interrupt is disabled.
				1 Interrupt is enabled.
8	TBTOIM	R/W	0	GPTM TimerB Time-Out Interrupt Mask
				The TBTOIM values are defined as follows:
				Value Description
				0 Interrupt is disabled.
				1 Interrupt is enabled.
7:4	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.

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

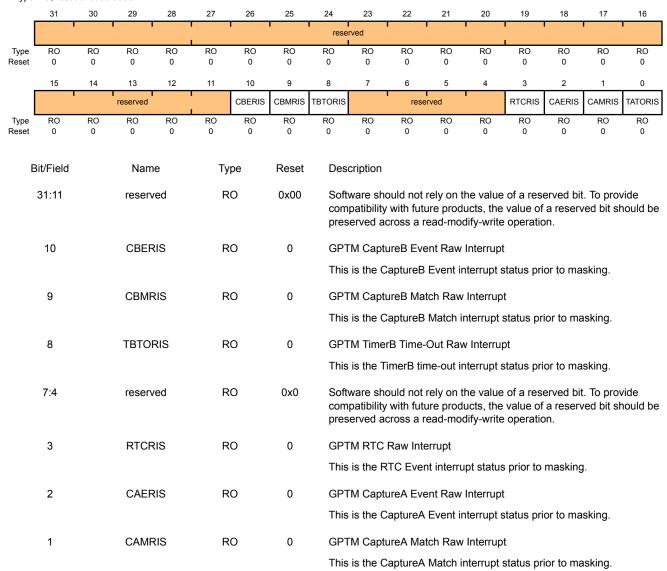
This register shows the state of the GPTM's internal interrupt signal. These bits are set whether or not the interrupt is masked in the **GPTMIMR** register. Each bit can be cleared by writing a 1 to its corresponding bit in **GPTMICR**.

GPTM Raw Interrupt Status (GPTMRIS)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x01C

Type RO, reset 0x0000.0000



GPTM TimerA Time-Out Raw Interrupt

This the TimerA time-out interrupt status prior to masking.

0

TATORIS

RO

0

Register 7: GPTM Masked Interrupt Status (GPTMMIS), offset 0x020

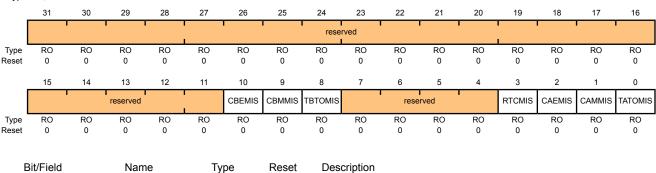
This register show the state of the GPTM's controller-level interrupt. If an interrupt is unmasked in **GPTMIMR**, and there is an event that causes the interrupt to be asserted, the corresponding bit is set in this register. All bits are cleared by writing a 1 to the corresponding bit in **GPTMICR**.

GPTM Masked Interrupt Status (GPTMMIS)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

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

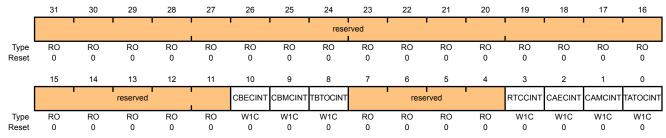
This register is used to clear the status bits in the **GPTMRIS** and **GPTMMIS** registers. Writing a 1 to a bit clears the corresponding bit in the **GPTMRIS** and **GPTMMIS** registers.

GPTM Interrupt Clear (GPTMICR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x024

Type W1C, 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	CBECINT	W1C	0	GPTM CaptureB Event Interrupt Clear
				The CBECINT values are defined as follows:
				Value Description
				0 The interrupt is unaffected.
				1 The interrupt is cleared.
9	CBMCINT	W1C	0	GPTM CaptureB Match Interrupt Clear
				The CBMCINT values are defined as follows:
				Value Description
				0 The interrupt is unaffected.
				1 The interrupt is cleared.
8	TBTOCINT	W1C	0	GPTM TimerB Time-Out Interrupt Clear
				The TBTOCINT values are defined as follows:
				Value Description
				0 The interrupt is unaffected.
				1 The interrupt is cleared.
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.

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. 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 Raw Interrupt This is the CaptureA match interrupt status after masking.
0	TATOCINT	W1C	0	GPTM TimerA Time-Out Raw Interrupt The TATOCINT values are defined as follows:
				Value Description O 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 Timer3 base: 0x4003.3000

Offset 0x028

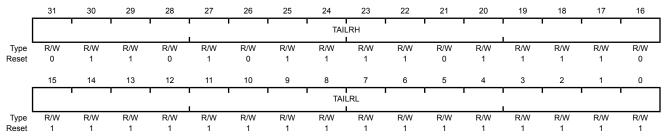
Bit/Field

15:0

Type R/W, reset 0x0000.FFFF (16-bit mode) and 0xFFFF.FFFF (32-bit mode)

Name

TAILRL



Reset

0xFFFF

Type

R/W

31:16	TAILRH	R/W	0xFFFF (32-bit mode) 0x0000 (16-bit mode)	When configured for 32-bit mode via the GPTMCFG register, the GPTM Timer P. Interval Lead (GPTMTPILE) register leads this value on a
				In 16-bit mode, this field reads as 0 and does not have an effect on the state of GPTMTBILR .

Description

For both 16- and 32-bit modes, writing this field loads the counter for TimerA. A read returns the current value of **GPTMTAILR**.

GPTM TimerA Interval Load Register Low

Register 10: GPTM TimerB Interval Load (GPTMTBILR), offset 0x02C

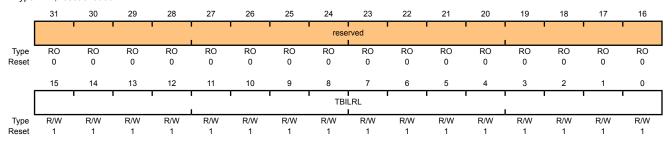
This register is used to load the starting count value into TimerB. When the GPTM is configured to a 32-bit mode, **GPTMTBILR** returns the current value of TimerB and ignores writes.

GPTM TimerB Interval Load (GPTMTBILR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

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

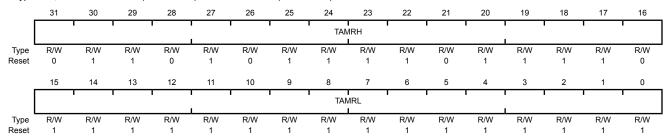
This register is used in 32-bit Real-Time Clock mode and 16-bit PWM and Input Edge Count modes.

GPTM TimerA Match (GPTMTAMATCHR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x030

Type R/W, reset 0x0000.FFFF (16-bit mode) and 0xFFFF.FFFF (32-bit mode)



Bit/Field	Name	Type	Reset	Description
31:16	TAMRH	R/W	0xFFFF (32-bit mode) 0x0000 (16-bit mode)	When configured for 32-bit Real-Time Clock (RTC) mode via the
				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

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)

TBMRL

R/W

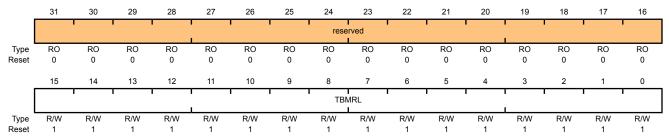
0xFFFF

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x034

15:0

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.

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

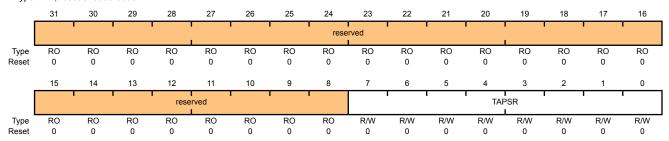
This register allows software to extend the range of the 16-bit timers when operating in one-shot or periodic mode.

GPTM TimerA Prescale (GPTMTAPR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

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 10-2 on page 215 for more details and an example.

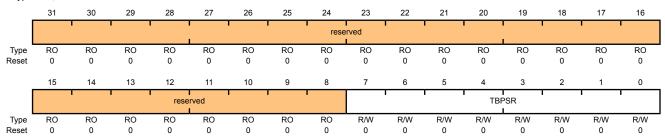
Register 14: GPTM TimerB Prescale (GPTMTBPR), offset 0x03C

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
Timer3 base: 0x4003.3000
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 10-2 on page 215 for more details and an example.

Register 15: GPTM TimerA Prescale Match (GPTMTAPMR), offset 0x040

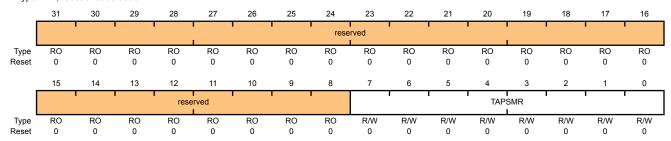
This register effectively extends the range of **GPTMTAMATCHR** to 24 bits when operating in 16-bit one-shot or periodic mode.

GPTM TimerA Prescale Match (GPTMTAPMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x040

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	TAPSMR	R/W	0x00	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

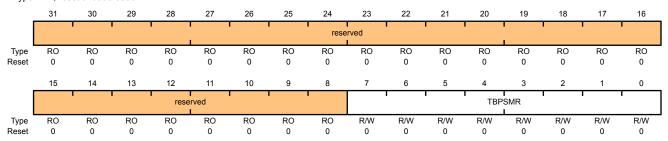
This register effectively extends the range of **GPTMTBMATCHR** to 24 bits when operating in 16-bit one-shot or periodic mode.

GPTM TimerB Prescale Match (GPTMTBPMR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x044

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	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

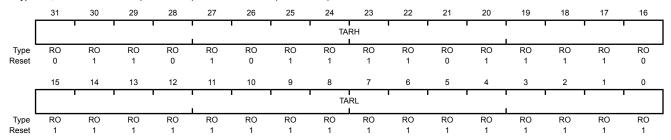
This register shows the current value of the TimerA counter in all cases except for Input Edge Count mode. When in this mode, this register contains the time at which the last edge event took place.

GPTM TimerA (GPTMTAR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

Offset 0x048

Type RO, reset 0x0000.FFFF (16-bit mode) and 0xFFFF.FFFF (32-bit mode)



Bit/Field	Name	Type	Reset	Description
31:16	TARH		(32-bit mode) 0x0000	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 timestamp from the last edge event.

Register 18: GPTM TimerB (GPTMTBR), offset 0x04C

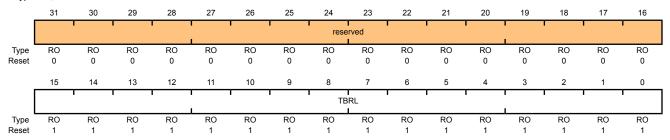
This register shows the current value of the TimerB counter in all cases except for Input Edge Count mode. When in this mode, this register contains the time at which the last edge event took place.

GPTM TimerB (GPTMTBR)

Timer0 base: 0x4003.0000 Timer1 base: 0x4003.1000 Timer2 base: 0x4003.2000 Timer3 base: 0x4003.3000

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 timestamp from the last edge event.

11 Watchdog Timer

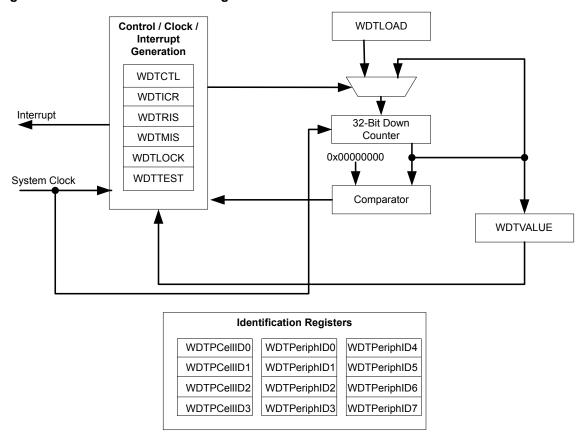
A watchdog timer can generate nonmaskable interrupts (NMIs) or a reset when a time-out value is reached. The watchdog timer is used to regain control when a system has failed due to a software error or due to the failure of an external device to respond in the expected way.

The Stellaris[®] Watchdog Timer module consists of a 32-bit down counter, a programmable load register, interrupt generation logic, a locking register, and user-enabled stalling.

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.

11.1 Block Diagram

Figure 11-1. WDT Module Block Diagram



11.2 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.

11.3 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.
- If the Watchdog is configured to trigger system resets, set the RESEN bit in the WDTCTL register.
- 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.

11.4 Register Map

Table 11-1 on page 248 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 11-1. Watchdog Timer Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	WDTLOAD	R/W	0xFFFF.FFFF	Watchdog Load	250
0x004	WDTVALUE	RO	0xFFFF.FFFF	Watchdog Value	251
800x0	WDTCTL	R/W	0x0000.0000	Watchdog Control	252
0x00C	WDTICR	WO	-	Watchdog Interrupt Clear	253
0x010	WDTRIS	RO	0x0000.0000	Watchdog Raw Interrupt Status	254
0x014	WDTMIS	RO	0x0000.0000	Watchdog Masked Interrupt Status	255
0x418	WDTTEST	R/W	0x0000.0000	Watchdog Test	256
0xC00	WDTLOCK	R/W	0x0000.0000	Watchdog Lock	257

Offset	Name	Туре	Reset	Description	See page
0xFD0	WDTPeriphID4	RO	0x0000.0000	Watchdog Peripheral Identification 4	258
0xFD4	WDTPeriphID5	RO	0x0000.0000	Watchdog Peripheral Identification 5	259
0xFD8	WDTPeriphID6	RO	0x0000.0000	Watchdog Peripheral Identification 6	260
0xFDC	WDTPeriphID7	RO	0x0000.0000	Watchdog Peripheral Identification 7	261
0xFE0	WDTPeriphID0	RO	0x0000.0005	Watchdog Peripheral Identification 0	262
0xFE4	WDTPeriphID1	RO	0x0000.0018	Watchdog Peripheral Identification 1	263
0xFE8	WDTPeriphID2	RO	0x0000.0018	Watchdog Peripheral Identification 2	264
0xFEC	WDTPeriphID3	RO	0x0000.0001	Watchdog Peripheral Identification 3	265
0xFF0	WDTPCellID0	RO	0x0000.000D	Watchdog PrimeCell Identification 0	266
0xFF4	WDTPCellID1	RO	0x0000.00F0	Watchdog PrimeCell Identification 1	267
0xFF8	WDTPCellID2	RO	0x0000.0005	Watchdog PrimeCell Identification 2	268
0xFFC	WDTPCellID3	RO	0x0000.00B1	Watchdog PrimeCell Identification 3	269

11.5 Register Descriptions

The remainder of this section lists and describes the WDT registers, in numerical order by address offset.

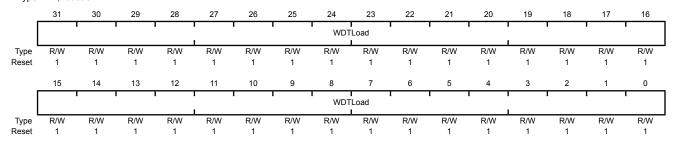
Register 1: Watchdog Load (WDTLOAD), offset 0x000

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 Type Reset Description

31:0 WDTLoad R/W 0xFFF.FFFF Watchdog Load Value

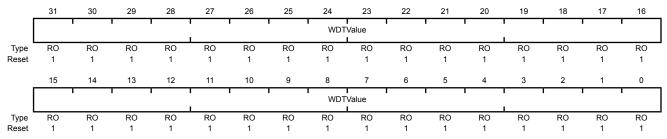
Register 2: Watchdog Value (WDTVALUE), offset 0x004

This register contains the current count value of the timer.

Watchdog Value (WDTVALUE)

Base 0x4000.0000 Offset 0x004

Type RO, reset 0xFFFF.FFFF



Bit/Field Name Type Reset Description

31:0 WDTValue RO 0xFFF.FFFF Watchdog Value

Current value of the 32-bit down counter.

Register 3: Watchdog Control (WDTCTL), offset 0x008

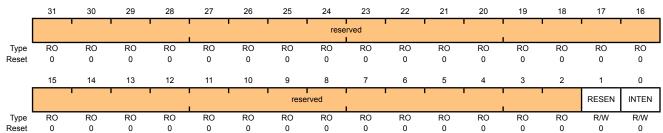
This register is the watchdog control register. The watchdog timer can be configured to generate a reset signal (on second time-out) or an interrupt on time-out.

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

Value Description

The INTEN values are defined as follows:

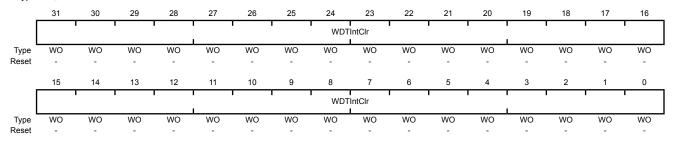
- 0 Interrupt event disabled (once this bit is set, it can only be cleared by a hardware reset).
- 1 Interrupt event enabled. Once enabled, all writes are ignored.

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 -



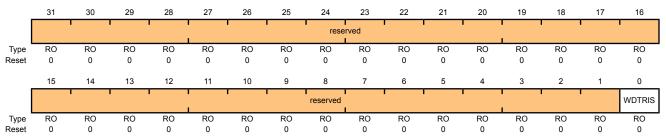
Bit/Field	Name	Type	Reset	Description
31:0	WDTIntClr	WO	-	Watchdog Interrupt Clear

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

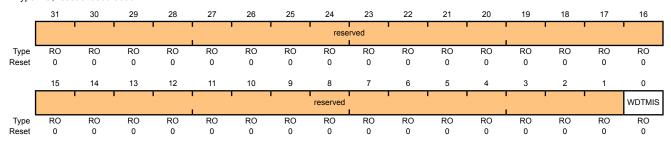
Gives the raw interrupt state (prior to masking) of WDTINTR.

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

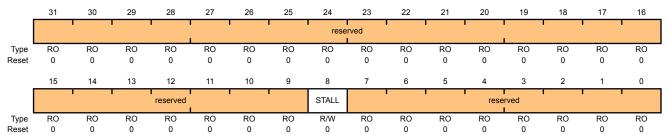
Gives the masked interrupt state (after masking) of the WDTINTR interrupt.

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.

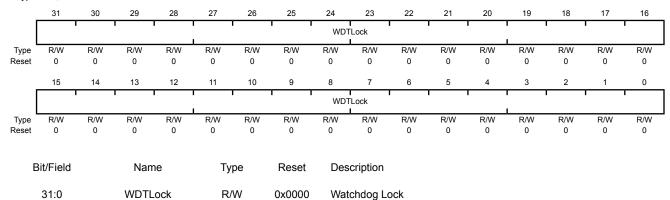
Register 8: Watchdog Lock (WDTLOCK), offset 0xC00

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:

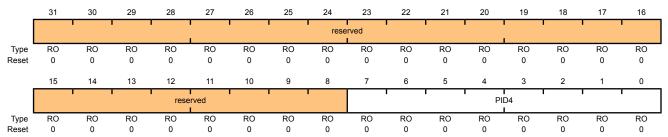
Value Description
0x0000.0001 Locked
0x0000.0000 Unlocked

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]

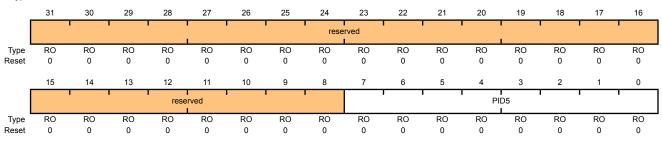
Register 10: Watchdog Peripheral Identification 5 (WDTPeriphID5), offset 0xFD4

The **WDTPeriphIDn** registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 5 (WDTPeriphID5)

Base 0x4000.0000

Offset 0xFD4
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	PID5	RO	0x00	WDT Peripheral ID Register[15:8]

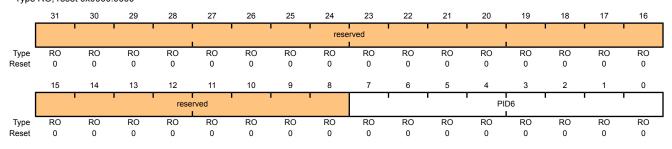
Register 11: Watchdog Peripheral Identification 6 (WDTPeriphID6), offset 0xFD8

The **WDTPeriphIDn** registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 6 (WDTPeriphID6)

Base 0x4000.0000

Offset 0xFD8
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	PID6	RO	0x00	WDT Peripheral ID Register[23:16]

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)

Base 0x4000.0000

Offset 0xFDC 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	PID7	RO	0x00	WDT Peripheral ID Register[31:24]

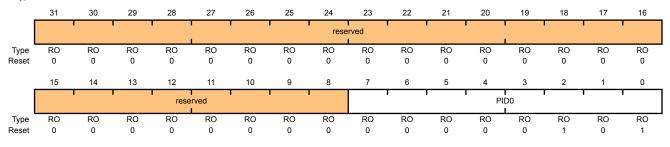
Register 13: Watchdog Peripheral Identification 0 (WDTPeriphID0), offset 0xFE0

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 0 (WDTPeriphID0)

Base 0x4000.0000

Offset 0xFE0
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	PID0	RO	0x05	Watchdog Peripheral ID Register[7:0]

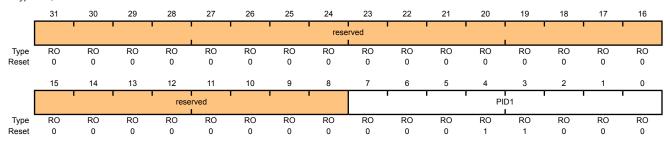
Register 14: Watchdog Peripheral Identification 1 (WDTPeriphID1), offset 0xFE4

The **WDTPeriphIDn** registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 1 (WDTPeriphID1)

Base 0x4000.0000

Offset 0xFE4
Type RO, reset 0x0000.0018



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	0x18	Watchdog Peripheral ID Register[15:8]

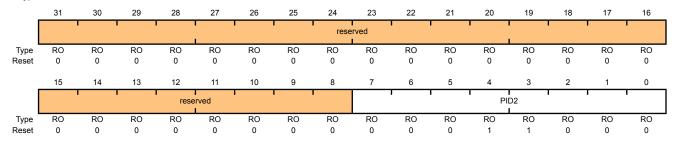
Register 15: Watchdog Peripheral Identification 2 (WDTPeriphID2), offset 0xFE8

The **WDTPeriphIDn** registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 2 (WDTPeriphID2)

Base 0x4000.0000

Offset 0xFE8
Type RO, reset 0x0000.0018



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	PID2	RO	0x18	Watchdog Peripheral ID Register[23:16]

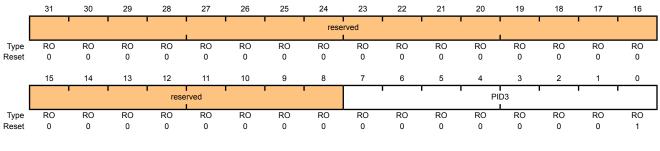
Register 16: Watchdog Peripheral Identification 3 (WDTPeriphID3), offset 0xFEC

The WDTPeriphIDn registers are hard-coded and the fields within the register determine the reset value.

Watchdog Peripheral Identification 3 (WDTPeriphID3)

Base 0x4000.0000

Offset 0xFEC
Type RO, reset 0x0000.0001



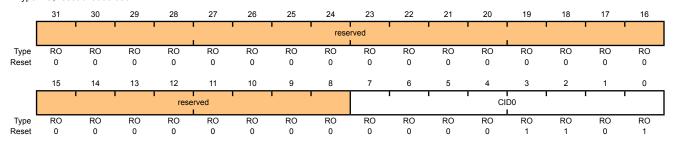
Bit/Field	Name	Туре	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	Watchdog Peripheral ID Register[31:24]

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



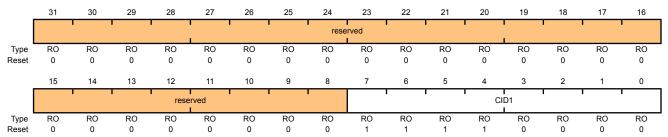
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	Watchdog PrimeCell ID Register[7:0]

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



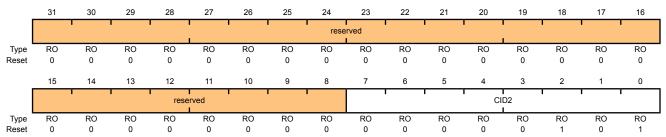
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	Watchdog PrimeCell ID Register[15:8]

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



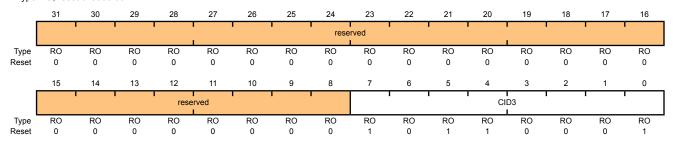
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	Watchdog PrimeCell ID Register[23:16]

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]

12 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.

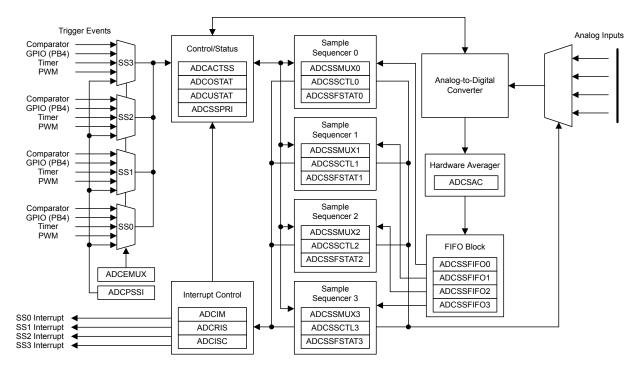
The Stellaris[®] ADC module features 10-bit conversion resolution and supports four input channels, plus an internal temperature sensor. The ADC module contains a programmable sequencer which allows for the sampling of multiple analog input sources without controller intervention. Each sample sequence provides flexible programming with fully configurable input source, trigger events, interrupt generation, and sequence priority.

The Stellaris® ADC provides the following features:

- Four analog input channels
- Single-ended and differential-input configurations
- Internal temperature sensor
- Sample rate of one million samples/second
- Four programmable sample conversion sequences from one to eight entries long, with corresponding conversion result FIFOs
- Flexible trigger control
 - Controller (software)
 - Timers
 - Analog Comparators
 - GPIO
- Hardware averaging of up to 64 samples for improved accuracy
- An internal 3-V reference is used by the converter.
- Power and ground for the analog circuitry is separate from the digital power and ground.

12.1 Block Diagram

Figure 12-1. ADC Module Block Diagram



12.2 Functional Description

The Stellaris[®] ADC collects sample data by using a programmable sequence-based approach instead of the traditional single or double-sampling approach 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.

12.2.1 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 12-1 on page 271 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 12-1. Samples and FIFO Depth of Sequencers

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, but can 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 Interrupt Enable (IE) bits can be set for any combination of samples, allowing interrupts to be generated after every sample in the sequence if necessary. Also, the END bit can be set at any point within a sample sequence. For example, if Sequencer 0 is used, the END 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.

12.2.2 Module Control

Outside of the Sample Sequencers, the remainder of the control logic is responsible for tasks such as interrupt generation, sequence prioritization, and 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.

12.2.2.1 Interrupts

The Sample Sequencers dictate the events that cause interrupts, but they don't have control over whether the interrupt is actually sent to the interrupt controller. The ADC module's interrupt signal is 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 a Sample Sequencer's interrupt signal, and the ADC Interrupt Status and Clear (ADCISC) register, which shows the logical AND of the ADCRIS register's INR bit and the ADCIM register's MASK bits. Interrupts are cleared by writing a 1 to the corresponding IN bit in ADCISC.

12.2.2.2 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.

12.2.2.3 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 CH bits in the ADC Processor Sample Sequence Initiate (ADCPSSI) register.

When using the "Always" trigger, care must be taken. If a sequence's priority is too high, it is possible to starve other lower priority sequences.

12.2.3 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 289). There is a single averaging circuit and all input channels receive the same amount of averaging whether they are single-ended or differential.

12.2.4 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.

12.2.5 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 **D** bit (in the **ADCSSCTL0** register) in a step's configuration nibble.

When a sequence step is configured for differential sampling, its corresponding value in the **ADCSSMUX** 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 12-2 on page 273). 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 12-2 on page 273).

Table 12-2. 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:

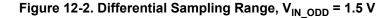
 ΔV (differential voltage) = V_{IN} (even channels) – V_{IN} (odd channels), therefore:

- 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 12-2 on page 274 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 12-3 on page 274 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 12-4 on page 275 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.



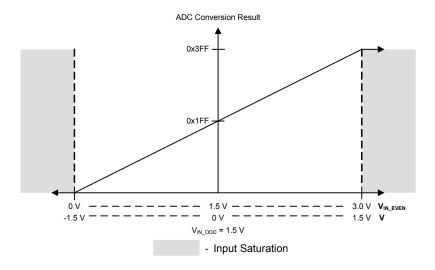
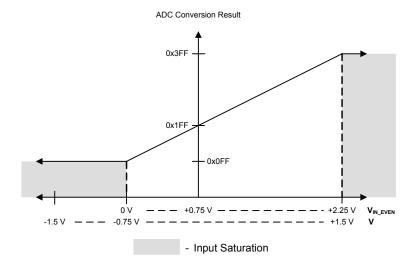


Figure 12-3. Differential Sampling Range, $V_{IN\ ODD} = 0.75 \text{ V}$



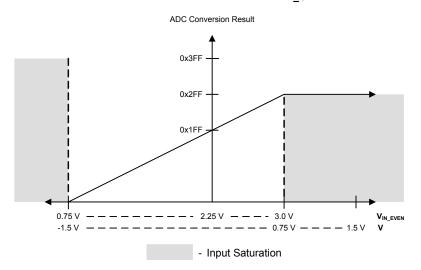


Figure 12-4. Differential Sampling Range, $V_{IN\ ODD}$ = 2.25 V

12.2.6 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 302).

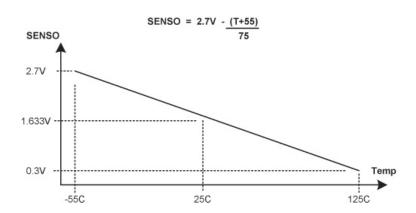
12.2.7 Internal Temperature Sensor

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 12-5 on page 275.

Figure 12-5. Internal Temperature Sensor Characteristic



12.3 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.

12.3.1 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:

- Enable the ADC clock by writing a value of 0x0001.0000 to the RCGC0 register (see page 99).
- 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.

12.3.2 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:

- Ensure that the Sample Sequencer is disabled by writing a 0 to the corresponding ASEN 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.
- 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.
- 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 ASEN bit in the ADCACTSS register.

12.4 Register Map

Table 12-3 on page 276 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.

Table 12-3. ADC Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	ADCACTSS	R/W	0x0000.0000	ADC Active Sample Sequencer	278

Offset	Name	Туре	Reset	Description	See page
0x004	ADCRIS	RO	0x0000.0000	ADC Raw Interrupt Status	279
800x0	ADCIM	R/W	0x0000.0000	ADC Interrupt Mask	280
0x00C	ADCISC	R/W1C	0x0000.0000	ADC Interrupt Status and Clear	281
0x010	ADCOSTAT	R/W1C	0x0000.0000	ADC Overflow Status	282
0x014	ADCEMUX	R/W	0x0000.0000	ADC Event Multiplexer Select	283
0x018	ADCUSTAT	R/W1C	0x0000.0000	ADC Underflow Status	286
0x020	ADCSSPRI	R/W	0x0000.3210	ADC Sample Sequencer Priority	287
0x028	ADCPSSI	WO	-	ADC Processor Sample Sequence Initiate	288
0x030	ADCSAC	R/W	0x0000.0000	ADC Sample Averaging Control	289
0x040	ADCSSMUX0	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 0	290
0x044	ADCSSCTL0	R/W	0x0000.0000	ADC Sample Sequence Control 0	292
0x048	ADCSSFIFO0	RO	0x0000.0000	ADC Sample Sequence Result FIFO 0	295
0x04C	ADCSSFSTAT0	RO	0x0000.0100	ADC Sample Sequence FIFO 0 Status	296
0x060	ADCSSMUX1	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 1	297
0x064	ADCSSCTL1	R/W	0x0000.0000	ADC Sample Sequence Control 1	298
0x068	ADCSSFIFO1	RO	0x0000.0000	ADC Sample Sequence Result FIFO 1	295
0x06C	ADCSSFSTAT1	RO	0x0000.0100	ADC Sample Sequence FIFO 1 Status	296
0x080	ADCSSMUX2	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 2	297
0x084	ADCSSCTL2	R/W	0x0000.0000	ADC Sample Sequence Control 2	298
0x088	ADCSSFIFO2	RO	0x0000.0000	ADC Sample Sequence Result FIFO 2	295
0x08C	ADCSSFSTAT2	RO	0x0000.0100	ADC Sample Sequence FIFO 2 Status	296
0x0A0	ADCSSMUX3	R/W	0x0000.0000	ADC Sample Sequence Input Multiplexer Select 3	300
0x0A4	ADCSSCTL3	R/W	0x0000.0002	ADC Sample Sequence Control 3	301
0x0A8	ADCSSFIFO3	RO	0x0000.0000	ADC Sample Sequence Result FIFO 3	295
0x0AC	ADCSSFSTAT3	RO	0x0000.0100	ADC Sample Sequence FIFO 3 Status	296
0x100	ADCTMLB	R/W	0x0000.0000	ADC Test Mode Loopback	302

12.5 Register Descriptions

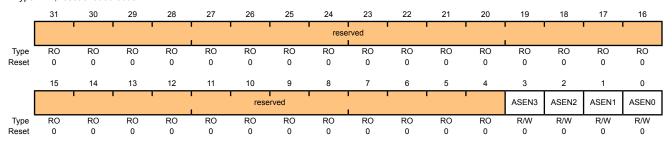
The remainder of this section lists and describes the ADC registers, in numerical order by address offset.

Register 1: ADC Active Sample Sequencer (ADCACTSS), offset 0x000

This register controls the activation of the Sample Sequencers. Each Sample Sequencer can be enabled/disabled independently.

ADC Active Sample Sequencer (ADCACTSS)

Base 0x4003.8000 Offset 0x000 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	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	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.

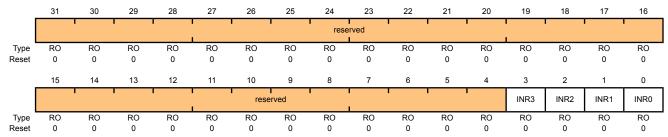
Register 2: ADC Raw Interrupt Status (ADCRIS), offset 0x004

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	Type	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	INR3	RO	0	SS3 Raw Interrupt Status
				Set by hardware when a sample with its respective ADCSSCTL3 IE bit has completed conversion. This bit is cleared by writing a 1 to the ADCISC IN3 bit.
2	INR2	RO	0	SS2 Raw Interrupt Status
				Set by hardware when a sample with its respective ADCSSCTL2 IE bit has completed conversion. This bit is cleared by writing a 1 to the ADCISC IN2 bit.
1	INR1	RO	0	SS1 Raw Interrupt Status
				Set by hardware when a sample with its respective ADCSSCTL1 IE bit has completed conversion. This bit is cleared by writing a 1 to the ADCISC IN1 bit.
0	INR0	RO	0	SS0 Raw Interrupt Status
				Set by hardware when a sample with its respective ADCSSCTI 0 TE bit

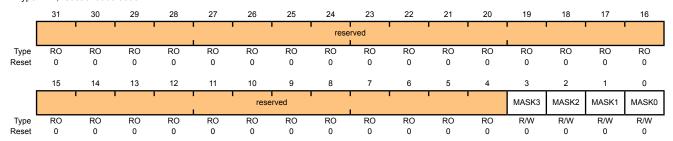
Set by hardware when a sample with its respective ADCSSCTL0 IE bit has completed conversion. This bit is cleared by writing a 1 to the ADCISC INO bit.

Register 3: ADC Interrupt Mask (ADCIM), offset 0x008

This register controls whether the Sample Sequencer raw interrupt signals are promoted to controller interrupts. The raw interrupt signal for each Sample Sequencer can be masked independently.

ADC Interrupt Mask (ADCIM)

Base 0x4003.8000 Offset 0x008 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	MASK3	R/W	0	SS3 Interrupt Mask
				Specifies whether the raw interrupt signal from Sample Sequencer 3 (ADCRIS register INR3 bit) is promoted to a controller interrupt. If set, the raw interrupt signal is promoted to a controller interrupt. Otherwise, it is not.
2	MASK2	R/W	0	SS2 Interrupt Mask
				Specifies whether the raw interrupt signal from Sample Sequencer 2 (ADCRIS register INR2 bit) is promoted to a controller interrupt. If set, the raw interrupt signal is promoted to a controller interrupt. Otherwise, it is not.
1	MASK1	R/W	0	SS1 Interrupt Mask
				Specifies whether the raw interrupt signal from Sample Sequencer 1 (ADCRIS register INR1 bit) is promoted to a controller interrupt. If set, the raw interrupt signal is promoted to a controller interrupt. Otherwise, it is not.
0	MASK0	R/W	0	SS0 Interrupt Mask

Specifies whether the raw interrupt signal from Sample Sequencer 0 (ADCRIS register INRO bit) is promoted to a controller interrupt. If set, the raw interrupt signal is promoted to a controller interrupt. Otherwise, it is not.

Register 4: ADC Interrupt Status and Clear (ADCISC), offset 0x00C

This register provides the mechanism for clearing 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. Interrupts are cleared by writing a 1 to the corresponding bit position. If software is polling the **ADCRIS** instead of generating interrupts, the 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

_	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										IN3	IN2	IN1	IN0		
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Type	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	IN3	R/W1C	0	SS3 Interrupt Status and Clear
				This bit is set by hardware when the MASK3 and INR3 bits are both 1, providing a level-based interrupt to the controller. It is cleared by writing a 1, and also clears the INR3 bit.
2	IN2	R/W1C	0	SS2 Interrupt Status and Clear
				This bit is set by hardware when the MASK2 and INR2 bits are both 1, providing a level based interrupt to the controller. It is cleared by writing a 1, and also clears the INR2 bit.
1	IN1	R/W1C	0	SS1 Interrupt Status and Clear
				This bit is set by hardware when the MASK1 and INR1 bits are both 1, providing a level based interrupt to the controller. It is cleared by writing a 1, and also clears the INR1 bit.
0	IN0	R/W1C	0	SS0 Interrupt Status and Clear

This bit is set by hardware when the MASK0 and INR0 bits are both 1, providing a level based interrupt to the controller. It is cleared by writing a 1, and also clears the INR0 bit.

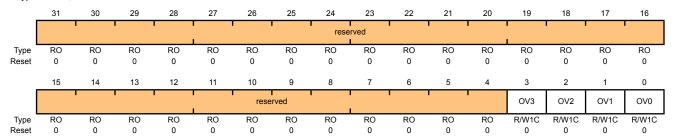
Register 5: ADC Overflow Status (ADCOSTAT), offset 0x010

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	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	OV3	R/W1C	0	SS3 FIFO Overflow
				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 and this bit is set by hardware to indicate the occurrence of dropped data. This bit is cleared by writing a 1.
2	OV2	R/W1C	0	SS2 FIFO Overflow
				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 and this bit is set by hardware to indicate the occurrence of dropped data. This bit is cleared by writing a 1.
1	OV1	R/W1C	0	SS1 FIFO Overflow
				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 and this bit is set by hardware to indicate the occurrence of dropped data. This bit is cleared by writing a 1.
0	OV0	R/W1C	0	SS0 FIFO Overflow
				This hit are alifes that the FIFO for Commis Commence O has hit are

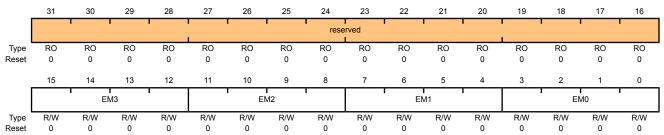
This bit specifies that the FIFO for Sample Sequencer 0 has hit an overflow condition where the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped and this bit is set by hardware to indicate the occurrence of dropped data. This bit is cleared by writing 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	Type	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.
15:12	EM3	R/W	0x00	SS3 Trigger Select

This field selects the trigger source for Sample Sequencer 3.

The valid configurations for this field are:

Value	Event
0x0	Controller (default)
0x1	Analog Comparator 0
0x2	Analog Comparator 1
0x3	Analog Comparator 2
0x4	External (GPIO PB4)
0x5	Timer
0x6	Reserved
0x7	Reserved
8x0	Reserved
0x9-0xE	reserved
0xF	Always (continuously sample)

Bit/Field	Name	Type	Reset	Description	
11:8	EM2	R/W	0x00	SS2 Trigger Select	
				This field selects the trigger s	source for Sample Sequencer 2.
				The valid configurations for the	his field are:
				Value Event	
				0x0 Controller (default)	
				0x1 Analog Comparato	r 0
				0x2 Analog Comparato	r 1
				0x3 Analog Comparato	r 2
				0x4 External (GPIO PB	44)
				0x5 Timer	
				0x6 Reserved	
				0x7 Reserved	
				0x8 Reserved	
				0x9-0xE reserved	
				0xF Always (continuous	sly sample)
7:4	EM1	R/W	0x00	SS1 Trigger Select	
				This field selects the trigger s	source for Sample Sequencer 1.
				The valid configurations for the	his field are:
				Value Event	
				0x0 Controller (default)	
				0x1 Analog Comparato	r 0
				0x2 Analog Comparato	r 1
				0x3 Analog Comparato	r 2
				0x4 External (GPIO PB	44)
				0x5 Timer	
				0x6 Reserved	
				0x7 Reserved	
				0x8 Reserved	
				0x9-0xE reserved	

Bit/Field	Name	Туре	Reset	Descripti	on
3:0	EM0	R/W	0x00	SS0 Trig	ger Select
				This field	selects the trigger source for Sample Sequencer 0.
				The valid	configurations for this field are:
				Value	Event
				0x0	Controller (default)
				0x1	Analog Comparator 0
				0x2	Analog Comparator 1
				0x3	Analog Comparator 2
				0x4	External (GPIO PB4)
				0x5	Timer
				0x6	Reserved
				0x7	Reserved
				8x0	Reserved
				0x9-0xE	reserved
				0xF	Always (continuously sample)

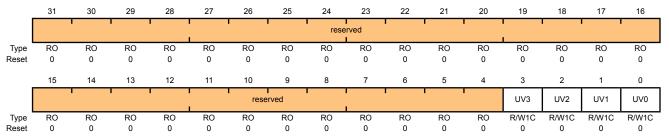
Register 7: ADC Underflow Status (ADCUSTAT), offset 0x018

This register indicates underflow conditions in the Sample Sequencer FIFOs. The corresponding underflow condition can be 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	Type	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	UV3	R/W1C	0	SS3 FIFO Underflow
				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
				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
				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

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 0s are returned. This bit is cleared by writing a 1.

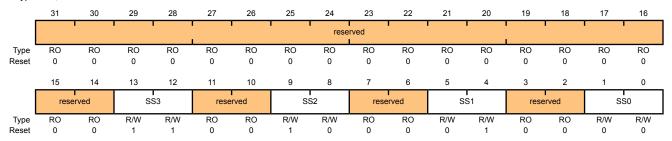
Register 8: ADC Sample Sequencer Priority (ADCSSPRI), offset 0x020

This register sets the priority for each of the Sample Sequencers. Out of reset, Sequencer 0 has the highest priority, and sample sequence 3 has the lowest priority. When reconfiguring sequence priorities, each sequence must have a unique priority or the ADC behavior is inconsistent.

ADC Sample Sequencer Priority (ADCSSPRI)

Base 0x4003.8000

Offset 0x020 Type R/W, reset 0x0000.3210



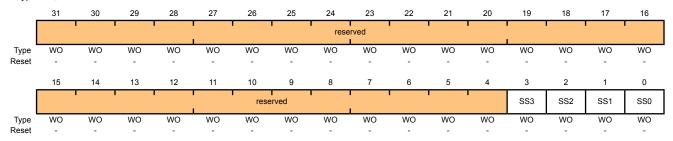
Bit/Field	Name	Туре	Reset	Description
31:14	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
13:12	SS3	R/W	0x3	SS3 Priority
				The SS3 field contains 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. ADC behavior is not consistent 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
				The ${\tt SS2}$ field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 2.
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
				The SS1 field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 1.
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.
1:0	SS0	R/W	0x0	SS0 Priority
				The SS0 field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer 0.

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	WO	-	Software should not rely on the value of 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
				Only a write by software is valid; a read of the register returns no meaningful data. When set by software, sampling is triggered on Sample Sequencer 3, assuming the Sequencer is enabled in the ADCACTSS register.
2	SS2	WO	-	SS2 Initiate
				Only a write by software is valid; a read of the register returns no meaningful data. When set by software, sampling is triggered on Sample Sequencer 2, assuming the Sequencer is enabled in the ADCACTSS register.
1	SS1	WO	-	SS1 Initiate
				Only a write by software is valid; a read of the register returns no meaningful data. When set by software, sampling is triggered on Sample Sequencer 1, assuming the Sequencer is enabled in the ADCACTSS register.
0	SS0	WO	-	SS0 Initiate
				Only a write by software is valid; a read of the register returns no

meaningful data. When set by software, sampling is triggered on Sample Sequencer 0, assuming the Sequencer is enabled in the **ADCACTSS**

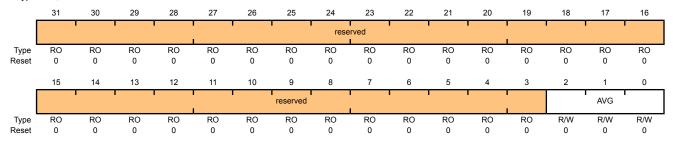
register.

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	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.
2:0	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
0x1	2x hardware oversampling
0x2	4x hardware oversampling
0x3	8x hardware oversampling
0x4	16x hardware oversampling
0x5	32x hardware oversampling
0x6	64x hardware oversampling
0x7	Reserved

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

	reser	ved	MU	IX7	rese	rved	MU	мих6		reserved		MUX5		reserved		X4		
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		
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	reser	ved	MU	IX3	rese	rved	MU	IX2	reserved		MUX1		reserved		мúхо			
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		Nam	ne	Ту	ре	Reset	Des	Description									
	31:30		reserv	/ed	R	0	compatib			Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.								
	29:28		MUX	(7	R/	R/W 0 8th Sample			Sample	Input Se	lect							
					with sam the	The MUX7 field is used during the eighth sample of a sequence with the Sample Sequencer. It specifies which of the analog sampled for the analog-to-digital conversion. The value set her the corresponding pin, for example, a value of 1 indicates th ADC1.					nalog inp et here ir	outs is ndicates						
	27:26		reserv	/ed	R	0	0	com	patibility	ould not with futo cross a r	ure prod	ucts, the	value of	a reserv				
	25:24		MUX	(6	R/	W	0	7th	7th Sample Input Se									
								exe	The MUX6 field is used during the seventh sample of a sequence executed with the Sample Sequencer and specifies which of the analog inputs is sampled for the analog-to-digital conversion.									
	23:22		reserved R		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.										
	21:20		MUX	(5	R/	W	0	6th	Sample	Input Se	lect							
								The MUX5 field is used during the sixth sample of a sequence with the Sample Sequencer and specifies which of the analog sampled for the analog-to-digital conversion.										
	19:18	9:18 reserved RO 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 preserved across a read-modify-write operation.													

Bit/Field	Name	Туре	Reset	Description
17:16	MUX4	R/W	0	5th Sample Input Select
				The $\mathtt{MUX4}$ field is used during the fifth sample of a sequence executed with the Sample Sequencer and 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	0	4th Sample Input Select
				The MUX3 field is used during the fourth sample of a sequence executed with the Sample Sequencer and 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	0	3rd Sample Input Select
				The MUX2 field is used during the third sample of a sequence executed with the Sample Sequencer and 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	0	2nd Sample Input Select
				The $\mathtt{MUX1}$ field is used during the second sample of a sequence executed with the Sample Sequencer and 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	0	1st Sample Input Select
				The MUX0 field is used during the first sample of a sequence executed with the Sample Sequencer and 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 Sample Sequencer 0. 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

_	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
Type •	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
Type •	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

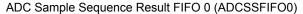
Bit/Field	Name	Type	Reset	Description
31	TS7	R/W	0	8th Sample Temp Sensor Select
				The TS7 bit is used during the eighth sample of the sample sequence and specifies the input source of the sample. If set, the temperature sensor is read. Otherwise, the input pin specified by the ADCSSMUX register is read.
30	IE7	R/W	0	8th Sample Interrupt Enable
				The IE7 bit is used during the eighth sample of the sample sequence and specifies whether the raw interrupt signal (INR0 bit) is asserted at the end of the sample's conversion. If the MASK0 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, otherwise it is not. 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.

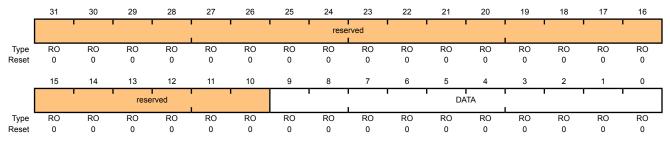
Bit/Field	Name	Туре	Reset	Description
10	IE2	R/W	0	3rd Sample Interrupt Enable
				Same definition as IE7 but used during the third sample.
9	END2	R/W	0	3rd Sample is End of Sequence
				Same definition as END7 but used during the third sample.
8	D2	R/W	0	3rd Sample Diff Input Select
				Same definition as ${\tt D7}$ but used during the third sample.
7	TS1	R/W	0	2nd Sample Temp Sensor Select
				Same definition as ${\tt TS7}$ 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 ${\tt D7}$ but used during the second sample.
3	TS0	R/W	0	1st Sample Temp Sensor Select
				Same definition as ${\tt TS7}$ 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.
				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 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

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.



Base 0x4003.8000 Offset 0x048 Type RO, 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.
9:0	DATA	RO	0x00	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 FIFO, **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	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		FULL		reserved		EMPTY		HP'	TR			TP	TR	
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	1	0	0	0	0	0	0	0	0

Bit/Field	Name	Type	Reset	Description
31:13	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.
12	FULL	RO	0	FIFO Full
				When set, indicates that the FIFO is currently full.
11:9	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	EMPTY	RO	1	FIFO Empty
				When set, indicates that the FIFO is currently empty.
7:4	HPTR	RO	0x00	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	0x00	FIFO Tail Pointer
				This field contains the current "tail" pointer index for the FIFO, that is, the next entry to be read.

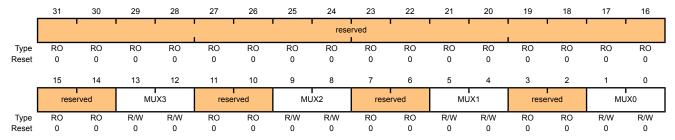
Register 21: ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1), offset 0x060

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 290 for detailed bit descriptions.

ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1)

Base 0x4003.8000 Offset 0x060



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.
13:12	MUX3	R/W	0	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	0	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	0	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	0	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. This register is 16-bits wide and contains information for four possible samples. See the **ADCSSCTL0** register on page 292 for detailed bit descriptions.

ADC Sample Sequence Control 1 (ADCSSCTL1)

Base 0x4003.8000 Offset 0x064

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	1					rese	rved	1						
Туре	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
	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
Type Reset	R/W 0															

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	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.
8	D2	R/W	0	3rd Sample Diff Input Select Same definition as D7 but used during the third sample.

Bit/Field	Name	Туре	Reset	Description
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 $\mathtt{END7}$ but used during the second sample.
4	D1	R/W	0	2nd Sample Diff Input Select
				Same definition as ${\tt D7}$ 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.
				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 25: ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3), offset 0x0A0

This register defines the analog input configuration for each sample in a sequence executed with Sample Sequencer 3. This register is 4-bits wide and contains information for one possible sample. See the **ADCSSMUX0** register on page 290 for detailed bit descriptions.

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

preserved across a read-modify-write operation.

1st Sample Input Select

ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3)

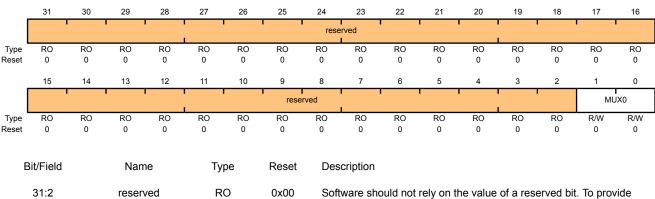
Base 0x4003.8000 Offset 0x0A0

1:0

MUX0

R/W

0

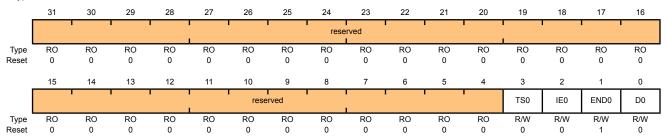


Register 26: ADC Sample Sequence Control 3 (ADCSSCTL3), offset 0x0A4

This register contains the configuration information for each sample for a sequence 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 292 for detailed bit descriptions.

ADC Sample Sequence Control 3 (ADCSSCTL3)

Base 0x4003.8000 Offset 0x0A4



Bit/Field	Name	Type	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	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.

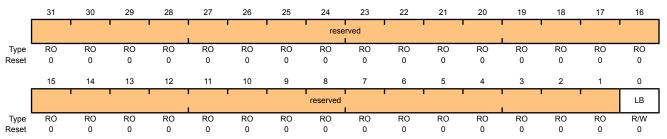
Register 27: ADC Test Mode Loopback (ADCTMLB), offset 0x100

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	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	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.

13 Universal Asynchronous Receivers/Transmitters (UARTs)

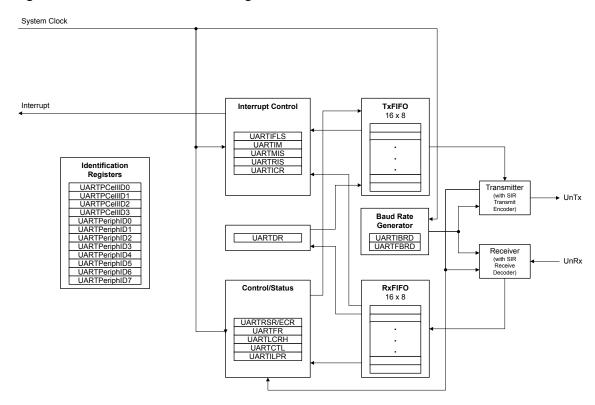
The Stellaris[®] Universal Asynchronous Receiver/Transmitter (UART) provides fully programmable, 16C550-type serial interface characteristics. The LM3S8933 controller is equipped with two UART modules.

Each UART has the following features:

- Separate transmit and receive FIFOs
- 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
- Programmable baud-rate generator allowing rates up to 3.125 Mbps
- Standard asynchronous communication bits for start, stop, and parity
- False start bit detection
- 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

13.1 Block Diagram

Figure 13-1. UART Module Block Diagram



13.2 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 322). 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.

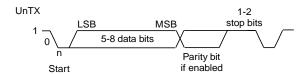
The UART peripheral also includes a serial IR (SIR) encoder/decoder block that can be connected to an infrared transceiver to implement an IrDA SIR physical layer. The SIR function is programmed using the UARTCTL register.

13.2.1 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 13-2 on page 305 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 13-2. UART Character Frame



13.2.2 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 318) and the 6-bit fractional part is loaded with the **UART Fractional Baud-Rate Divisor (UARTFBRD)** register (see page 319). 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 320), 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

13.2.3 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 315) 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 304).

The start bit is valid if UnRx is still low on the eighth cycle of Baud16, otherwise a false start bit is detected and it is ignored. Start bit errors can be viewed in the **UART Receive Status (UARTRSR)** register (see page 313). If the start bit was valid, 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.

13.2.4 **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 317 for more information on IrDA low-power pulse-duration configuration.

Figure 13-3 on page 307 shows the UART transmit and receive signals, with and without IrDA modulation.

UnTx with IrDA

UnRx with IrDA

Figure 13-3. 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.

13.2.5 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 311). 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 320).

FIFO status can be monitored via the **UART Flag (UARTFR)** register (see page 315) 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 324). 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.

13.2.6 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 329).

The interrupt events that can trigger a controller-level interrupt are defined in the **UART Interrupt Mask (UARTIM**) register (see page 326) 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 328).

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 330).

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received over a 32-bit period. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), or when a 1 is written to the corresponding bit in the **UARTICR** register.

13.2.7 Loopback Operation

The UART can be placed into an internal loopback mode for diagnostic or debug work. This is accomplished by setting the $\tt LBE$ bit in the **UARTCTL** register (see page 322). In loopback mode, data transmitted on UnTx is received on the UnRx input.

13.2.8 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.

13.3 Initialization and Configuration

To use the UARTs, the peripheral clock must be enabled by setting the UART0 or UART1 bits in the 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 305, 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 318) should be set to 10. The value to be loaded into the **UARTFBRD** register (see page 319) 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.

13.4 Register Map

Table 13-1 on page 309 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.C000

UART1: 0x4000.D000

Note: The UART must be disabled (see the UARTEN bit in the **UARTCTL** register on page 322) 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 13-1. UART Register Map

Offset	Name	Type	Reset	Description	See page
0x000	UARTDR	R/W	0x0000.0000	UART Data	311
0x004	UARTRSR/UARTECR	R/W	0x0000.0000	UART Receive Status/Error Clear	313
0x018	UARTFR	RO	0x0000.0090	UART Flag	315
0x020	UARTILPR	R/W	0x0000.0000	UART IrDA Low-Power Register	317
0x024	UARTIBRD	R/W	0x0000.0000	UART Integer Baud-Rate Divisor	318

Offset	Name	Туре	Reset	Description	See page
0x028	UARTFBRD	R/W	0x0000.0000	UART Fractional Baud-Rate Divisor	319
0x02C	UARTLCRH	R/W	0x0000.0000	UART Line Control	320
0x030	UARTCTL	R/W	0x0000.0300	UART Control	322
0x034	UARTIFLS	R/W	0x0000.0012	UART Interrupt FIFO Level Select	324
0x038	UARTIM	R/W	0x0000.0000	UART Interrupt Mask	326
0x03C	UARTRIS	RO	0x0000.000F	UART Raw Interrupt Status	328
0x040	UARTMIS	RO	0x0000.0000	UART Masked Interrupt Status	329
0x044	UARTICR	W1C	0x0000.0000	UART Interrupt Clear	330
0xFD0	UARTPeriphID4	RO	0x0000.0000	UART Peripheral Identification 4	332
0xFD4	UARTPeriphID5	RO	0x0000.0000	UART Peripheral Identification 5	333
0xFD8	UARTPeriphID6	RO	0x0000.0000	UART Peripheral Identification 6	334
0xFDC	UARTPeriphID7	RO	0x0000.0000	UART Peripheral Identification 7	335
0xFE0	UARTPeriphID0	RO	0x0000.0011	UART Peripheral Identification 0	336
0xFE4	UARTPeriphID1	RO	0x0000.0000	UART Peripheral Identification 1	337
0xFE8	UARTPeriphID2	RO	0x0000.0018	UART Peripheral Identification 2	338
0xFEC	UARTPeriphID3	RO	0x0000.0001	UART Peripheral Identification 3	339
0xFF0	UARTPCellID0	RO	0x0000.000D	UART PrimeCell Identification 0	340
0xFF4	UARTPCellID1	RO	0x0000.00F0	UART PrimeCell Identification 1	341
0xFF8	UARTPCellID2	RO	0x0000.0005	UART PrimeCell Identification 2	342
0xFFC	UARTPCellID3	RO	0x0000.00B1	UART PrimeCell Identification 3	343

13.5 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

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.

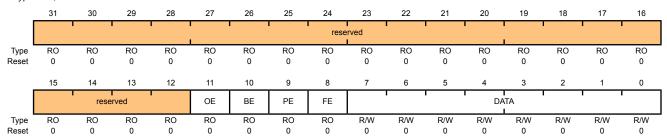
For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If FIFOs are disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received 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	Туре	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 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 set to 1 when a break condition is detected, indicating that 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	Type	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.

Read-Only Receive Status (UARTRSR) Register

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				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							OE	BE	PE	FE
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
31:4	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	Type	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)

stop bit (a valid stop bit is 1).

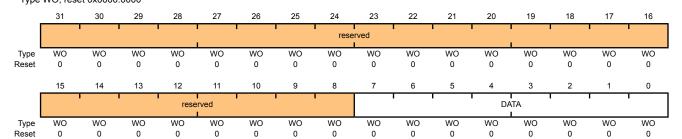
This bit is cleared to 0 by a write to **UARTECR**.

In FIFO mode, this error is associated with the character at the top of the FIFO.

Write-Only Error Clear (UARTECR) Register

UART Receive Status/Error Clear (UARTRSR/UARTECR)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x004 Type WO, reset 0x0000.0000



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

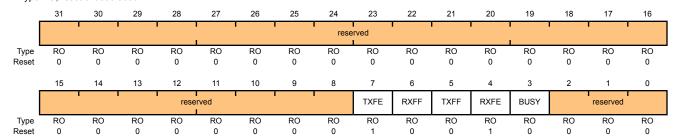
A write to this register of any data clears the framing, parity, break, and overrun flags.

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	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	TXFE	RO	1	UART Transmit FIFO Empty
				The meaning of this bit depends on the state of the ${\tt FEN}$ bit in the ${\tt UARTLCRH}$ 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 ${\tt FEN}$ bit in the ${\tt UARTLCRH}$ 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 ${\tt FEN}$ bit in the ${\tt UARTLCRH}$ register.

If the FIFO is enabled, this bit is set when the transmit FIFO is full.

If the FIFO is disabled, this bit is set when the transmit holding register

is full.

Bit/Field	Name	Туре	Reset	Description
4	RXFE	RO	1	UART Receive FIFO Empty
				The meaning of this bit depends on the state of the ${\tt FEN}$ bit in the ${\tt UARTLCRH}$ register.
				If the FIFO is disabled, this bit is set when the receive holding register is empty.
				If the FIFO is enabled, this bit is set when the receive FIFO is empty.
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}

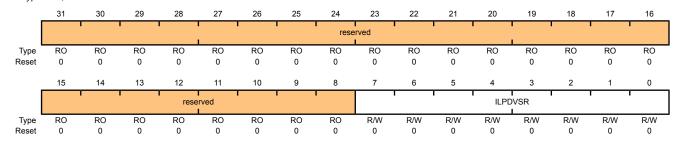
where F_{TrI.PBaud16} is nominally 1.8432 MHz.

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	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.
7:0	ILPDVSR	R/W	0x00	IrDA Low-Power Divisor

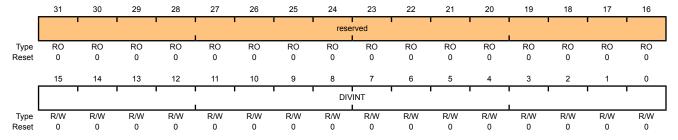
This is an 8-bit low-power divisor value.

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 305 for configuration details.

UART Integer Baud-Rate Divisor (UARTIBRD)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x024



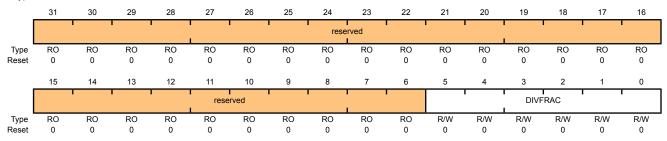
Bit/Field	Name	Туре	Reset	Description
31:16	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15:0	DIVINT	R/W	0x0000	Integer Baud-Rate Divisor

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 305 for configuration details.

UART Fractional Baud-Rate Divisor (UARTFBRD)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x028



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

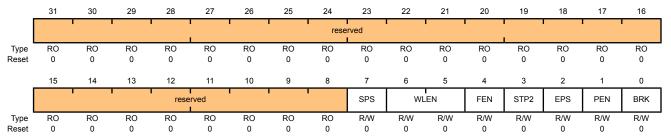
The **UARTLCRH** register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.

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.

UART Line Control (UARTLCRH)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x02C

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	SPS	R/W	0	UART Stick Parity Select
				When bits 1, 2, and 7 of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set and 2 is cleared, the parity bit is transmitted and checked as a 1.
				When this bit is cleared, stick parity is disabled.
6:5	WLEN	R/W	0	UART Word Length
				The bits indicate the number of data bits transmitted or received in a frame as follows:
				Value Description
				0x3 8 bits
				0x2 7 bits
				0x1 6 bits
				0x0 5 bits (default)
4	FEN	R/W	0	UART Enable FIFOs
				If this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO

320 July 25, 2008 **Preliminary**

mode).

When cleared to 0, FIFOs are disabled (Character mode). The FIFOs

become 1-byte-deep holding registers.

Downloaded from Elcodis.com electronic components distributor

Bit/Field	Name	Туре	Reset	Description
3	STP2	R/W	0	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.
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.

 $Downloaded \ from \ \underline{Elcodis.com} \ \ electronic \ components \ distributor$

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.

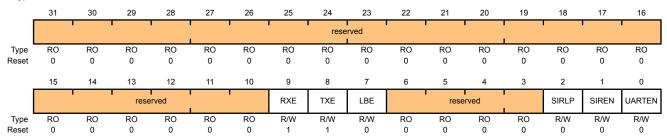
Note: The UARTCTL register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the UARTCTL register.

- 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.
- Enable the UART.

UART Control (UARTCTL)

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x030

Type R/W, reset 0x0000.0300



Bit/Field	Name	Туре	Reset	Description
31:10	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	RXE	R/W	1	UART Receive Enable If this bit is set to 1, the receive section of the UART is enabled. When the UART is disabled in the middle of a receive, it completes the current
				character before stopping.
				Note: To enable reception, the UARTEN bit must also be set.
8	TXE	R/W	1	UART Transmit Enable
				If this bit is set to 1, the transmit section of the UART is enabled. When the UART is disabled in the middle of a transmission, it completes the

Note: To enable transmission, the UARTEN bit must also be set.

current character before stopping.

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 317 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.

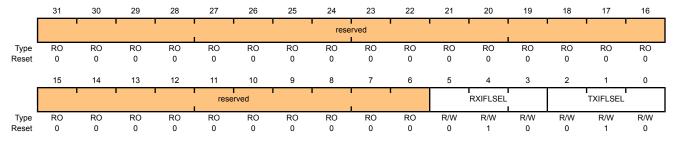
Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

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	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.
5:3	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 ≥ 1/4 full 0x2 RX FIFO ≥ ½ full (default) 0x3 RX FIFO ≥ ¾ full 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 ≤ 1/8 full
				0x1 TX FIFO ≤ ¼ full
				0x2 TX FIFO ≤ ½ full (default)
				0x3 TX FIFO ≤ ¾ full
				0x4 TX FIFO ≤ 7/8 full
				0x5-0x7 Reserved

28

Register 10: UART Interrupt Mask (UARTIM), offset 0x038

25

26

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.

reserved

UART Interrupt Mask (UARTIM)

30

UART0 base: 0x4000.C000 UART1 base: 0x4000.D000 Offset 0x038

Type R/W, reset 0x0000.0000

31

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			OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM		rese	rved	
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
В	it/Field		Nam	e	Ту	ре	Reset	Des	cription							
	31:11		reserv	ed	R	0	0x00	com	patibility	with futu	ıre prodi	he value ucts, the dify-write	value of	a reserv		
	10		OEIN	Л	R/	W	0	On a	a read, tl		nt mask	t Mask for the or		•		ontroller.
	9		BEIN	Л	R/	W	0	On a	a read, tl		nt mask	Mask for the Bi ne BEIM		•		ontroller.
	8		PEIN	1	R/	W	0	On a	a read, tl		nt mask	Mask for the PI		•		ontroller.
	7		FEIN	1	R/	W	0	On a	a read, tl		nt mask	t Mask for the Fine Fine		•		ontroller.

UART Receive Time-Out Interrupt Mask

UART Transmit Interrupt Mask

On a read, the current mask for the RTIM interrupt is returned.

On a read, the current mask for the TXIM interrupt is returned.

Setting this bit to 1 promotes the \mathtt{RTIM} interrupt to the interrupt controller.

Setting this bit to 1 promotes the ${\tt TXIM}$ interrupt to the interrupt controller.

6

5

RTIM

TXIM

R/W

R/W

0

0

Bit/Field	Name	Type	Reset	Description
4	RXIM	R/W	0	UART Receive Interrupt Mask
				On a read, the current mask for the ${\tt RXIM}$ interrupt is returned.
				Setting this bit to 1 promotes the ${\tt RXIM}$ interrupt to the interrupt controller.
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 0x0000.000F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1	1					rese	rved	1				1		1
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			OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS		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	1	1	1	1

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	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	0xF	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

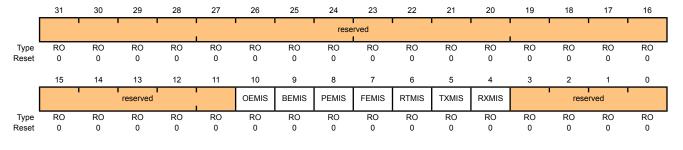
Register 12: UART Masked Interrupt Status (UARTMIS), offset 0x040

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

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.

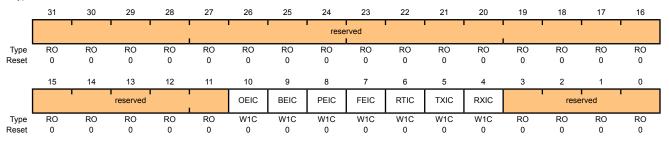
Register 13: UART Interrupt Clear (UARTICR), offset 0x044

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

Type W1C, 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	OEIC	W1C	0	Overrun Error Interrupt Clear The OEIC values are defined as follows: Value Description 0 No effect on the interrupt. 1 Clears interrupt.
9	BEIC	W1C	0	Break Error Interrupt Clear The BEIC values are defined as follows: Value Description 0 No effect on the interrupt. 1 Clears interrupt.
8	PEIC	W1C	0	Parity Error Interrupt Clear The PEIC values are defined as follows:

Value Description

- No effect on the interrupt.
- Clears interrupt.

Bit/Field	Name	Туре	Reset	Description
7	FEIC	W1C	0	Framing Error Interrupt Clear
				The FEIC values are defined as follows:
				Value Description
				0 No effect on the interrupt.
				1 Clears interrupt.
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.

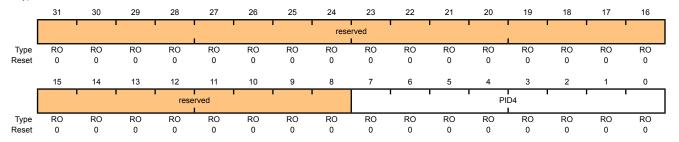
Register 14: UART Peripheral Identification 4 (UARTPeriphID4), offset 0xFD0

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	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	0x0000	UART Peripheral ID Register[7:0]

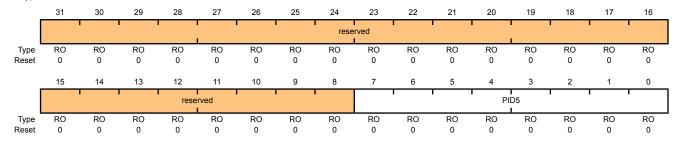
Register 15: UART Peripheral Identification 5 (UARTPeriphID5), offset 0xFD4

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]

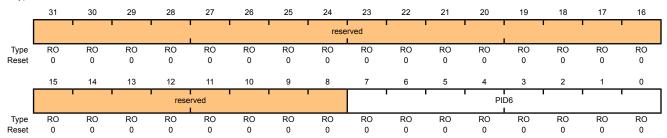
Register 16: UART Peripheral Identification 6 (UARTPeriphID6), offset 0xFD8

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]

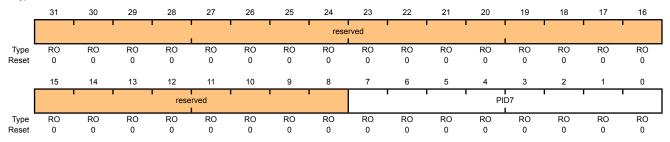
Register 17: UART Peripheral Identification 7 (UARTPeriphID7), offset 0xFDC

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]

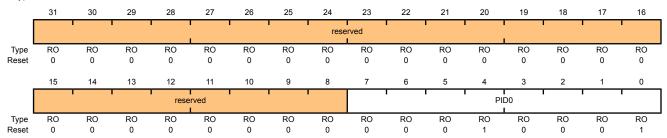
Register 18: UART Peripheral Identification 0 (UARTPeriphID0), offset 0xFE0

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	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	0x11	UART Peripheral ID Register[7:0]

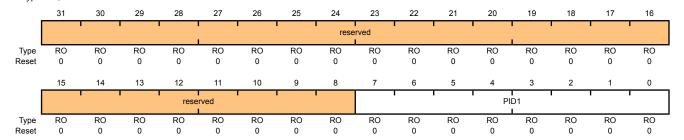
Register 19: UART Peripheral Identification 1 (UARTPeriphID1), offset 0xFE4

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	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	UART Peripheral ID Register[15:8]

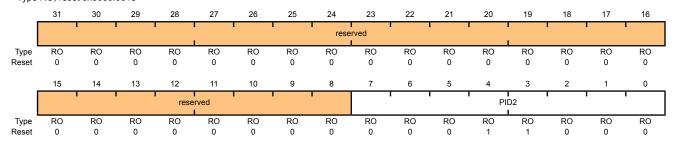
Register 20: UART Peripheral Identification 2 (UARTPeriphID2), offset 0xFE8

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]

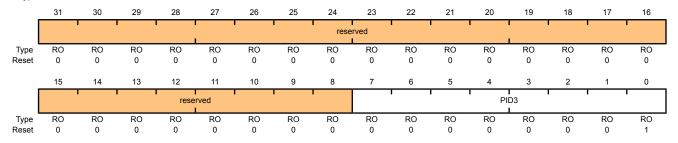
Register 21: UART Peripheral Identification 3 (UARTPeriphID3), offset 0xFEC

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	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	UART Peripheral ID Register[31:24]

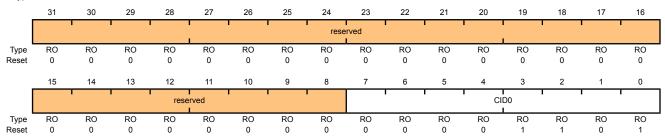
Register 22: UART PrimeCell Identification 0 (UARTPCellID0), offset 0xFF0

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]

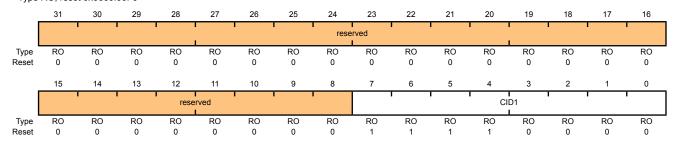
Register 23: UART PrimeCell Identification 1 (UARTPCellID1), offset 0xFF4

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	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	UART PrimeCell ID Register[15:8]

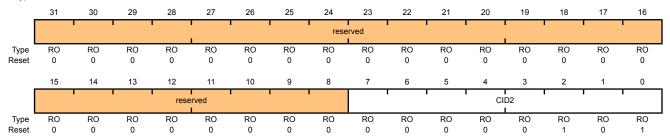
Register 24: UART PrimeCell Identification 2 (UARTPCellID2), offset 0xFF8

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

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	UART PrimeCell ID Register[23:16]

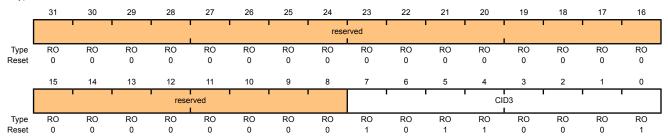
Register 25: UART PrimeCell Identification 3 (UARTPCellID3), offset 0xFFC

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	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	UART PrimeCell ID Register[31:24]

14 Synchronous Serial Interface (SSI)

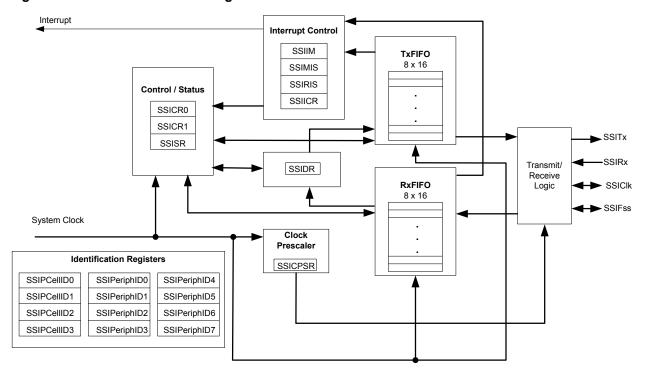
The Stellaris[®] Synchronous Serial Interface (SSI) is a master or slave interface for synchronous serial communication with peripheral devices that have either Freescale SPI, MICROWIRE, or Texas Instruments synchronous serial interfaces.

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

14.1 Block Diagram

Figure 14-1. SSI Module Block Diagram



14.2 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.

14.2.1 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 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 363). 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 356).

The frequency of the output clock SSIClk is defined by:

```
SSIClk = FSysClk / (CPSDVSR * (1 + SCR))
```

Note: Although the SSIClk transmit clock can theoretically be 25 MHz, the module may not be able to operate at that speed. 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 553 to view SSI timing parameters.

14.2.2 FIFO Operation

14.2.2.1 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 360), 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.

14.2.2.2 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.

14.2.3 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 364). 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 366 and page 367, respectively).

14.2.4 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.

14.2.4.1 Texas Instruments Synchronous Serial Frame Format

Figure 14-2 on page 347 shows the Texas Instruments synchronous serial frame format for a single transmitted frame.

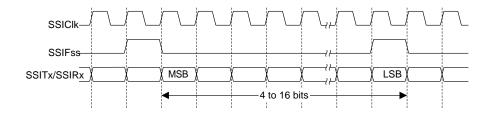
Figure 14-2. TI Synchronous Serial Frame Format (Single Transfer)

In this mode, <code>SSIClk</code> and <code>SSIFss</code> are forced Low, and the transmit data line <code>SSITx</code> is tristated whenever the SSI is idle. Once the bottom entry of the transmit FIFO contains data, <code>SSIFss</code> is pulsed High for one <code>SSIClk</code> 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 <code>SSIClk</code>, the MSB of the 4 to 16-bit data frame is shifted out on the <code>SSITx</code> pin. Likewise, the MSB of the received data is shifted onto the <code>SSIRx</code> 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 14-3 on page 347 shows the Texas Instruments synchronous serial frame format when back-to-back frames are transmitted.

Figure 14-3. TI Synchronous Serial Frame Format (Continuous Transfer)



14.2.4.2 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 SSIClk 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 SSIClk pin. If the SPO bit is High, a steady state High value is placed on the SSIClk pin when data is not being transferred.

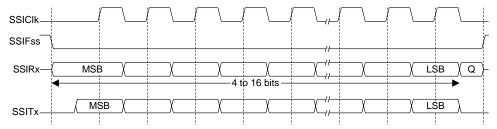
SPH Phase Control Bit

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.

14.2.4.3 Freescale SPI Frame Format with SPO=0 and SPH=0

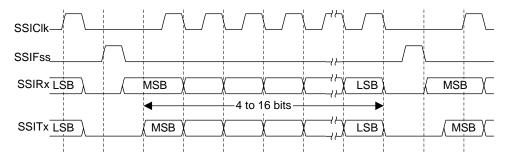
Single and continuous transmission signal sequences for Freescale SPI format with SPO=0 and SPH=0 are shown in Figure 14-4 on page 348 and Figure 14-5 on page 348.

Figure 14-4. Freescale SPI Format (Single Transfer) with SPO=0 and SPH=0



Note: Q is undefined.

Figure 14-5. Freescale SPI Format (Continuous Transfer) with SPO=0 and SPH=0



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. This causes slave data to be enabled onto the SSIRx input line of the master. The master SSITx output pad is enabled.

One half SSIC1k period later, valid master data is transferred to the SSITx pin. Now that both the master and slave data have been set, the SSIC1k master clock pin goes High after one further half SSIC1k period.

The data is now captured on the rising and propagated on the falling edges of the SSIC1k 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 SSIClk period after the last bit has been captured.

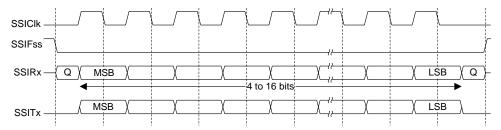
However, in the case of continuous back-to-back transmissions, the SSIFss signal must be pulsed High between each data word transfer. This is because the slave select pin freezes the data in its

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.

14.2.4.4 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 14-6 on page 349, which covers both single and continuous transfers.

Figure 14-6. 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.

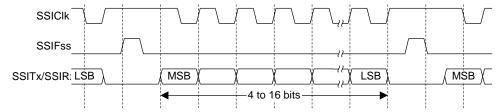
14.2.4.5 Freescale SPI Frame Format with SPO=1 and SPH=0

Single and continuous transmission signal sequences for Freescale SPI format with SPO=1 and SPH=0 are shown in Figure 14-7 on page 350 and Figure 14-8 on page 350.

Figure 14-7. Freescale SPI Frame Format (Single Transfer) with SPO=1 and SPH=0

Note: Q is undefined.

Figure 14-8. Freescale SPI Frame Format (Continuous Transfer) with SPO=1 and SPH=0



In this configuration, during idle periods:

- SSIC1k 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, 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 \mathtt{SSITx} line. Now that both the master and slave data have been set, the \mathtt{SSIClk} master clock pin becomes Low after one further half \mathtt{SSIClk} period. This means that data is captured on the falling edges and propagated on the rising edges of the \mathtt{SSIClk} signal.

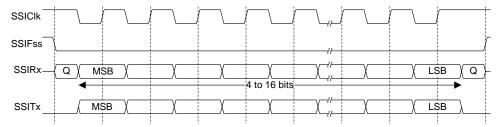
In the case of a single word transmission, after all bits of the data word are transferred, the SSIFss line is returned to its idle High state one SSIClk period after the last bit has been captured.

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.

14.2.4.6 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 14-9 on page 351, which covers both single and continuous transfers.

Figure 14-9. 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.

14.2.4.7 MICROWIRE Frame Format

Figure 14-10 on page 352 shows the MICROWIRE frame format, again for a single frame. Figure 14-11 on page 353 shows the same format when back-to-back frames are transmitted.

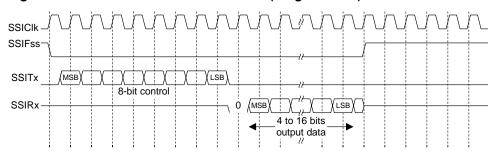


Figure 14-10. MICROWIRE Frame Format (Single Frame)

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:

- SSIClk 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 SSITxpin. SSIFss remains Low for the duration of the frame transmission. The SSIRxpin 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 SSIC1k 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.

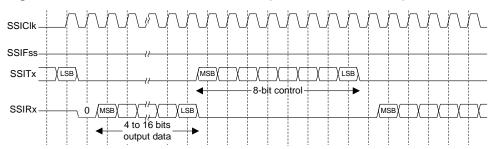


Figure 14-11. MICROWIRE Frame Format (Continuous Transfer)

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 14-12 on page 353 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.

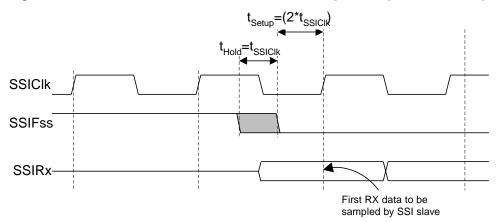


Figure 14-12. MICROWIRE Frame Format, SSIFss Input Setup and Hold Requirements

14.3 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))
1x106 = 20x106 / (CPSDVSR * (1 + SCR))
```

In this case, if CPSDVSR=2, SCR must be 9.

The configuration sequence would be as follows:

- Ensure that the SSE bit in the SSICR1 register is disabled.
- 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.

14.4 Register Map

Table 14-1 on page 354 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: The SSI must be disabled (see the SSE bit in the **SSICR1** register) before any of the control registers are reprogrammed.

Table 14-1. SSI Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	SSICR0	R/W	0x0000.0000	SSI Control 0	356

Offset	Name	Туре	Reset	Description	See page
0x004	SSICR1	R/W	0x0000.0000	SSI Control 1	358
0x008	SSIDR	R/W	0x0000.0000	SSI Data	360
0x00C	SSISR	RO	0x0000.0003	SSI Status	361
0x010	SSICPSR	R/W	0x0000.0000	SSI Clock Prescale	363
0x014	SSIIM	R/W	0x0000.0000	SSI Interrupt Mask	364
0x018	SSIRIS	RO	0x0000.0008	SSI Raw Interrupt Status	366
0x01C	SSIMIS	RO	0x0000.0000	SSI Masked Interrupt Status	367
0x020	SSIICR	W1C	0x0000.0000	SSI Interrupt Clear	368
0xFD0	SSIPeriphID4	RO	0x0000.0000	SSI Peripheral Identification 4	369
0xFD4	SSIPeriphID5	RO	0x0000.0000	SSI Peripheral Identification 5	370
0xFD8	SSIPeriphID6	RO	0x0000.0000	SSI Peripheral Identification 6	371
0xFDC	SSIPeriphID7	RO	0x0000.0000	SSI Peripheral Identification 7	372
0xFE0	SSIPeriphID0	RO	0x0000.0022	SSI Peripheral Identification 0	373
0xFE4	SSIPeriphID1	RO	0x0000.0000	SSI Peripheral Identification 1	374
0xFE8	SSIPeriphID2	RO	0x0000.0018	SSI Peripheral Identification 2	375
0xFEC	SSIPeriphID3	RO	0x0000.0001	SSI Peripheral Identification 3	376
0xFF0	SSIPCellID0	RO	0x0000.000D	SSI PrimeCell Identification 0	377
0xFF4	SSIPCellID1	RO	0x0000.00F0	SSI PrimeCell Identification 1	378
0xFF8	SSIPCellID2	RO	0x0000.0005	SSI PrimeCell Identification 2	379
0xFFC	SSIPCellID3	RO	0x0000.00B1	SSI PrimeCell Identification 3	380

14.5 Register Descriptions

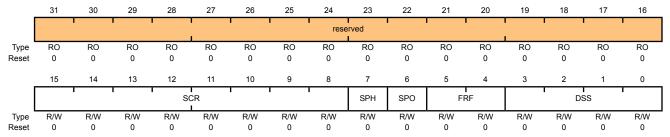
The remainder of this section lists and describes the SSI registers, in numerical order by address offset.

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 ${\tt SPH}$ bit is 0, data is captured on the first clock edge transition. If ${\tt SPH}$ is 1, data is captured on the second clock edge transition.

SSI Serial Clock Polarity

This bit is only applicable to the Freescale SPI Format.

When the ${\tt SPO}$ bit is 0, it produces a steady state Low value on the SSIC1k pin. If SPO is 1, a steady state High value is placed on the SSIC1k pin when data is not being transferred.

6

SPO

R/W

0

Bit/Field	Name	Туре	Reset	Description
5:4	FRF	R/W	0x0	SSI Frame Format Select
				The FRF values are defined as follows:
				Value Frame Format
				0x0 Freescale SPI Frame Format
				0x1 Texas Intruments Synchronous Serial Frame Format
				0x2 MICROWIRE Frame Format
				0x3 Reserved
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.

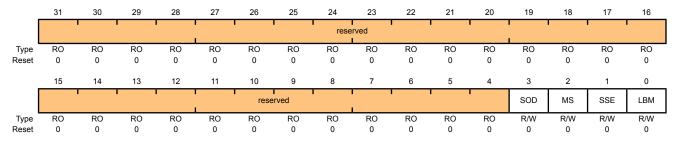
SSI Control 1 (SSICR1)

SSI0 base: 0x4000.8000

3

SOD

Offset 0x004 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.

0

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:

SSI Slave Mode Output Disable

Value Description

- SSI can drive SSITx output in Slave Output mode.
- SSI must not drive the SSITx output in Slave mode.

2 MS R/W 0 SSI Master/Slave Select

R/W

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

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.

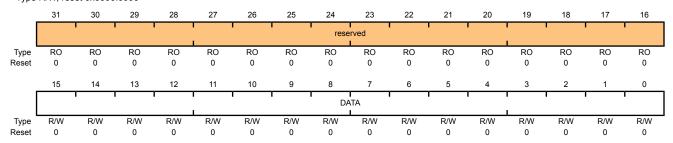
When the SSI is programmed for MICROWIRE frame format, the default size for transmit data is eight bits (the most significant byte is ignored). The receive data size is controlled by the programmer. The transmit FIFO and the receive FIFO are not cleared even when the SSE bit in the **SSICR1** register 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

D:4/E: -1-4

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

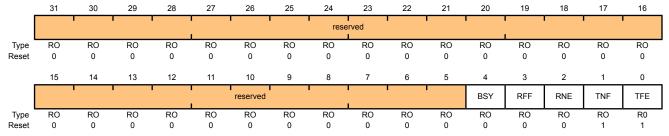
SSISR is a status register that contains bits that indicate the FIFO fill status and the SSI busy status.

SSI Status (SSISR)

SSI0 base: 0x4000.8000

Offset 0x00C

Type RO, reset 0x0000.0003



Name	Туре	Reset	Description
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.
BSY	RO	0	SSI Busy Bit
			The BSY values are defined as follows:
			 Value Description SSI is idle. SSI is currently transmitting and/or receiving a frame, or the transmit FIFO is not empty.
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.
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.
TNF	RO	1	SSI Transmit FIFO Not Full
			The TNF values are defined as follows:
			Value Description
			0 Transmit FIFO is full.
	RFF RNE	reserved RO BSY RO RFF RO RNE RO	reserved RO 0x00 BSY RO 0 RFF RO 0 RNE RO 0

Transmit FIFO is not full.

Bit/Field	Name	Туре	Reset	Description
0	TFE	R0	1	SSI Transmit FIFO Empty
				The ${\tt TFE}$ values are defined as follows:
				Value Description
				0 Transmit FIFO is not empty.
				1 Transmit FIFO is empty.

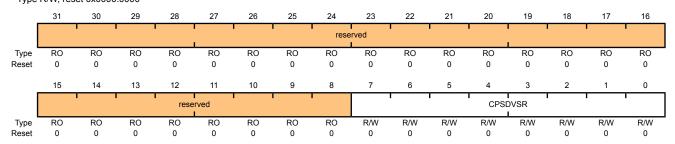
Register 5: SSI Clock Prescale (SSICPSR), offset 0x010

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	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	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.

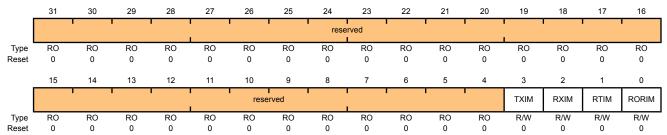
Register 6: SSI Interrupt Mask (SSIIM), offset 0x014

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-full or less condition interrupt is masked. 1 TX FIFO half-full 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.

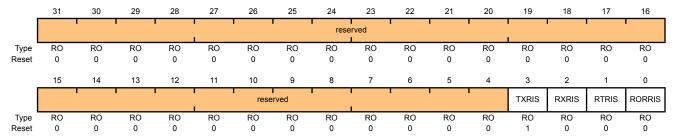
Bit/Field	Name	Туре	Reset	Description
0	RORIM	R/W	0	SSI Receive Overrun Interrupt Mask
				The RORIM values are defined as follows:
				Value Description
				0 RX FIFO overrun interrupt is masked.
				1 RX FIFO overrun interrupt is not masked.

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



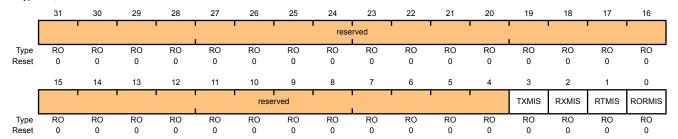
Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TXRIS	RO	1	SSI Transmit FIFO Raw Interrupt Status Indicates that the transmit FIFO is half full 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



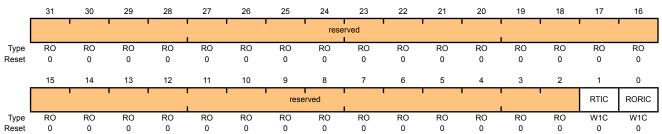
Bit/Field	Name	Туре	Reset	Description
31:4	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3	TXMIS	RO	0	SSI Transmit FIFO Masked Interrupt Status Indicates that the transmit FIFO is half full 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	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	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

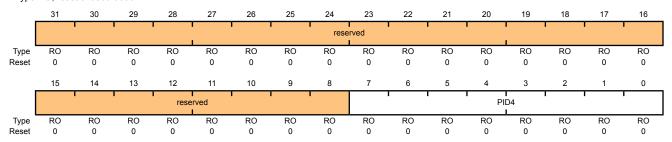
- No effect on interrupt.
- Clears interrupt.

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



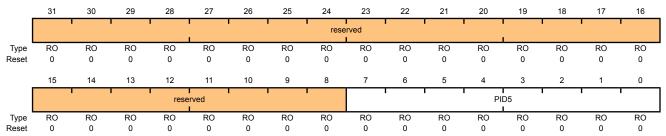
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID4	RO	0x00	SSI Peripheral ID Register[7:0]

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



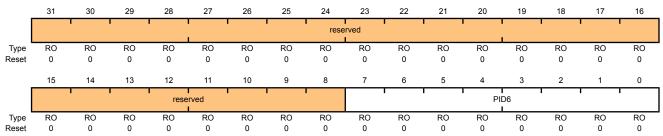
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID5	RO	0x00	SSI Peripheral ID Register[15:8]

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



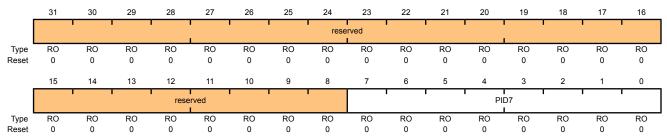
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID6	RO	0x00	SSI Peripheral ID Register[23:16]

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



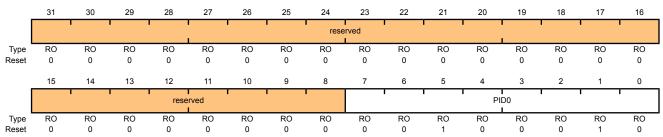
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID7	RO	0x00	SSI Peripheral ID Register[31:24]

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



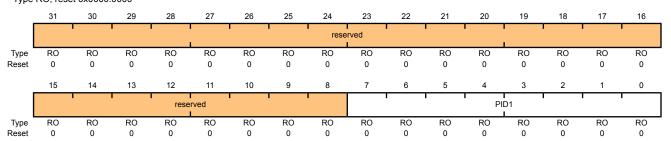
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID0	RO	0x22	SSI Peripheral ID Register[7:0]

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



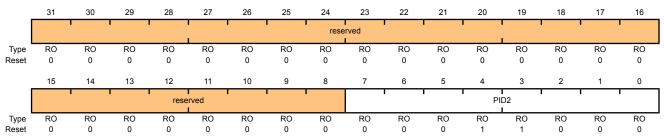
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID1	RO	0x00	SSI Peripheral ID Register [15:8]

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



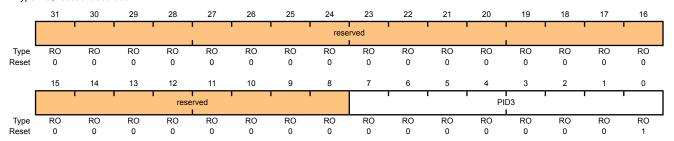
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID2	RO	0x18	SSI Peripheral ID Register [23:16]

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



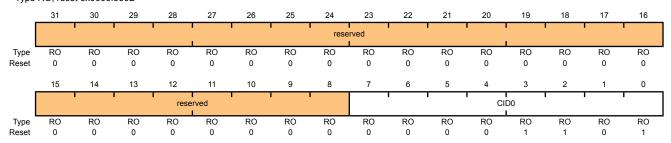
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	PID3	RO	0x01	SSI Peripheral ID Register [31:24]

Register 18: SSI PrimeCell Identification 0 (SSIPCellID0), offset 0xFF0

The SSIPCeIIIDn 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



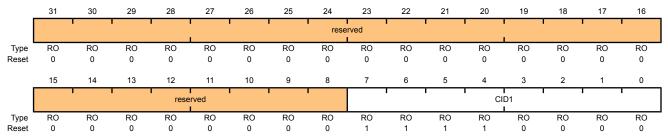
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID0	RO	0x0D	SSI PrimeCell ID Register [7:0]

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



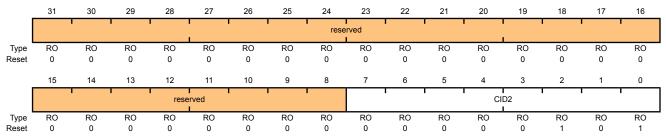
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID1	RO	0xF0	SSI PrimeCell ID Register [15:8]

Register 20: SSI PrimeCell Identification 2 (SSIPCellID2), offset 0xFF8

The SSIPCeIIIDn 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



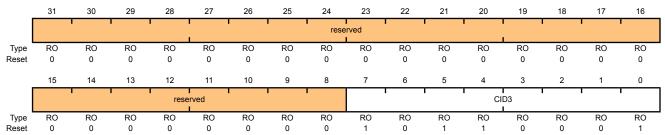
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	CID2	RO	0x05	SSI PrimeCell ID Register [23:16]

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]

15 Inter-Integrated Circuit (I²C) Interface

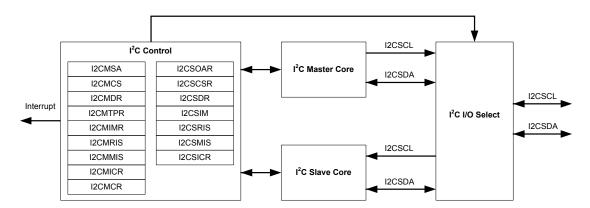
The Inter-Integrated Circuit (I^2C) bus provides bi-directional data transfer through a two-wire design (a serial data line SDA and a serial clock line SCL), and interfaces to external I^2C devices such as serial memory (RAMs and ROMs), networking devices, LCDs, tone generators, and so on. The I^2C bus may also be used for system testing and diagnostic purposes in product development and manufacture. The LM3S8933 microcontroller includes one I^2C module, providing the ability to interact (both send and receive) with other I^2C devices on the bus.

Devices on the I²C bus can be designated as either a master or a slave. The Stellaris[®] 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. There are a total of four I²C modes: Master Transmit, Master Receive, Slave Transmit, and Slave Receive. The 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) and the I²C slave generates interrupts when data has been sent or requested by a master.

15.1 Block Diagram

Figure 15-1. I²C Block Diagram

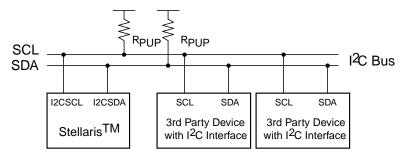


15.2 Functional Description

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 15-2 on page 382.

See "I²C" on page 549 for I²C timing diagrams.

Figure 15-2. I²C Bus Configuration



15.2.1 I²C Bus Functional Overview

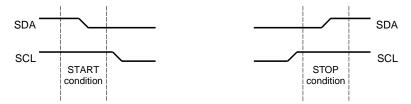
The I²C bus uses only two signals: SDA and SCL, named I2CSDA and I2CSCL on Stellaris[®] microcontrollers. SDA is the bi-directional serial data line and SCL is the bi-directional serial clock line. The bus is considered idle when both lines are high.

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 382) 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.

15.2.1.1 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 15-3 on page 382.

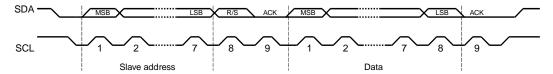
Figure 15-3. START and STOP Conditions



15.2.1.2 Data Format with 7-Bit Address

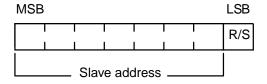
Data transfers follow the format shown in Figure 15-4 on page 383. After the START condition, a slave address is sent. This address is 7-bits long followed by an eighth bit, which is a data direction bit (\mathbb{R}/\mathbb{S} bit in the **I2CMSA** register). A zero indicates a transmit operation (send), 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 15-4. Complete Data Transfer with a 7-Bit Address



The first seven bits of the first byte make up the slave address (see Figure 15-5 on page 383). 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.

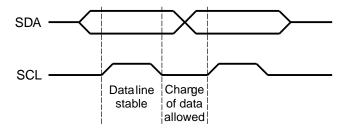
Figure 15-5. R/S Bit in First Byte



15.2.1.3 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 15-6 on page 383).

Figure 15-6. Data Validity During Bit Transfer on the I²C Bus



15.2.1.4 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 383.

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.

15.2.1.5 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.

15.2.2 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 401).

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 15-1 on page 384 gives examples of timer period, system clock, and speed mode (Standard or Fast).

Table 15-1. 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
33Mhz	0x10	97.1 Kbps	0x04	330 Kbps
40Mhz	0x13	100 Kbps	0x04	400 Kbps

System Clock	Timer Period	Standard Mode	Timer Period	Fast Mode
50Mhz	0x18	100 Kbps	0x06	357 Kbps

15.2.3 Interrupts

The I²C can generate interrupts when the following conditions are observed:

- Master transaction completed
- 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.

15.2.3.1 I²C Master Interrupts

The I²C master module generates an interrupt when a transaction completes (either transmit or receive), or when an error occurs during a transaction. To enable the I²C master interrupt, software must write a '1' to the I²C Master Interrupt Mask (I2CMIMR) register. When an interrupt condition is met, software must check the ERROR bit in the I²C Master Control/Status (I2CMCS) register to verify that an error didn't occur during the last transaction. An error condition is asserted if the last transaction wasn't acknowledge by the slave or if the master was forced to give up ownership of the bus due to a lost arbitration round with another master. If an error is not detected, the application can proceed with the transfer. The interrupt is cleared by writing a '1' to the I²C 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.

15.2.3.2 I²C Slave Interrupts

The slave module generates interrupts as it receives requests from an I^2C master. To enable the I^2C slave interrupt, write a '1' to 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 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.

15.2.4 Loopback Operation

The I^2C modules can be placed into an internal loopback mode for diagnostic or debug work. This is accomplished by setting the LPBK bit in the I^2C Master Configuration (I2CMCR) register. In loopback mode, the SDA and SCL signals from the master and slave modules are tied together.

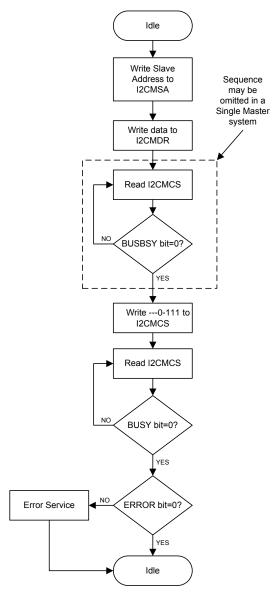
15.2.5 Command Sequence Flow Charts

This section details the steps required to perform the various I^2C transfer types in both master and slave mode.

15.2.5.1 I²C Master Command Sequences

The figures that follow show the command sequences available for the I²C master.

Figure 15-7. Master Single SEND



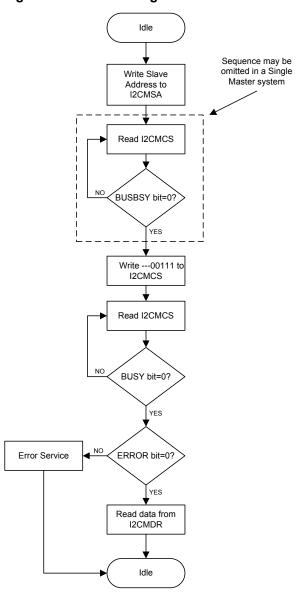


Figure 15-8. Master Single RECEIVE

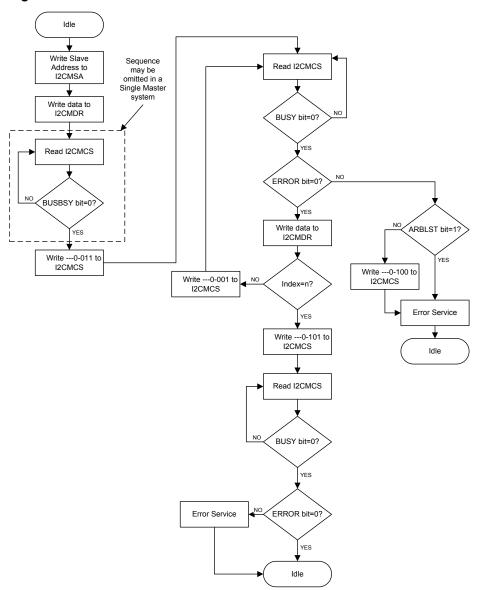


Figure 15-9. Master Burst SEND

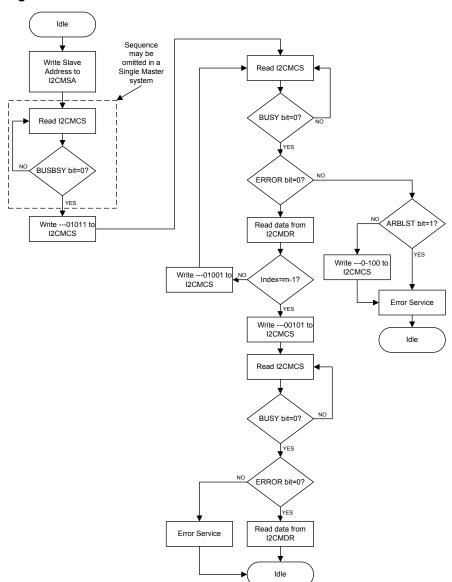


Figure 15-10. Master Burst RECEIVE

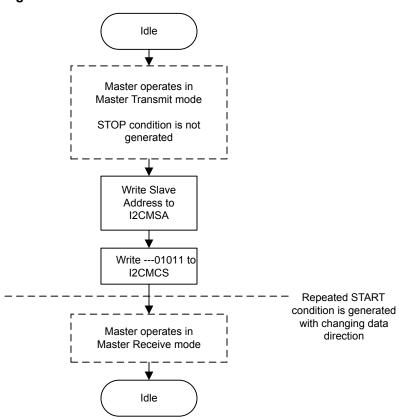


Figure 15-11. Master Burst RECEIVE after Burst SEND

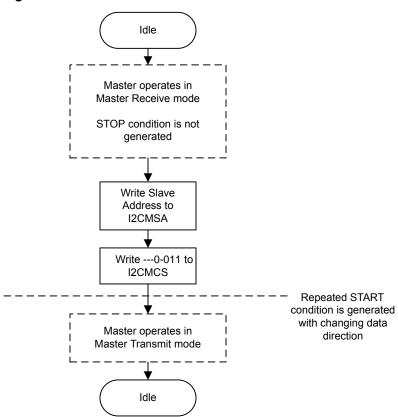


Figure 15-12. Master Burst SEND after Burst RECEIVE

15.2.5.2 I²C Slave Command Sequences

Figure 15-13 on page 392 presents the command sequence available for the I²C slave.

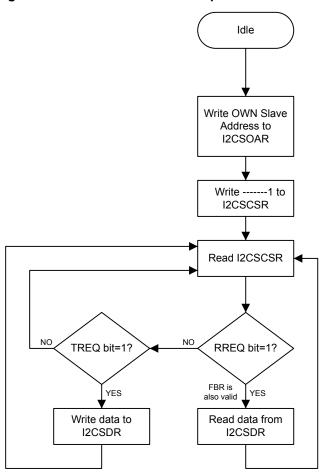


Figure 15-13. Slave Command Sequence

15.3 Initialization and Configuration

The following example shows how to configure the I^2C 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.

15.4 Register Map

Table 15-2 on page 393 lists the I²C registers. All addresses given are relative to the I²C base addresses for the master and slave:

I²C Master 0: 0x4002.0000

I²C Slave 0: 0x4002.0800

Table 15-2. Inter-Integrated Circuit (I²C) Interface Register Map

Offset	Name	Туре	Reset	Description	See page				
I ² C Maste	² C Master								
0x000	I2CMSA	R/W	0x0000.0000	I2C Master Slave Address	395				
0x004	I2CMCS	R/W	0x0000.0000	I2C Master Control/Status	396				
0x008	I2CMDR	R/W	0x0000.0000	I2C Master Data	400				
0x00C	I2CMTPR	R/W	0x0000.0001	I2C Master Timer Period	401				
0x010	I2CMIMR	R/W	0x0000.0000	I2C Master Interrupt Mask	402				
0x014	I2CMRIS	RO	0x0000.0000	I2C Master Raw Interrupt Status	403				
0x018	I2CMMIS	RO	0x0000.0000	I2C Master Masked Interrupt Status	404				
0x01C	I2CMICR	WO	0x0000.0000	I2C Master Interrupt Clear	405				
0x020	I2CMCR	R/W	0x0000.0000	I2C Master Configuration	406				
I ² C Slave									
0x000	I2CSOAR	R/W	0x0000.0000	I2C Slave Own Address	408				
0x004	I2CSCSR	RO	0x0000.0000	I2C Slave Control/Status	409				
0x008	I2CSDR	R/W	0x0000.0000	I2C Slave Data	411				
0x00C	I2CSIMR	R/W	0x0000.0000	I2C Slave Interrupt Mask	412				

Offset	Name	Туре	Reset	Description	See page
0x010	I2CSRIS	RO	0x0000.0000	I2C Slave Raw Interrupt Status	413
0x014	I2CSMIS	RO	0x0000.0000	I2C Slave Masked Interrupt Status	414
0x018	I2CSICR	WO	0x0000.0000	I2C Slave Interrupt Clear	415

15.5 Register Descriptions (I²C Master)

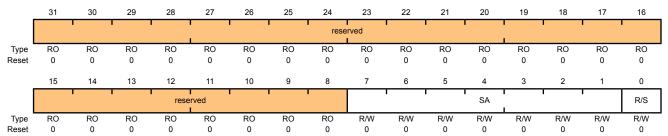
The remainder of this section lists and describes the I^2C master registers, in numerical order by address offset. See also "Register Descriptions (I2C Slave)" on page 407.

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 Master 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 R/S bit specifies if the next operation is a Receive (High) or Send (Low).

Value Description

0 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.

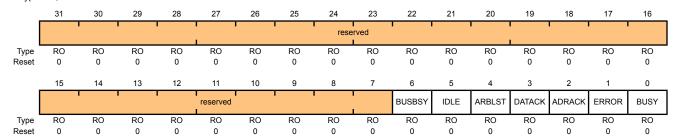
The STOP bit determines if the cycle stops at the end of the data cycle, or continues on to a burst. To generate a single send cycle, the I^2C Master Slave Address (I2CMSA) register is written with the desired address, the R/S bit is set to 0, and the Control register is written with ACK=X (0 or 1), 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.

Read-Only Status Register

I2C Master Control/Status (I2CMCS)

I2C Master 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^2C 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	I ² C Idle
				This bit specifies the 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.

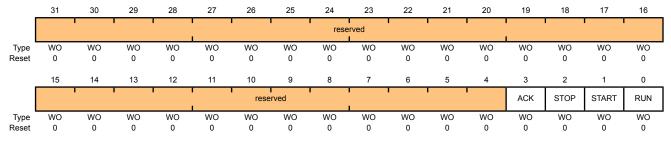
Bit/Field	Name	Type	Reset	Description
3	DATACK	RO	0	Acknowledge Data
				This bit specifies the result of the last data operation. If set, the transmitted data was not acknowledged; otherwise, the data was acknowledged.
2	ADRACK	RO	0	Acknowledge Address
				This bit specifies the result of the last address 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, the transmit data not being acknowledged, or because the controller lost arbitration.
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.

Write-Only Control Register

I2C Master Control/Status (I2CMCS)

I2C Master 0 base: 0x4002.0000 Offset 0x004 Type WO, reset 0x0000.0000



Bit/Field	Name	Type	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 15-3 on page 398.
2	STOP	WO	0	Generate STOP When set, causes the generation of the STOP condition. See field

decoding in Table 15-3 on page 398.

Bit/Field	Name	Type	Reset	Description
1	START	WO	0	Generate START
				When set, causes the generation of a START or repeated START condition. See field decoding in Table 15-3 on page 398.
0	RUN	WO	0	I ² C Master Enable

When set, allows the master to send or receive data. See field decoding in Table 15-3 on page 398.

Table 15-3. Write Field Decoding for I2CMCS[3:0] Field (Sheet 1 of 3)

	I2CMSA[0]		I2CMC	S[3:0]		Description
State	R/S	ACK	STOP	START	RUN	
Idle	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).
	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	mbinations	not listed	are non-or	perations.	NOP.
Master Transmit	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).
	0	Х	1	1	1	Repeated START condition followed by SEND and STOP condition (master goes to Idle state).
	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	mbinations	s not listed	are non-or	perations.	NOP.

Current	I2CMSA[0]		I2CMC	CS[3:0]		Description
State	R/S	ACK	STOP	START	RUN	
Master Receive	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.
	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	not listed	are non-op	perations.	NOP.

a. An X in a table cell indicates the bit can be 0 or 1.

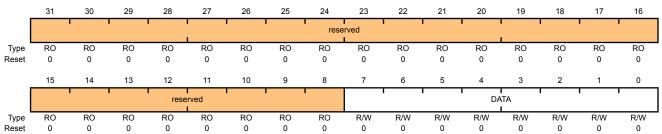
b. In Master Receive mode, a STOP condition should be generated only after a Data Negative Acknowledge executed by the master or an Address Negative Acknowledge executed by the slave.

Register 3: I²C Master Data (I2CMDR), offset 0x008

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 Master 0 base: 0x4002.0000 Offset 0x008 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	0x00	Data Transferred

Data transferred during transaction.

Register 4: I²C Master Timer Period (I2CMTPR), offset 0x00C

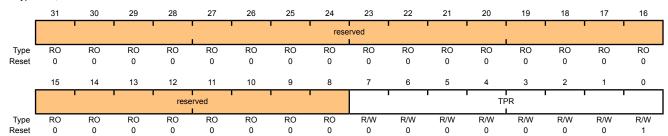
This register specifies the period of the SCL clock.

I2C Master Timer Period (I2CMTPR)

I2C Master 0 base: 0x4002.0000

Offset 0x00C

Type R/W, 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	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 255).

 ${\tt SCL_LP}$ 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

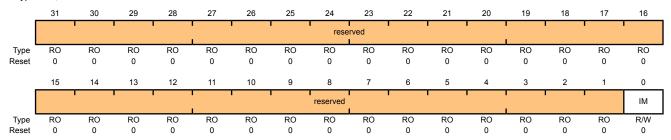
This register controls whether a raw interrupt is promoted to a controller interrupt.

I2C Master Interrupt Mask (I2CMIMR)

I2C Master 0 base: 0x4002.0000

Offset 0x010

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	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

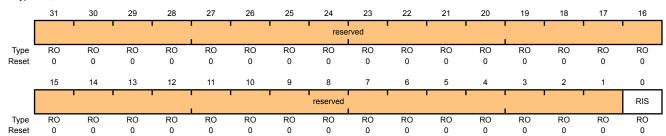
This register specifies whether an interrupt is pending.

I2C Master Raw Interrupt Status (I2CMRIS)

I2C Master 0 base: 0x4002.0000

Offset 0x014

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	RIS	RO	0	Raw Interrupt Status

This bit specifies the raw interrupt state (prior to masking) of the I^2C master block. If set, an interrupt is pending; otherwise, an interrupt is not pending.

Register 7: I²C Master Masked Interrupt Status (I2CMMIS), offset 0x018

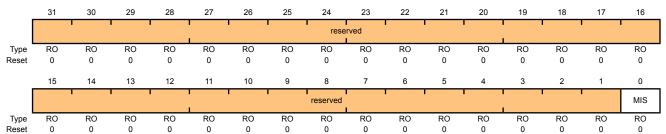
This register specifies whether an interrupt was signaled.

I2C Master Masked Interrupt Status (I2CMMIS)

I2C Master 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 I^2C 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

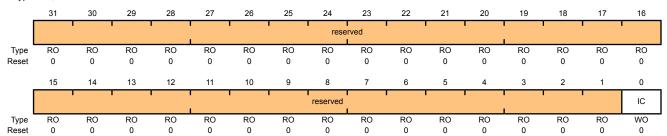
This register clears the raw interrupt.

I2C Master Interrupt Clear (I2CMICR)

I2C Master 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

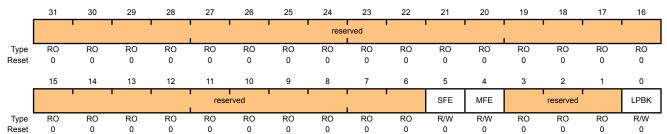
This register configures the mode (Master or Slave) and sets the interface for test mode loopback.

I2C Master Configuration (I2CMCR)

I2C Master 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.

15.6 Register Descriptions (I2C Slave)

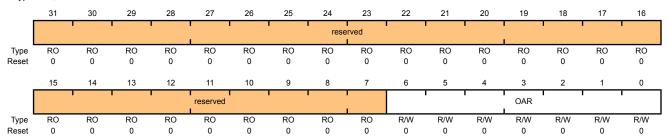
The remainder of this section lists and describes the I^2C slave registers, in numerical order by address offset. See also "Register Descriptions (I^2C Master)" on page 394.

Register 10: I²C Slave Own Address (I2CSOAR), offset 0x000

This register consists of seven address bits that identify the Stellaris[®] I²C device on the I²C bus.

I2C Slave Own Address (I2CSOAR)

I2C Slave 0 base: 0x4002.0800 Offset 0x000 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 0x004

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 $^{\$}$ I²C slave operation.

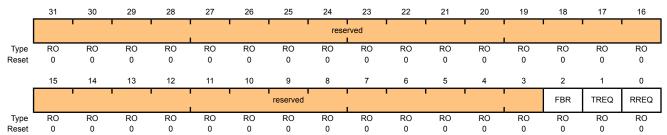
Read-Only Status Register

I2C Slave Control/Status (I2CSCSR)

I2C Slave 0 base: 0x4002.0800

Offset 0x004

Type RO, 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	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 RREQ bit is set, and is automatically cleared when data has been read from the I2CSDR 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 I^2CSDR register. Otherwise, there is no outstanding transmit request.

Bit/Field	Name	Type	Reset	Description
0	RREQ	RO	0	Receive Request

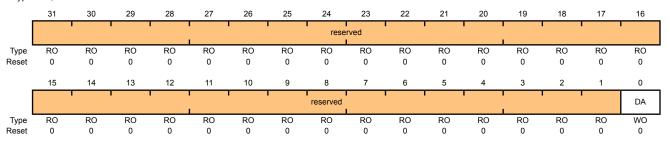
This bit specifies the status of the I^2C slave with regards to outstanding receive requests. If set, the I^2C unit has outstanding receive data from the I^2C master and uses clock stretching to delay the master until the data has been read from the **I2CSDR** register. Otherwise, no receive data is outstanding.

Write-Only Control Register

I2C Slave Control/Status (I2CSCSR)

I2C Slave 0 base: 0x4002.0800 Offset 0x004

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

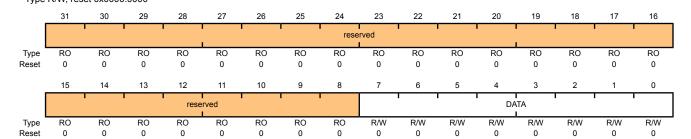
- 0 Disables the I²C slave operation.
- 1 Enables the I²C slave operation.

Register 12: I²C Slave Data (I2CSDR), offset 0x008

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 Slave 0 base: 0x4002.0800 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	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 0x00C

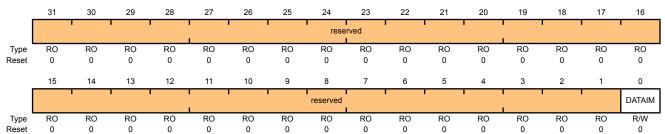
This register controls whether a raw interrupt is promoted to a controller interrupt.

I2C Slave Interrupt Mask (I2CSIMR)

I2C Slave 0 base: 0x4002.0800

Offset 0x00C

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 0x010

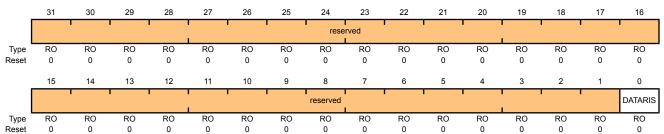
This register specifies whether an interrupt is pending.

I2C Slave Raw Interrupt Status (I2CSRIS)

I2C Slave 0 base: 0x4002.0800

Offset 0x010

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 0x014

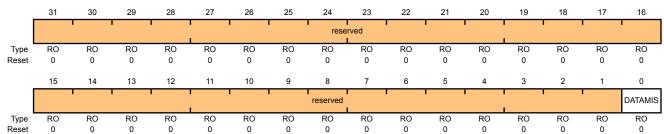
This register specifies whether an interrupt was signaled.

I2C Slave Masked Interrupt Status (I2CSMIS)

I2C Slave 0 base: 0x4002.0800

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	DATAMIS	RO	0	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 0x018

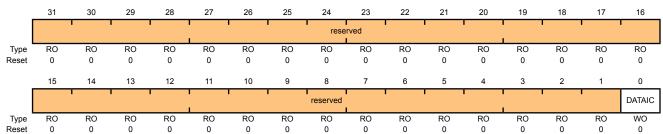
This register clears the raw interrupt. A read of this register returns no meaningful data.

I2C Slave Interrupt Clear (I2CSICR)

I2C Slave 0 base: 0x4002.0800

Offset 0x018

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	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 <code>DATARIS</code> interrupt bit; otherwise, it has no effect on the <code>DATARIS</code> bit value.

16 Controller Area Network (CAN) Module

16.1 Controller Area Network Overview

Controller Area Network (CAN) is a multicast shared serial bus standard for connecting electronic control units (ECUs). CAN was specifically designed to be robust in electromagnetically noisy environments and can utilize a differential balanced line like RS-485 or a more robust twisted-pair wire. Originally created for automotive purposes, it is also used in many embedded control applications (such as industrial and medical). Bit rates up to 1 Mbps are possible at network lengths below 40 meters. Decreased bit rates allow longer network distances (for example, 125 Kbps at 500 m).

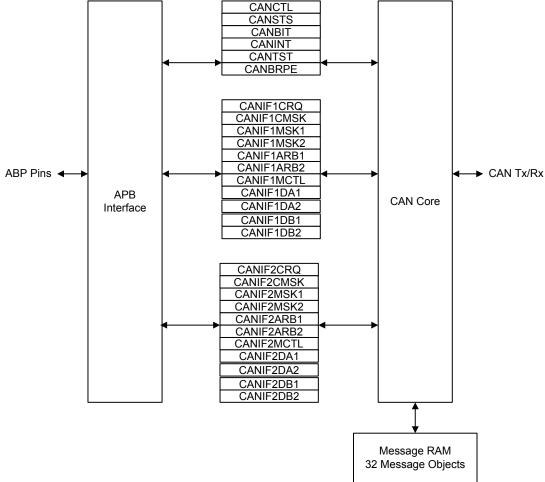
16.2 Controller Area Network Features

The Stellaris® CAN module supports the following features:

- CAN protocol version 2.0 part A/B
- Bit rates up to 1 Mbps
- 32 message objects
- Each message object has its own identifier mask
- Maskable interrupt
- Disable Automatic Retransmission mode for Time Triggered CAN (TTCAN) applications
- Programmable Loopback mode for self-test operation
- Programmable FIFO mode
- Gluelessly attachable to an external CAN PHY through the CANOTx and CANORx pins

16.3 **Controller Area Network Block Diagram**

Figure 16-1. CAN Module Block Diagram



16.4 **Controller Area Network Functional Description**

The CAN module conforms to the CAN protocol version 2.0 (parts A and B). Message transfers that include data, remote, error, and overload frames with an 11-bit identifier (standard) or a 29-bit identifier (extended) are supported. Transfer rates can be programmed up to 1 Mbps.

The CAN module consists of three major parts:

- CAN protocol controller and message handler
- Message memory
- CAN register interface

The protocol controller transfers and receives the serial data from the CAN bus and passes the data on to the message handler. The message handler then loads this information into the appropriate message object based on the current filtering and identifiers in the message object memory. The message handler is also responsible for generating interrupts based on events on the CAN bus.

The message object memory is a set of 32 identical memory blocks that hold the current configuration, status, and actual data for each message object. These are accessed via the CAN message object register interface. The message memory is not directly accessable in the Stellaris[®] memory map, so the Stellaris[®] CAN controller provides an interface to communicate with the message memory.

The CAN message object register interface provides two register sets for communicating with the message objects. Since there is no direct access to the message object memory, these two interfaces must be used to read or write to each message object. The two message object interfaces allow parallel access to the CAN controller message objects when multiple objects may have new information that needs to be processed.

16.4.1 Initialization

The software initialization is started by setting the INIT bit in the **CAN Control (CANCTL)** register (with software or by a hardware reset) or by going bus-off, which occurs when the transmitter's error counter exceeds a count of 255. While INIT is set, all message transfers to and from the CAN bus are stopped and the status of the CAN transmit output is recessive (High). Entering the initialization state does not change the configuration of the CAN controller, the message objects, or the error counters. However, some configuration registers are only accessible when in the initialization state.

To initialize the CAN controller, set the **CAN Bit Timing (CANBIT)** register and configure each message object. If a message object is not needed, it is sufficient to set it as not valid by clearing the MsgVal bit in the **CANIFnARB2** register. Otherwise, the whole message object has to be initialized, as the fields of the message object may not have valid information, causing unexpected results. Access to the **CAN Bit Timing (CANBIT)** register and to the **CAN Baud Rate Prescalar Extension (CANBRPE)** register to configure the bit timing is enabled when both the INIT and CCE bits in the **CANCTL** register are set. To leave the initialization state, the INIT bit must be cleared. Afterwards, the internal Bit Stream Processor (BSP) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (Bus Idle) before it takes part in bus activities and starts message transfers. The initialization of the message objects is independent of being in the initialization state and can be done on the fly, but message objects should all be configured to particular identifiers or set to not valid before the BSP starts the message transfer. To change the configuration of a message object during normal operation, set the MsgVal bit in the **CANIFnARB2** register to 0 (not valid). When the configuration is completed, MsgVal is set to 1 again (valid).

16.4.2 Operation

Once the CAN module is initialized and the INIT bit in the **CANCTL** register is reset to 0, the CAN module synchronizes itself to the CAN bus and starts the message transfer. As messages are received, they are stored in their appropriate message objects if they pass the message handler's filtering. The whole message (including all arbitration bits, data-length code, and eight data bytes) is stored in the message object. If the Identifier Mask (the Msk bits in the **CANIFnMSKn** registers) is used, the arbitration bits that are masked to "don't care" may be overwritten in the message object.

The CPU may read or write each message at any time via the CAN Interface Registers (CANIFnCRQ, CANIFnCMSK, CANIFnMSKn, CANIFnARBn, CANIFnMCTL, CANIFnDAn, and CANIFnDBn). The message handler guarantees data consistency in case of concurrent accesses.

The transmission of message objects is under the control of the software that is managing the CAN hardware. These can be message objects used for one-time data transfers, or permanent message objects used to respond in a more periodic manner. Permanent message objects have all arbitration and control set up, and only the data bytes are updated. To start the transmission, the \mathtt{TxRqst} bit in the **CANTXRQn** register and the \mathtt{NewDat} bit in the **CANNWDAn** register are set. If several transmit messages are assigned to the same message object (when the number of message objects is not

sufficient), the whole message object has to be configured before the transmission of this message is requested.

The transmission of any number of message objects may be requested at the same time; they are transmitted according to their internal priority, which is based on the message identifier for the message object. Messages may be updated or set to not valid any time, even when their requested transmission is still pending. The old data is discarded when a message is updated before its pending transmission has started. Depending on the configuration of the message object, the transmission of a message may be requested autonomously by the reception of a remote frame with a matching identifier.

There are two sets of CAN Interface Registers (**CANIF1x** and **CANIF2x**), which are used to access the Message Objects in the Message RAM. The CAN controller coordinates transfers to and from the Message RAM to and from the registers. The function of the two sets are independent and identical and can be used to queue transactions.

16.4.3 Transmitting Message Objects

If the internal transmit shift register of the CAN module is ready for loading, and if there is no data transfer between the CAN Interface Registers and message RAM, the valid message object with the highest priority that has a pending transmission request is loaded into the transmit shift register by the message handler and the transmission is started. The message object's NewDat bit is reset and can be viewed in the CANNWDAn register. After a successful transmission, and if no new data was written to the message object since the start of the transmission, the TxRqst bit in the CANIFnCMSK register is reset. If the TxIE bit in the CANIFnMCTL register is set, the IntPnd bit in the CANIFnMCTL register is set after a successful transmission. If the CAN module has lost the arbitration or if an error occurred during the transmission, the message is re-transmitted as soon as the CAN bus is free again. If, meanwhile, the transmission of a message with higher priority has been requested, the messages are transmitted in the order of their priority.

16.4.4 Configuring a Transmit Message Object

Table 16-1 on page 419 specifies the bit settings for a transmit message object.

Table 16-1. Transmit Message Object Bit Settings

Register	CANIFnARB2	CANIFnCMSK		IIFnCMSK CANIFnMCTL CANIFnARB2			CANIFnMCTL						
Bit	MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
Value	1	appl	appl	appl	1	1	0	0	0	appl	0	appl	0

The Xtd and ID bit fields in the **CANIFnARBn** registers are set by an application. They define the identifier and type of the outgoing message. If an 11-bit Identifier (Standard Frame) is used, it is programmed to bits [12:2] of **CANIFnARB2**, and the remaining identifier bits are not used by the CAN controller.

If the TxIE bit is set, the IntPnd bit is set after a successful transmission of the message object.

When the RmtEn bit is set, a matching received remote frame causes the TxRqst bit to be set and the message object automatically transfers the message object's data or generates an interrupt indicating a remote frame was requested. This can be strictly a single message identifier or it can be a range of values specified in the message object. The CAN mask registers, CANIFnMSKn, configure which groups of frames are identified as remote frame requests. The UMask bit in the CANIFnMCTL register enables the Msk bits in the CANIFnMSKn register to filter which frames are identified as a remote frame request. The MXtd bit should be set if only 29-bit extended identifiers should trigger a remote frame request.

The DLC bit in the **CANIFnMCTL** register is set to the number of bytes to transfer to the message object. TxRqst and RmtEn should not be set before the data is valid, as the current data in the message object can be transmitted as soon as these bits are set.

16.4.5 Updating a Transmit Message Object

The CPU may update the data bytes of a Transmit Message Object any time via the CAN Interface Registers and neither the MsqVal nor the TxRqst bits have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes of the corresponding **CANIFnDAn** or **CANIFnDBn** register have to be valid before the content of that register is transferred to the message object. Either the CPU has to write all four bytes into the **CANIFnDAn** or **CANIFnDBn** register or the message object is transferred to the **CANIFnDAn** or **CANIFnDBn** register before the CPU writes the new data bytes.

In order to only update the data in a message object, the WR, NewDat, DataA, and DataB bits are written to the CAN IFn Command Mask (CANIFnMSKn) register, followed by writing the CAN IFn Data registers, and then the number of the message object is written to the CAN IFn Command Request (CANIFnCRQ) register, to update the data bytes and the TxRqst bit at the same time.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst. When NewDat is set together with TxRqst, NewDat is reset as soon as the new transmission has started.

16.4.6 Accepting Received Message Objects

When the arbitration and control field (ID + Xtd + RmtEn + DLC) of an incoming message is completely shifted into the CAN module, the message handling capability of the module starts scanning the message RAM for a matching valid message object. To scan the message RAM for a matching message object, the Acceptance Filtering unit is loaded with the arbitration bits from the core. Then the arbitration and mask fields (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the Acceptance Filtering unit and compared with the arbitration field from the shift register. This is repeated with each following message object until a matching message object is found or until the end of the message RAM is reached. If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of frame received.

16.4.7 Receiving a Data Frame

The message handler stores the message from the CAN module receive shift register into the respective message object in the message RAM. It stores the data bytes, all arbitration bits, and the Data Length Code into the corresponding message object. This is implemented to keep the data bytes connected with the identifier even if arbitration mask registers are used. The NewDat bit of the CANIFnMCTL register is set to indicate that new data has been received. The CPU should reset this bit when it reads the message object to indicate to the controller that the message has been received and the buffer is free to receive more messages. If the CAN controller receives a message and the NewDat bit was already set, the MsgLst bit is set to indicate that the previous data was lost. If the RxIE bit of the CANIFnMCTL register is set, the IntPnd bit of the same register is set, causing the CANINT interrupt register to point to the message object that just received a message. The TxRqst bit of this message object should be cleared to prevent the transmission of a remote frame.

16.4.8 Receiving a Remote Frame

When a remote frame is received, three different configurations of the matching message object have to be considered:

Configuration	Description
Dir = 1 (direction = transmit)	At the reception of a matching remote frame, the TxRqst bit of this message object is set.
RmtEn = 1	The rest of the message object remains unchanged, and the controller will transfer the data in the message object.
UMask = 1 or 0	
Dir = 1 (direction = transmit)	At the reception of a matching remote frame, the TxRqst bit of this message object remains
RmtEn = 0	unchanged; the remote frame is ignored. This remote frame is disabled and will not automatically respond or indicate that the remote frame ever happened.
UMask = 0	
Dir = 1 (direction = transmit)	At the reception of a matching remote frame, the TxRqst bit of this message object is reset.
RmtEn = 0	The arbitration and control field (ID + Xtd + RmtEn + DLC) from the shift register is stored into the message object in the message RAM and the NewDat bit of this message object is
UMask = 1	set. The data field of the message object remains unchanged; the remote frame is treated
	similar to a received data frame. This is useful for a remote data request from another CAN device for which the Stellaris [®] controller does not have readily available data. The software
	must fill the data and answer the frame manually.

16.4.9 Receive/Transmit Priority

The receive/transmit priority for the message objects is controlled by the message number. Message object 1 has the highest priority, while message object 32 has the lowest priority. If more than one transmission request is pending, the message objects are transmitted in order based on the message object with the lowest message number. This should not be confused with the message identifier as that priority is enforced by the CAN bus. This means that if message object 1 and message object 2 both have valid messages that need to be transmitted, message object 1 will always be transmitted first regardless of the message identifier in the message object itself.

16.4.10 Configuring a Receive Message Object

Table 16-2 on page 421 specifies the bit settings for a transmit message object.

Table 16-2. Receive Message Object Bit Settings

Register	CANIFnARB2	CAI	NIFnC	MSK	CANIFnMCTL	CANIFnARB2	CANIFnMCTL						
Bit	MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
Value	1	appl	appl	appl	1	0	0	0	appl	0	0	0	0

The Xtd and ID bit fields in the **CANIFnARBn** registers are set by an application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (Standard Frame) is used, it is programmed to bits [12:2] of **CANIFnARB2**, and the remaining identifier bits are ignored by the CAN controller. When a data frame with an 11-bit Identifier is received, only bits 12:2 of **CANIFnARB2** are valid and the rest are set to 0.

If the RXIE bit is set, the IntPnd bit is set when a received data frame is accepted and stored in the message object.

When the message handler stores a data frame in the message object, it stores the received Data Length Code and eight data bytes. If the Data Length Code is less than 8, the remaining bytes of the message object are overwritten by nonspecified values.

The CAN mask registers can be used to allow groups of data frames to be received by a message object. The CAN mask registers, **CANIFnMSKn**, configure which groups of frames are received by a message object. The UMask bit in the **CANIFnMCTL** register enables the Msk bits in the **CANIFnMSKn** register to filter which frames are received. The Mxtd bit should be set if only 29-bit extended identifiers should be received by this message object.

16.4.11 Handling of Received Message Objects

The CPU may read a received message any time via the CAN Interface registers because the data consistency is guaranteed by the message handler state machine.

Typically, the CPU first writes 0x007F to the CAN IFn Command Mask (CANIFnCMSK) register and then writes the number of the message object to the CAN IFn Command Request (CANIFnCRQ) register. That combination transfers the whole received message from the message RAM into the Message Buffer registers (CANIFnMSKn, CANIFnARBn, and CANIFnMCTL). Additionally, the NewDat and IntPnd bits are cleared in the message RAM, acknowledging that the message has been read and clearing the pending interrupt being generated by this message object.

If the message object uses masks for acceptance filtering, the arbitration bits show which of the matching messages has been received.

The actual value of NewDat shows whether a new message has been received since the last time this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time this message object was read. MsgLst is not automatically reset.

Using a remote frame, the CPU may request new data from another CAN node on the CAN bus. Setting the \mathtt{TxRqst} bit of a receive object causes the transmission of a remote frame with the receive object's identifier. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the \mathtt{TxRqst} bit is automatically reset. This prevents the possible loss of data when the other device on the CAN bus has already transmitted the data slightly earlier than expected.

16.4.12 Handling of Interrupts

If several interrupts are pending, the **CAN Interrupt (CANINT)** register points to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the CPU has cleared it.

The Status Interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. A message interrupt is cleared by clearing the message object's IntPnd bit. The Status Interrupt is cleared by reading the **CAN Status** (**CANSTS**) register.

The interrupt identifier IntId in the **CANINT** register indicates the cause of the interrupt. When no interrupt is pending, the register holds the value to 0. If the value of **CANINT** is different from 0, then there is an interrupt pending. If the IE bit is set in the **CANCTL** register, the interrupt line to the CPU is active. The interrupt line remains active until **CANINT** is 0, all interrupt sources have been cleared (the cause of the interrupt is reset), or until IE is reset, which disables interrupts from the CAN controller.

The value 0x8000 in the **CANINT** register indicates that an interrupt is pending because the CAN module has updated, but not necessarily changed, the **CANSTS** register (Error Interrupt or Status Interrupt). This indicates that there is either a new Error Interrupt or a new Status Interrupt. A write access can clear the RxOK, TxOK, and LEC flags in the **CANSTS** register, however, only a read access to the **CANSTS** register will clear the source of the Status Interrupt.

IntId points to the pending message interrupt with the highest interrupt priority. The SIE bit in the **CANCTL** register controls whether a change of the status register may cause an interrupt. The EIE bit in the **CANCTL** register controls whether any interrupt from the CAN controller actually generates an interrupt to the microcontroller's interrupt controller. The **CANINT** interrupt register is updated even when the IE bit is set to zero.

There are two possibilities when handling the source of a message interrupt. The first is to read the IntId bit in the **CANINT** interrupt register to determine the highest priority interrupt that is pending, and the second is to read the **CAN Message Interrupt Pending (CANMSGnINT)** register to see all of the message objects that have pending interrupts.

An interrupt service routine reading the message that is the source of the interrupt may read the message and reset the message object's IntPnd at the same time by setting the ClrIntPnd bit in the CAN IFn Command Mask (CANIFnCMSK) register. When the IntPnd bit is cleared, the CANINT register will contain the message number for the next message object with a pending interrupt.

16.4.13 Bit Timing Configuration Error Considerations

Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly. In many cases, the CAN bit synchronization amends a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration, however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive. The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

16.4.14 Bit Time and Bit Rate

The CAN system supports bit rates in the range of lower than 1 Kbps up to 1000 Kbps. Each member of the CAN network has its own clock generator. The timing parameter of the bit time can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods may be different.

Because of small variations in frequency caused by changes in temperature or voltage and by deteriorating components, these oscillators are not absolutely stable. As long as the variations remain inside a specific oscillator's tolerance range, the CAN nodes are able to compensate for the different bit rates by periodically resynchronizing to the bit stream.

According to the CAN specification, the bit time is divided into four segments (see Figure 16-2 on page 424): the Synchronization Segment, the Propagation Time Segment, the Phase Buffer Segment 1, and the Phase Buffer Segment 2. Each segment consists of a specific, programmable number of time quanta (see Table 16-3 on page 424). The length of the time quantum (t_q), which is the basic time unit of the bit time, is defined by the CAN controller's system clock (fsys) and the Baud Rate Prescaler (graphicolor):

$$t_{\alpha} = BRP / fsys$$

The CAN module's system clock fsys is the frequency of its CAN module clock input.

The Synchronization Segment Sync_Seg is that part of the bit time where edges of the CAN bus level are expected to occur; the distance between an edge that occurs outside of Sync_Seg and the Sync Seg is called the *phase error* of that edge.

The Propagation Time Segment Prop_Seg is intended to compensate for the physical delay times within the CAN network.

The Phase Buffer Segments Phase_Seg1 and Phase_Seg2 surround the Sample Point.

The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

A given bit rate may be met by different bit-time configurations, but for the proper function of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

Figure 16-2. CAN Bit Time

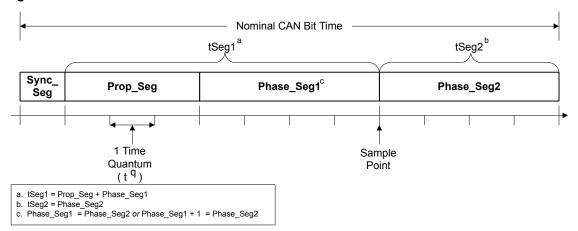


Table 16-3. CAN Protocol Ranges^a

Parameter	Range	Remark
BRP	[1 32]	Defines the length of the time quantum t _q
Sync_Seg	1 t _q	Fixed length, synchronization of bus input to system clock
Prop_Seg	[1 8] t _q	Compensates for the physical delay times
Phase_Seg1	[1 8] t _q	May be lengthened temporarily by synchronization
Phase_Seg2	[1 8] t _q	May be shortened temporarily by synchronization
SJW	[1 4] t _q	May not be longer than either Phase Buffer Segment

a. This table describes the minimum programmable ranges required by the CAN protocol.

The bit timing configuration is programmed in two register bytes in the **CANBIT** register. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, and SJW and BRP are combined in the other byte.

In these bit timing registers, the four components TSEG1, TSEG2, SJW, and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1..n], values in the range of [0..n-1] are programmed. That way, for example, SJW (functional range of [1..4]) is represented by only two bits. Therefore, the length of the bit time is (programmed values):

[TSEG1 + TSEG2 + 3]
$$\times$$
 t_a

or (functional values):

[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2]
$$\times t_q$$

The data in the bit timing registers are the configuration input of the CAN protocol controller. The Baud Rate Prescalar (configured by BRP) defines the length of the time quantum, the basic time unit of the bit time; the Bit Timing Logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the Sample Point, and occasional synchronizations are controlled by the CAN controller and are evaluated once per time quantum.

The CAN controller translates messages to and from frames. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the Sample Point and processes the sampled bus input bit. The time after the Sample Point that is needed to calculate the next bit to be sent (that is, the data bit, CRC bit, stuff bit, error flag, or idle) is called the Information Processing Time (IPT).

The IPT is application-specific but may not be longer than 2 t_q ; the CAN's IPT is 0 t_q . Its length is the lower limit of the programmed length of Phase_Seg2. In case of synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

16.4.15 Calculating the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1/bit rate) must be an integer multiple of the system clock period.

The bit time may consist of 4 to 25 time quanta. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

The first part of the bit time to be defined is the $Prop_Seg$. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandable CAN bus systems. The resulting time for $Prop_Seg$ is converted into time quanta (rounded up to the nearest integer multiple of tg).

The $Sync_Seg$ is 1 t_q long (fixed), which leaves (bit time - $Prop_Seg$ - 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, that is, $Phase_Seg2$ = $Phase_Seg1$, else $Phase_Seg2$ = $Phase_Seg1$ + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than the CAN controller's Information Processing Time, which is, depending on the actual implementation, in the range of $[0..2] t_n$.

The length of the Synchronization Jump Width is set to its maximum value, which is the minimum of 4 and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formula given below:

```
(1 - df) \times fnom <= fosc <= (1 + df) \times fnom
```

where:

- df = Maximum tolerance of oscillator frequency
- fosc = Actual oscillator frequency
- fnom = Nominal oscillator frequency

Maximum frequency tolerance must take into account the following formulas:

```
df <= (Phase_Seg1,Phase_Seg2)min/ 2 × (13 × tbit - Phase_Seg2) dfmax = 2 \times df \times fnom
```

where:

Phase_Seg1 and Phase_Seg2 are from Table 16-3 on page 424

- tbit = Bit Time
- dfmax = Maximum difference between two oscillators

If more than one configuration is possible, that configuration allowing the highest oscillator tolerance range should be chosen.

CAN nodes with different system clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol-compliant configuration of the CAN bit timing.

The resulting configuration is written into the CAN Bit Timing (CANBIT) register :

```
(Phase_Seg2-1)&(Phase_Seg1+Prop_Seg-1)&(SynchronizationJumpWidth-1)&(Prescaler-1)
```

16.4.15.1 Example for Bit Timing at High Baud Rate

In this example, the frequency of CAN clock is 25 MHz, BRP is 0, and the bit rate is 1 Mbps.

```
t_q 40 ns = 1/((BRP + 1) × CAN Clock) delay of bus driver 50 ns delay of receiver circuit 30 ns delay of bus line (40m) 220 ns tProp 640 ns = 16 × t_q tSJW 160 ns = 4 × t_q tTSeg1 800 ns = tProp + tSJW tTSeg2 160 ns = Information Processing Time + 4 × t_q tSync-Seg 40 ns = 1 × t_q bit time 1000 ns = tSync-Seg + tTSeg1 + tTSeg2 tolerance for CAN_CLK 0.39 % = min(PB1,PB2)/ 2 × (13 x bit time - PB2) = 0.1us/ 2 x (13x 1us - 2us)
```

In the above example, the parameters for the **CANBIT** register are: TSeg2=3, TSeg1=15, SJW =3 and BRP=0. This makes the final value programmed into the **CANBIT** register, 0x3FC0.

16.4.15.2 Example for Bit Timing at Low Baud Rate

In this example, the frequency of CAN clock is 50 MHz, BRP is 25, and the bit rate is 100 Kbps.

```
t_q 500 ns = 1/((BRP + 1) × CAN clock) delay of bus driver 200 ns delay of receiver circuit 80 ns delay of bus line (40m) 220 ns tProp 4.5 ms = 9 × t_q tSJW 2 ms = 4 × t_q tTSeg1 6.5 ms = tProp + tSJW tTSeg2 3 ms = Information Processing Time + 6 × t_q tSync-Seg 500 ns = 1 × t_q bit time 10 ms = tSync-Seg + tTSeg1 + tTSeg2
```

```
tolerance for CAN_CLK 1.58 % =
  min(PB1,PB2)/ 2 x (13 x bit time - PB2) =
  4us/ 2 x (13 x 10us - 4us)
```

In this example, the concatenated bit time parameters are (4-1)3&(5-1)4&(4-1)2&(2-1)6, and **CANBIT** is programmed to 0x34C1.

In the above example, the parameters for the **CANBIT** register are: TSeg2=5, TSeg1=12, SJW =3 and BRP=24. This makes the final value programmed into the **CANBIT** register, 0x5CD8.

16.5 Controller Area Network Register Map

Table 16-4 on page 427 lists the registers. All addresses given are relative to the CAN base address of:

CAN0: 0x4004.0000

Table 16-4. CAN Register Map

Offset	Name	Туре	Reset	Description	See page
0x000	CANCTL	R/W	0x0000.0001	CAN Control	429
0x004	CANSTS	R/W	0x0000.0000	CAN Status	431
800x0	CANERR	RO	0x0000.0000	CAN Error Counter	434
0x00C	CANBIT	R/W	0x0000.2301	CAN Bit Timing	435
0x010	CANINT	RO	0x0000.0000	CAN Interrupt	437
0x014	CANTST	R/W	0x0000.0000	CAN Test	438
0x018	CANBRPE	R/W	0x0000.0000	CAN Baud Rate Prescalar Extension	440
0x020	CANIF1CRQ	R/W	0x0000.0001	CAN IF1 Command Request	441
0x024	CANIF1CMSK	R/W	0x0000.0000	CAN IF1 Command Mask	442
0x028	CANIF1MSK1	R/W	0x0000.FFFF	CAN IF1 Mask 1	445
0x02C	CANIF1MSK2	R/W	0x0000.FFFF	CAN IF1 Mask 2	446
0x030	CANIF1ARB1	R/W	0x0000.0000	CAN IF1 Arbitration 1	447
0x034	CANIF1ARB2	R/W	0x0000.0000	CAN IF1 Arbitration 2	448
0x038	CANIF1MCTL	R/W	0x0000.0000	CAN IF1 Message Control	450
0x03C	CANIF1DA1	R/W	0x0000.0000	CAN IF1 Data A1	452
0x040	CANIF1DA2	R/W	0x0000.0000	CAN IF1 Data A2	452
0x044	CANIF1DB1	R/W	0x0000.0000	CAN IF1 Data B1	452
0x048	CANIF1DB2	R/W	0x0000.0000	CAN IF1 Data B2	452
0x080	CANIF2CRQ	R/W	0x0000.0001	CAN IF2 Command Request	441
0x084	CANIF2CMSK	R/W	0x0000.0000	CAN IF2 Command Mask	442
0x088	CANIF2MSK1	R/W	0x0000.FFFF	CAN IF2 Mask 1	445

Offset	Name	Type	Reset	Description	See page
0x08C	CANIF2MSK2	R/W	0x0000.FFFF	CAN IF2 Mask 2	446
0x090	CANIF2ARB1	R/W	0x0000.0000	CAN IF2 Arbitration 1	447
0x094	CANIF2ARB2	R/W	0x0000.0000	CAN IF2 Arbitration 2	448
0x098	CANIF2MCTL	R/W	0x0000.0000	CAN IF2 Message Control	450
0x09C	CANIF2DA1	R/W	0x0000.0000	CAN IF2 Data A1	452
0x0A0	CANIF2DA2	R/W	0x0000.0000	CAN IF2 Data A2	452
0x0A4	CANIF2DB1	R/W	0x0000.0000	CAN IF2 Data B1	452
0x0A8	CANIF2DB2	R/W	0x0000.0000	CAN IF2 Data B2	452
0x100	CANTXRQ1	RO	0x0000.0000	CAN Transmission Request 1	453
0x104	CANTXRQ2	RO	0x0000.0000	CAN Transmission Request 2	453
0x120	CANNWDA1	RO	0x0000.0000	CAN New Data 1	454
0x124	CANNWDA2	RO	0x0000.0000	CAN New Data 2	454
0x140	CANMSG1INT	RO	0x0000.0000	CAN Message 1 Interrupt Pending	455
0x144	CANMSG2INT	RO	0x0000.0000	CAN Message 2 Interrupt Pending	455
0x160	CANMSG1VAL	RO	0x0000.0000	CAN Message 1 Valid	456
0x164	CANMSG2VAL	RO	0x0000.0000	CAN Message 2 Valid	456

16.6 Register Descriptions

The remainder of this section lists and describes the CAN registers, in numerical order by address offset. There are two sets of Interface Registers that are used to access the Message Objects in the Message RAM: **CANIF1x** and **CANIF2x**. The function of the two sets are identical and are used to queue transactions.

Register 1: CAN Control (CANCTL), offset 0x000

This control register initializes the module and enables test mode and interrupts.

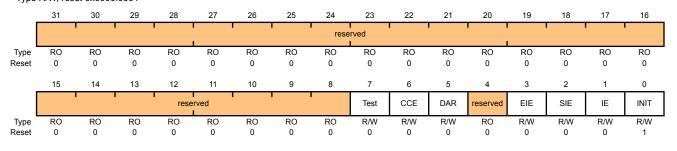
The bus-off recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting INIT. If the device goes bus-off, it sets INIT, stopping all bus activities. Once INIT has been cleared by the CPU, the device then waits for 129 occurrences of Bus Idle (129 * 11 consecutive High bits) before resuming normal operations. At the end of the bus-off recovery sequence, the Error Management Counters are reset.

During the waiting time after INIT is reset, each time a sequence of 11 High bits has been monitored, a BitOError code is written to the **CANSTS** status register, enabling the CPU to readily check whether the CAN bus is stuck Low or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.

CAN Control (CANCTL)

CAN0 base: 0x4004.0000

Offset 0x000 Type R/W, reset 0x0000.0001



Bit/Field	Name	Туре	Reset	Description
31:8	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.
7	Test	R/W	0	Test Mode Enable 0: Normal Operation
				1: Test Mode
6	CCE	R/W	0	Configuration Change Enable
				0: Do not allow write access to the CANBIT register.
				1: Allow write access to the CANBIT register if the INIT bit is 1.
5	DAR	R/W	0	Disable Automatic Retransmission
				0: Auto retransmission of disturbed messages is enabled.
				1: Auto retransmission is disabled.
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	EIE	R/W	0	Error Interrupt Enable
				0: Disabled. No Error Status interrupt is generated.
				1: Enabled. A change in the Boff or EWarn bits in the CANSTS register

generates an interrupt.

Bit/Field	Name	Туре	Reset	Description
2	SIE	R/W	0	Status Interrupt Enable
				0: Disabled. No Status interrupt is generated.
				1: Enabled. An interrupt is generated when a message has successfully been transmitted or received, or a CAN bus error has been detected. A change in the \mathtt{TxOK} , \mathtt{RxOK} or \mathtt{LEC} bits in the CANSTS register generates an interrupt.
1	ΙE	R/W	0	CAN Interrupt Enable
				0: Interrupts disabled.
				1: Interrupts enabled.
0	INIT	R/W	1	Initialization
				0: Normal operation.
				1: Initialization started.

Register 2: CAN Status (CANSTS), offset 0x004

The status register contains information for interrupt servicing such as Bus-Off, error count threshold, and error types.

The LEC field holds the code that indicates the type of the last error to occur on the CAN bus. This field is cleared to 0 when a message has been transferred (reception or transmission) without error. The unused error code 7 may be written by the CPU to manually set this field to an invalid error so that it can be checked for a change later.

An Error Interrupt is generated by the BOff and EWarn bits and a Status Interrupt is generated by the RXOK, TXOK, and LEC bits, assuming that the corresponding enable bits in the CAN Control (CANCTL) register are set. A change of the EPass bit or a write to the RXOK, TXOK, or LEC bits does not generate an interrupt.

Reading the CAN Status (CANSTS) register clears the CAN Interrupt (CANINT) register, if it is pending.

CAN Status (CANSTS)

CAN0 base: 0x4004.0000

Offset 0x004

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
Type Reset	0	0	0	NO n	0	ΛΟ	0	0	0	0	0	0	0	0	0	0
Neset	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
_	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		'		rese	rved		,		BOff	EWarn	EPass	RxOK	TxOK		LEC	
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	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
31:8	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.
7	BOff	RO	0	Bus-Off Status
				0: Module is not in bus-off state.
				1: Module is in bus-off state.
6	EWarn	RO	0	Warning Status
				0: Both error counters are below the error warning limit of 96.
				1: At least one of the error counters has reached the error warning limit of 96.
5	EPass	RO	0	Error Passive

0: The CAN module is in the Error Active state, that is, the receive or transmit error count is less than or equal to 127.

1: The CAN module is in the Error Passive state, that is, the receive or transmit error count is greater than 127.

Bit/Field	Name	Type	Reset	Description
4	RxOK	R/W	0	Received a Message Successfully
				0: Since this bit was last reset to 0, no message has been successfully received.
				1: Since this bit was last reset to 0, a message has been successfully received, independent of the result of the acceptance filtering.
				This bit is never reset by the CAN module.
3	TxOK	R/W	0	Transmitted a Message Successfully
				0: Since this bit was last reset to 0, no message has been successfully transmitted.
				1: Since this bit was last reset to 0, a message has been successfully transmitted error-free and acknowledged by at least one other node.
				This bit is never reset by the CAN module.

Bit/Field	Name	Type	Reset	Description
2:0	LEC	R/W	0x0	Last Error Code

This is the type of the last error to occur on the CAN bus.

Value Definition 0x0 No Error 0x1 Stuff Error

More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.

0x2 Format Error

A fixed format part of the received frame has the wrong format.

0x3 ACK Error

The message transmitted was not acknowledged by another node.

0x4 Bit 1 Error

When a message is transmitted, the CAN controller monitors the data lines to detect any conflicts. When the arbitration field is transmitted, data conflicts are a part of the arbitration protocol. When other frame fields are transmitted, data conflicts are considered errors.

A Bit 1 Error indicates that the device wanted to send a High level (logical 1) but the monitored bus value was Low (logical 0).

0x5 Bit 0 Error

A Bit 0 Error indicates that the device wanted to send a Low level (logical 0), but the monitored bus value was High (logical 1)

During bus-off recovery, this status is set each time a sequence of 11 High bits has been monitored. This enables the CPU to monitor the proceeding of the bus-off recovery sequence without any disturbances to the bus.

0x6 CRC Error

The CRC checksum was incorrect in the received message, indicating that the calculated value received did not match the calculated CRC of the data.

0x7 Unused

When the LEC bit shows this value, no CAN bus event was detected since the CPU wrote this value to LEC.

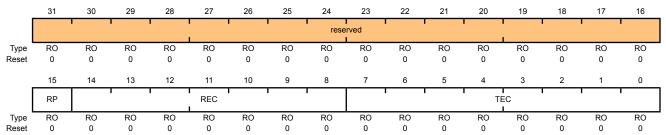
Register 3: CAN Error Counter (CANERR), offset 0x008

This register contains the error counter values, which can be used to analyze the cause of an error.

CAN Error Counter (CANERR)

CAN0 base: 0x4004.0000 Offset 0x008

Type RO, 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	RP	RO	0	Received Error Passive
				0: The Receive Error counter is below the Error Passive level (127 or less).
				1: The Receive Error counter has reached the Error Passive level (128 or greater).
14:8	REC	RO	0x0	Receive Error Counter
				State of the receiver error counter (0 to 127).
7:0	TEC	RO	0x0	Transmit Error Counter
				State of the transmit error counter (0 to 255).

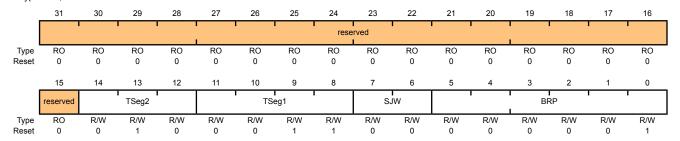
Register 4: CAN Bit Timing (CANBIT), offset 0x00C

This register is used to program the bit width and bit quantum. Values are to be programmed to the system clock frequency. This register is write-enabled by the CCE and INIT bits in the **CANCTL** register. See "Bit Time and Bit Rate" on page 423 for more information.

CAN Bit Timing (CANBIT)

CAN0 base: 0x4004.0000 Offset 0x00C

Type R/W, reset 0x0000.2301



Bit/Field	Name	Туре	Reset	Description
31:15	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.
14:12	TSeg2	R/W	0x2	Time Segment after Sample Point
				0x00-0x07: The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
				So, for example, a reset value of 0x2 defines that there is 3(2+1) bit time quanta defined for Phase_Seg2 (see Figure 16-2 on page 424). The bit time quanta is defined by BRP.
11:8	TSeg1	R/W	0x3	Time Segment Before Sample Point
				0x00-0x0F: The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
				So, for example, the reset value of 0x3 defines that there is 4(3+1) bit time quanta defined for $Phase_Seg1$ (see Figure 16-2 on page 424). The bit time quanta is define by BRP.
7:6	SJW	R/W	0x0	(Re)Synchronization Jump Width
				0x00-0x03: The actual interpretation by the hardware of this value is

0x00-0x03: The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

During the start of frame (SOF), if the CAN controller detects a phase error (misalignment), it can adjust the length of ${\tt TSeg2}$ or ${\tt TSeg1}$ by the value in ${\tt SJW}$. So the reset value of 0 adjusts the length by 1 bit time quanta.

Bit/Field	Name	Type	Reset	Description
5:0	BRP	R/W	0x1	Baud Rate Prescalar
				The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quantum.
				0x00-0x03F: The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
				${\tt BRP}$ defines the number of CAN clock periods that make up 1 bit time quanta, so the reset value is 2 bit time quanta (1+1).
				The CANBRPE register can be used to further divide the bit time.

Register 5: CAN Interrupt (CANINT), offset 0x010

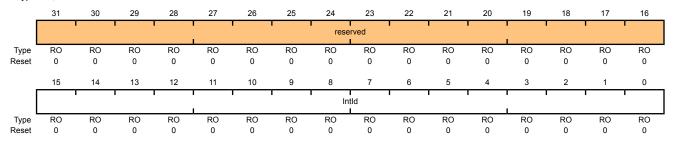
This register indicates the source of the interrupt.

If several interrupts are pending, the CAN Interrupt (CANINT) register points to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the CPU has cleared it. If the Intid bit is not 0x0000 (the default) and the IE bit in the CANCTL register is set, the interrupt is active. The interrupt line remains active until the IntId bit is set back to 0x0000 when the cause of all interrupts are reset, or until IE is reset.

Reading the CAN Status (CANSTS) register clears the CAN Interrupt (CANINT) register, if it is pending.

CAN Interrupt (CANINT)

CAN0 base: 0x4004.0000 Offset 0x010 Type RO, 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	Intld	RO	0x0000	Interrupt Identifier

The number in this field indicates the source of the interrupt.

Value Definition

0x0000 No interrupt pending

0x0001-0x0020 Number of the message object that caused the

interrupt

0x0021-0x7FFF Unused

0x8000 Status Interrupt

0x8001-0xFFFF Unused

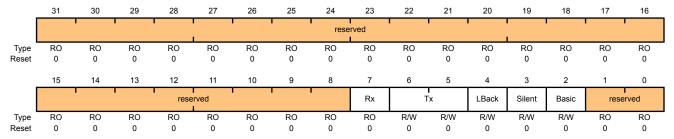
Register 6: CAN Test (CANTST), offset 0x014

This is the test mode register for self-test and external pin access. It is write-enabled by the Test bit in the CANCTL register. Different test functions may be combined, however, CAN transfers will be affected if the Tx bits in this register are not zero.

CAN Test (CANTST)

CAN0 base: 0x4004.0000

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



Bit/Field	Name	Туре	Reset	Description
31:8	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.
7	Rx	RO	0	Receive Observation
				Displays the value on the CANnRx pin.
6:5	Tx	R/W	0x0	Transmit Control
				Overrides control of the CANnTx pin.
				Value Description
				0x0 CANnTx is controlled by the CAN module
				0x1 Sample Point signal driven on the CANnTx pin
				0x2 CANnTx drives a Low value
				0x3 CANnTx drives a High value
4	LBack	R/W	0	Loopback Mode
				0: Disabled.
				1: Enabled.
3	Silent	R/W	0	Silent Mode
				Do not transmit data; monitor the bus. Also known as Bus Monitor mode.
				0: Disabled.
				1: Enabled.
2	Basic	R/W	0	Basic Mode
				0: Disabled.
				1: Use CANIF1 registers as transmit buffer, and use CANIF2 registers

as receive buffer.

Bit/Field	Name	Type	Reset	Description
1:0	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.

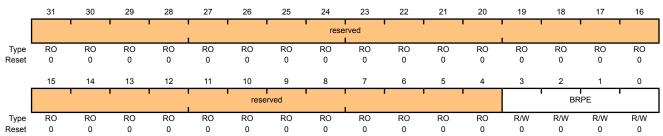
Register 7: CAN Baud Rate Prescalar Extension (CANBRPE), offset 0x018

This register is used to further divide the bit time set with the BRP bit in the CANBIT register. It is write-enabled with the \mathtt{CCE} bit in the **CANCTL** register.

CAN Baud Rate Prescalar Extension (CANBRPE)

CAN0 base: 0x4004.0000

Offset 0x018 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:4	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.
3:0	BRPE	R/W	0x0	Baud Rate Prescalar Extension

0x00-0x0F: Extend the BRP bit in the CANBIT register to values up to 1023. The actual interpretation by the hardware is one more than the value programmed by BRPE (MSBs) and BRP (LSBs).

Register 8: CAN IF1 Command Request (CANIF1CRQ), offset 0x020 Register 9: CAN IF2 Command Request (CANIF2CRQ), offset 0x080

This register is used to start a transfer when its MNUM bit field is updated. Its Busy bit indicates that the information is transferring from the CAN Interface Registers to the internal message RAM.

A message transfer is started as soon as there is a write of the message object number with the MNUM bit. With this write operation, the Busy bit is automatically set to 1 to indicate that a transfer is in progress. After a wait time of 3 to 6 CAN_CLK periods, the transfer between the interface register and the message RAM completes, which then sets the Busy bit back to 0.

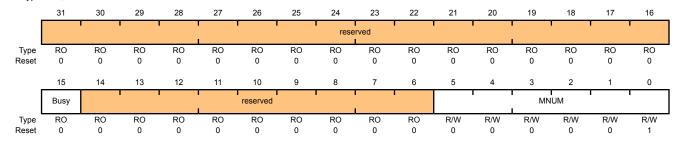
CAN IF1 Command Request (CANIF1CRQ)

CAN0 base: 0x4004.0000

Offset 0x020

D:4/E: -1-4

Type R/W, reset 0x0000.0001



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	Busy	RO	0x0	Busy Flag
				0: Reset when read/write action has finished.
				1: Set when a write occurs to the message number in this register.
14:6	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	MNUM	R/W	0x01	Message Number

D = = ==i=4:===

Selects one of the 32 message objects in the message RAM for data transfer. The message objects are numbered from 1 to 32.

Value Description

0x00 0 is not a valid message number; it is interpreted as 0x20,

or object 32.

0x01-0x20 Indicates specified message object 1 to 32.

 $\ensuremath{\text{0x21-0x3F}}$ Not a valid message number; values are shifted and it is

interpreted as 0x01-0x1F.

Register 10: CAN IF1 Command Mask (CANIF1CMSK), offset 0x024 Register 11: CAN IF2 Command Mask (CANIF2CMSK), offset 0x084

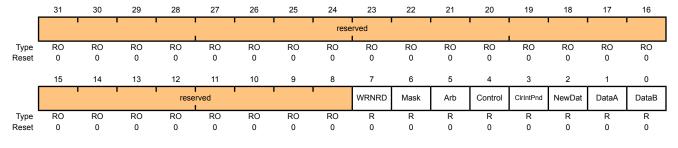
The Command Mask registers specify the transfer direction and select which buffer registers are the source or target of the data transfer.

Read-Only CANIFnCMSK Register

CAN IF1 Command Mask (CANIF1CMSK)

CAN0 base: 0x4004.0000

Offset 0x024 Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:8	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.
7	WRNRD	R	0	Write, Not Read
				Transfer the message object address specified by the CAN Command Request (CANIFnCRQ) register to the CAN message buffer registers (CANIFnMSK1, CANIFnMSK2, CANIFnARB1, CANIFnARB2, CANIFnCTL, CANIFnDA1, CANIFnDA2, CANIFnDB1, and CANIFnDB2).
6	Mask	R	0	Access Mask Bits
				0: Mask bits unchanged.
				1: Transfer ${\tt IDMask+Dir+MXtd}$ of the message object into the Interface registers.
5	Arb	R	0	Access Arbitration Bits
				0: Arbitration bits unchanged.
				1: Transfer ID + Dir + Xtd + MsgVal of the message object into the Interface registers.
4	Control	R	0	Access Control Bits
				0: Control bits unchanged.
				1: Transfer control bits into Interface registers.
3	ClrIntPnd	R	0	Clear Interrupt Pending Bit
				0: IntPnd bit in CANIFnMCTL register remains unchanged.
				1: Clear IntPnd bit in the CANIFnMCTL register in the message object.

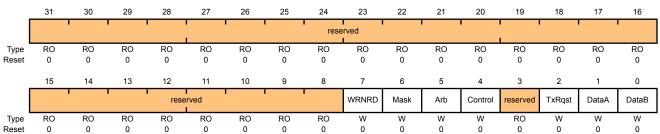
Bit/Field	Name	Type	Reset	Description
2	NewDat	R	0	Access New Data
				0: NewDat bit unchanged.
				1: Clear NewDat bit in the message object.
				Note: A read access to a message object can be combined with the reset of the control bits IntPdn and NewDat. The values of these bits that are transferred to the CANIFnMCTL register always reflect the status before resetting these bits.
1	DataA	R	0	Access Data Byte 0 to 3
				0: Data bytes 0-3 are unchanged.
				1: Transfer data bytes 0-3 in message object to CANIFnDA1 and CANIFnDA2 .
0	DataB	R	0	Access Data Byte 4 to 7
				0: Data bytes 4-7 unchanged.

CANIFnDB2.

Write-Only CANIFnCMSK Register

CAN IF1 Command Mask (CANIF1CMSK)

CAN0 base: 0x4004.0000 Offset 0x024 Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000	Software should not rely on the value of a reserved bit. To provide
0.1.0	.000.100		CACCC	compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7	WRNRD	W	0	Write, Not Read
				0: Read.
				1: Write. Transfer data from the message buffer registers to the message object address specified by the CANIFnCRQ register.
6	Mask	W	0	Access Mask Bits
				0: Mask bits unchanged.

1: Transfer IDMask + Dir + MXtd to message object.

1: Transfer data bytes 4-7 in message object to CANIFnDB1 and

Bit/Field	Name	Туре	Reset	Description			
5	Arb	W	0	Access Arbitration Bits			
				0: Arbitration bits unchanged.			
				1: Transfer ID + Dir + Xtd + MsgVal to message object.			
4	Control	W	0	Access Control Bits			
				0: Control bits unchanged.			
				1: Transfer control bits to message object.			
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	TxRqst	W	0	Access Transmission Request Bit			
				0: TxRqst bit unchanged.			
				1: Set TxRqst bit			
				Note: If a transmission is requested by programming this TxRqst bit, the parallel TxRqst in the CANIFnMCTL register is ignored.			
1	DataA	W	0	Access Data Byte 0 to 3			
				0: Data bytes 0-3 are unchanged.			
				1: Transfer data bytes 0-3 (CANIFnDA1 and CANIFnDA2) to message object.			
0	DataB	W	0	Access Data Byte 4 to 7			
				0: Data bytes 4-7 unchanged.			
				1: Transfer data bytes 4-7 (CANIFnDB1 and CANIFnDB2) to message object.			

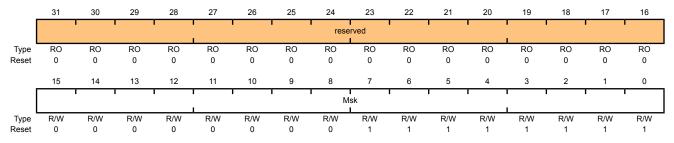
Register 12: CAN IF1 Mask 1 (CANIF1MSK1), offset 0x028 Register 13: CAN IF2 Mask 1 (CANIF2MSK1), offset 0x088

The mask information provided in this register accompanies the data (CANIFnDAn), arbitration information (CANIFnARBn), and control information (CANIFnMCTL) to the message object in the message RAM. The mask is used with the ID bit in the CANIFnARBn register for acceptance filtering. Additional mask information is contained in the CANIFnMSK2 register.

CAN IF1 Mask 1 (CANIF1MSK1)

CAN0 base: 0x4004.0000 Offset 0x028

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	Msk	R/W	0xFF	Identifier Mask

^{0:} The corresponding identifier bit (ID) in the message object cannot inhibit the match in acceptance filtering.

^{1:} The corresponding identifier bit (ID) is used for acceptance filtering.

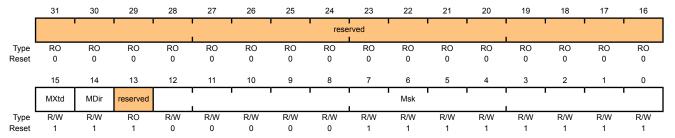
Register 14: CAN IF1 Mask 2 (CANIF1MSK2), offset 0x02C Register 15: CAN IF2 Mask 2 (CANIF2MSK2), offset 0x08C

This register holds extended mask information that accompanies the CANIFnMSK1 register.

CAN IF1 Mask 2 (CANIF1MSK2)

CAN0 base: 0x4004.0000 Offset 0x02C

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	MXtd	R/W	0x1	Mask Extended Identifier
				0: The extended identifier bit (Xtd in the CANIFnARB2 register) has no effect on the acceptance filtering.
				1: The extended identifier bit Xtd is used for acceptance filtering.
14	MDir	R/W	0x1	Mask Message Direction
				0: The message direction bit (Dir in the CANIFnARB2 register) has no effect for acceptance filtering.
				1: The message direction bit \mathtt{Dir} is used for acceptance filtering.
13	reserved	RO	0x1	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
12:0	Msk	R/W	0xFF	Identifier Mask

0: The corresponding identifier bit (${\tt ID}$) in the message object cannot inhibit the match in acceptance filtering.

1: The corresponding identifier bit (ID) is used for acceptance filtering.

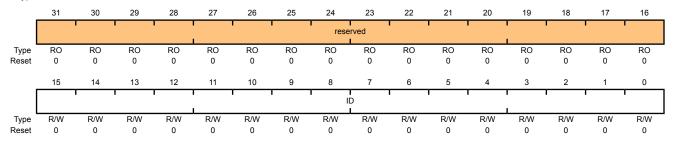
Register 16: CAN IF1 Arbitration 1 (CANIF1ARB1), offset 0x030 Register 17: CAN IF2 Arbitration 1 (CANIF2ARB1), offset 0x090

These registers hold the identifiers for acceptance filtering.

CAN IF1 Arbitration 1 (CANIF1ARB1)

CAN0 base: 0x4004.0000

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	ID	R/W	0x00	Message Identifier

This bit field is used with the ID field in the **CANIFnARB2** register to create the message identifier.

Bits 15:0 of the **CANIFnARB1** register are [15:0] of the ID, while bits 12:0 of the **CANIFnARB2** register are [28:16] of the ID.

If an 11-bit ID (Standard Frame) is used, ID[28:18] is used and ID[17:0] is disregarded (bits 15:0 of **CANIFnARB1** and bits 1:0 of **CANIFnARB2**).

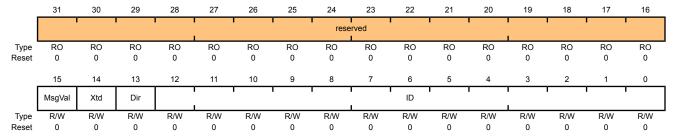
Register 18: CAN IF1 Arbitration 2 (CANIF1ARB2), offset 0x034 Register 19: CAN IF2 Arbitration 2 (CANIF2ARB2), offset 0x094

These registers hold information for acceptance filtering.

CAN IF1 Arbitration 2 (CANIF1ARB2)

CAN0 base: 0x4004.0000

Offset 0x034 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	MsgVal	R/W	0x0	Message Valid
				0: The message object is ignored by the message handler.
				1: The message object is configured and will be considered by the message handler within the CAN controller.
				All unused message objects should have this bit cleared during initialization and before clearing the Init bit in the CANCTL register. The MsgVal bit must also be cleared before any of the following bits are modified or if the message object is no longer required: the ID bit fields in the CANIFnARBn registers, the Xtd and Dir bits in the CANIFnARB2 register, or the DLC bits in the CANIFnMCTL register.
14	Xtd	R/W	0x0	Extended Identifier
				0: The 11-bit Standard Identifier will be used for this message object.
				1: The 29-bit Extended Identifier will be used for this message object.

Message Direction

- 0: Receive. On \mathtt{TxRqst} , a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, that message is stored in this message object.
- 1: Transmit. On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, TxRqst bit of this message object is set (if RmtEn=1).

13

Dir

R/W

0x0

Bit/Field	Name	Туре	Reset	Description
12:0	ID	R/W	0x0	Message Identifier
				This bit field is used with the ID field in the CANIFnARB2 register to create the message identifier.
				Bits 15:0 of the CANIFnARB1 register are [15:0] of the ID, while bits 12:0 of the CANIFnARB2 register are [28:16] of the ID.
				If an 11-bit ID (Standard Frame) is used, ID[28:18] is used and ID[17:0] is disregarded (bits 15:0 of CANIFnARB1 and bits 1:0 of CANIFnARB2).

Register 20: CAN IF1 Message Control (CANIF1MCTL), offset 0x038 Register 21: CAN IF2 Message Control (CANIF2MCTL), offset 0x098

This register holds the control information associated with the message object to be sent to the Message RAM.

23

CAN IF1 Message Control (CANIF1MCTL)

CAN0 base: 0x4004.0000

Offset 0x038

31

Type R/W, reset 0x0000.0000

							rese	rved							
RO 0	RO 0	RO	RO	RO 0	RO	RO	RO	RO	RO 0	RO 0	RO	RO 0	RO	RO	RO 0
			-										-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	EoB		reserved			DI	_C	·
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO 0	RO 0	RO	R/W	R/W	R/W	R/W 0
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
·· /=· · · ·				_			5								
sit/Field		Nam	ne	Iy	pe	Reset	Des	cription							
31:16		reserv	ved	R	0	0x0000	Soft	ware sho	ould not	rely on th	e value	of a rese	erved bit	. To prov	vide .
														ed bit sh	ould be
							pres	erved ac	cross a	read-modi	ify-write	operatio	n.		
15		New	Dat	R/	W	0x0	New	Data Data							
							0: N	o new da	ata has	been writt	en into	the data	portion o	of this m	essage
							obje	ct by the							•
							by tl	ne CPU.							
												as writte	n new d	ata into	the data
							port	ion of thi	s messa	age object	t.				
14		MsgL	₋st	R/	W	0x0	Mes	sage Lo	st						
							0:0	lo messa	age was	lost since	e the las	t time th	is bit wa	s reset b	y the
							CPL	J.							
														nis objec	t when
							New	Dat was	set; the	e CPU has	s lost a i	message) .		
							This	bit is on	ly valid	for messa	ige obje	cts with	the Dir	bit in the	•
							CAN	NFnARE	32 regis	ter set to (0 (receiv	/e).			
13		IntPr	nd	R/	W	0x0	Inte	rupt Per	nding						
	0 15 NewDat R/W 0 Bit/Field 31:16	0 0 15 14 NewDat MsgLst R/W 0 0 Bit/Field 31:16	0 0 0 15 14 13 NewDat MsgLst IntPnd R/W 0 0 0 Sit/Field Nam 31:16 reserved 15 NewI	0 0 0 0 0 15 14 13 12 12 NewDat MsgLst IntPnd UMask R/W 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 15 14 13 12 11 NewDat MsgLst IntPnd UMask TxIE R/W R/W R/W R/W R/W 0 0 0 0 0 3it/Field Name Tyl 31:16 reserved R/ 15 NewDat R/ R/ 14 MsgLst R/	0 RW RW RW RW RW RW 0 </td <td>0 0</td> <td> RO</td> <td>0 0</td> <td>RO RO R</td> <td>RO RO R</td> <td>RO RO R</td> <td>RO RO R</td> <td>RO RO R</td> <td>RO RO R</td>	0 0	RO	0 0	RO R	RO R	RO R	RO R	RO R	RO R

Use Acceptance Mask

0: Mask ignored.

priority.

0x0

R/W

UMask

1: Use mask (Msk, MXtd, and MDir) for acceptance filtering.

0: This message object is not the source of an interrupt.

1: This message object is the source of an interrupt. The interrupt identifier in the **CAN Interrupt (CANINT)** register will point to this message object if there is not another interrupt source with a higher

12

Bit/Field	Name	Туре	Reset	Description
11	TxIE	R/W	0x0	Transmit Interrupt Enable
				0: The IntPnd bit in the CANIFnMCTL register is unchanged after a successful transmission of a frame.
				1: The IntPnd bit in the CANIFnMCTL register is set after a successful transmission of a frame.
10	RxIE	R/W	0x0	Receive Interrupt Enable
				0: The IntPnd bit in the CANIFnMCTL register is unchanged after a successful reception of a frame.
				1: The IntPnd bit in the CANIFnMCTL register is set after a successful reception of a frame.
9	RmtEn	R/W	0x0	Remote Enable
				0: At the reception of a remote frame, the TxRqst bit in the CANIFnMCTL register is left unchanged.
				1: At the reception of a remote frame, the TxRqst bit in the CANIFnMCTL register is set.
8	TxRqst	R/W	0x0	Transmit Request
				0: This message object is not waiting for transmission.
				1: The transmission of this message object is requested and is not yet done.
7	EoB	R/W	0x0	End of Buffer
				0: Message object belongs to a FIFO Buffer and is not the last message object of that FIFO Buffer.
				1: Single message object or last message object of a FIFO Buffer.
				This bit is used to concatenate two or more message objects (up to 32) to build a FIFO buffer. For a single message object (thus not belonging to a FIFO buffer), this bit must be set to 1.
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.
3:0	DLC	R/W	0x0	Data Length Code
				Value Description
				0x0-0x8 Specifies the number of bytes in the data frame.
				0x9-0xF Defaults to a data frame with 8 bytes.
				The DLC bit in the CANIFnMCTL register of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it writes DLC to the value given by the received message.

Register 22: CAN IF1 Data A1 (CANIF1DA1), offset 0x03C

Register 23: CAN IF1 Data A2 (CANIF1DA2), offset 0x040

Register 24: CAN IF1 Data B1 (CANIF1DB1), offset 0x044

Register 25: CAN IF1 Data B2 (CANIF1DB2), offset 0x048

Register 26: CAN IF2 Data A1 (CANIF2DA1), offset 0x09C

Register 27: CAN IF2 Data A2 (CANIF2DA2), offset 0x0A0

Register 28: CAN IF2 Data B1 (CANIF2DB1), offset 0x0A4

Register 29: CAN IF2 Data B2 (CANIF2DB2), offset 0x0A8

These registers contain the data to be sent or that has been received. In a CAN data frame, data byte 0 is the first byte to be transmitted or received and data byte 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte is transmitted first.

CAN IF1 Data A1 (CANIF1DA1)

CAN0 base: 0x4004.0000

Offset 0x03C

Type R/W, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								rese	rved							
L																
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
ſ			1					1								
								Da	ata							
l.																
Type	R/W	R/W	R/W	R/W	R/W	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
31:16	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	0x00	Data

The **CANIFnDA1** registers contain data bytes 1 and 0; **CANIFnDA2** data bytes 3 and 2; **CANIFnDB1** data bytes 5 and 4; and **CANIFnDB2** data bytes 7 and 6.

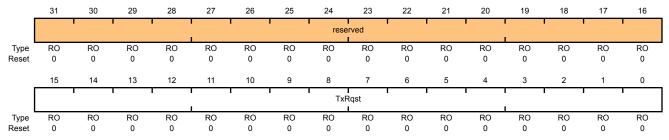
Register 30: CAN Transmission Request 1 (CANTXRQ1), offset 0x100 Register 31: CAN Transmission Request 2 (CANTXRQ2), offset 0x104

The **CANTXRQ1** and **CANTXRQ2** registers hold the \mathtt{TxRqst} bits of the 32 message objects. By reading out these bits, the CPU can check which message object has a transmission request pending. The \mathtt{TxRqst} bit of a specific message object can be changed by three sources: (1) the CPU via the **CAN IFn Message Control (CANIFnMCTL)** register, (2) the message handler state machine after the reception of a remote frame, or (3) the message handler state machine after a successful transmission.

The **CANTXRQ1** register contains the TxRqst bit of the first 16 message objects in the message RAM; the **CANTXRQ2** register contains the TxRqst bit of the second 16 message objects.

CAN Transmission Request 1 (CANTXRQ1)

CAN0 base: 0x4004.0000 Offset 0x100 Type RO, 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	TxRqst	RO	0x00	Transmission Request Bits

(of all message objects)

^{0:} The message object is not waiting for transmission.

^{1:} The transmission of the message object is requested and is not yet done.

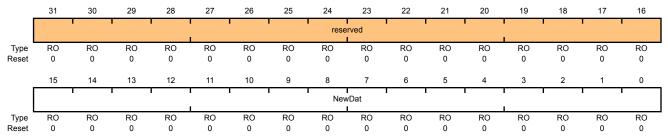
Register 32: CAN New Data 1 (CANNWDA1), offset 0x120 Register 33: CAN New Data 2 (CANNWDA2), offset 0x124

The **CANNWDA1** and **CANNWDA2** registers hold the NewDat bits of the 32 message objects. By reading these bits, the CPU can check which message object has its data portion updated. The NewDat bit of a specific message object can be changed by three sources: (1) the CPU via the **CAN IFn Message Control (CANIFnMCTL)** register, (2) the message handler state machine after the reception of a data frame, or (3) the message handler state machine after a successful transmission.

The **CANNWDA1** register contains the NewDat bit of the first 16 message objects in the message RAM; the **CANNWDA2** register contains the NewDat bit of the second 16 message objects.

CAN New Data 1 (CANNWDA1)

CAN0 base: 0x4004.0000 Offset 0x120 Type RO, 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	NewDat	RO	0x00	New Data Bits

(of all message objects)

^{0:} No new data has been written into the data portion of this message object by the message handler since the last time this flag was cleared by the CPU.

^{1:} The message handler or the CPU has written new data into the data portion of this message object.

Register 34: CAN Message 1 Interrupt Pending (CANMSG1INT), offset 0x140 Register 35: CAN Message 2 Interrupt Pending (CANMSG2INT), offset 0x144

The **CANMSG1INT** and **CANMSG2INT** registers hold the IntPnd bits of the 32 message objects. By reading these bits, the CPU can check which message object has an interrupt pending. The IntPnd bit of a specific message object can be changed through two sources: (1) the CPU via the **CAN IFN Message Control (CANIFNMCTL)** register, or (2) the message handler state machine after the reception or transmission of a frame.

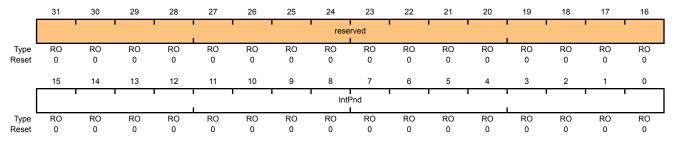
This field is also encoded in the CAN Interrupt (CANINT) register.

The **CANMSG1INT** register contains the IntPnd bit of the first 16 message objects in the message RAM; the **CANMSG2INT** register contains the IntPnd bit of the second 16 message objects.

CAN Message 1 Interrupt Pending (CANMSG1INT)

CAN0 base: 0x4004.0000 Offset 0x140

Type RO, 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	IntPnd	RO	0x00	Interrupt Pending Bits

(of all message objects)

0: This message object is not the source of an interrupt.

1: This message object is the source of an interrupt.

Register 36: CAN Message 1 Valid (CANMSG1VAL), offset 0x160 Register 37: CAN Message 2 Valid (CANMSG2VAL), offset 0x164

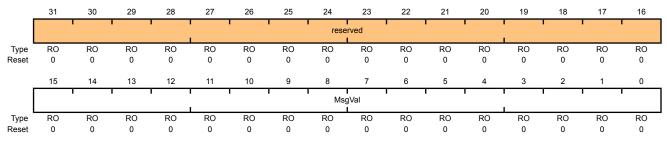
The CANMSG1VAL and CANMSG2VAL registers hold the MsqVal bits of the 32 message objects. By reading these bits, the CPU can check which message object is valid. The message value of a specific message object can be changed with the CAN IFn Message Control (CANIFnMCTL) register.

The CANMSG1VAL register contains the MsqVal bit of the first 16 message objects in the message RAM; the CANMSG2VAL register contains the MsgVal bit of the second 16 message objects in the message RAM.

CAN Message 1 Valid (CANMSG1VAL)

CAN0 base: 0x4004.0000

Offset 0x160 Type RO, 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	MsgVal	RO	0x00	Message Valid Bits

(of all message objects)

^{0:} This message object is not configured and is ignored by the message handler.

^{1:} This message object is configured and should be considered by the message handler.

17 Ethernet Controller

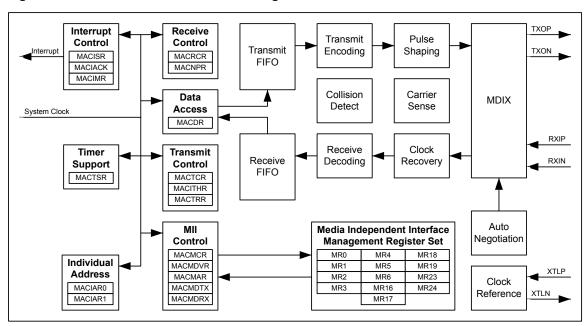
The Stellaris[®] Ethernet Controller consists of a fully integrated media access controller (MAC) and network physical (PHY) interface device. The Ethernet Controller conforms to *IEEE 802.3* specifications and fully supports 10BASE-T and 100BASE-TX standards.

The 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
- IEEE 1588 Precision Time Protocol

17.1 Block Diagram

Figure 17-1. Ethernet Controller Block Diagram

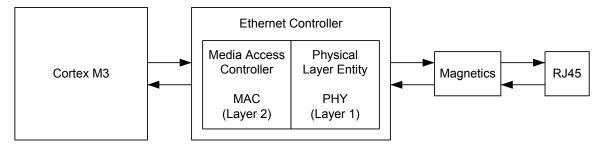


17.2 Functional Description

Note: Stellaris® Fury-class devices incorporating an Ethernet controller should have a 12.4-kΩ resistor connected between ERBIAS and ground to accommodate future device revisions. The 12.4-kΩ 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.

As shown in Figure 17-2 on page 458, the Ethernet Controller is functionally divided into two layers or modules: the Media Access Controller (MAC) layer and the Network Physical (PHY) layer. These correspond to the OSI model layers 2 and 1. The primary interface to the Ethernet Controller is a simple bus interface to the MAC layer. The MAC layer provides transmit and receive processing for Ethernet frames. The MAC layer also provides the interface to the PHY module via an internal Media Independent Interface (MII).

Figure 17-2. Ethernet Controller



17.2.1 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 **MACMDV** register contains the divider used for scaling down the system clock. See page 478 for more details about the use of this register.

17.2.2 PHY Configuration/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.

17.2.2.1 Clock Selection

The PHY 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.

17.2.2.2 Auto-Negotiation

The PHY supports the auto-negotiation functions of Clause 28 of the *IEEE 802.3* standard for 10/100 Mbps operation over copper wiring. This function can be enabled via register settings. The auto-negotiation function defaults to On and the ANEGEN bit in the **MR0** register is High after reset. Software can disable the auto-negotiation function by writing to the ANEGEN bit. The contents of the **MR4** register are sent to the PHY's link partner during auto-negotiation via fast-link pulse coding.

Once auto-negotiation is complete, the DPLX and RATE bits in the **MR18** register reflect the actual speed and duplex that was chosen. 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. Writing a 1 to the RANEG bit in the **MR0** register also causes auto-negotiation to restart.

17.2.2.3 Polarity Correction

The PHY 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 MR16 register control this feature. The default is automatic mode, where APOL is Low and RVSPOL indicates if the detection circuitry has inverted the input signal. To enter manual mode, APOL should be set High and RVSPOL then controls the signal polarity.

17.2.2.4 MDI/MDI-X Configuration

The PHY supports the automatic MDI/MDI-X configuration as defined in *IEEE 802.3-2002* specification. This eliminates the need for cross-over cables when connecting to another device, such as a hub. The algorithm is controlled via settings in the **MR24** register. Refer to page 501 for additional details about these settings.

17.2.2.5 LED Indicators

The PHY supports two LED signals that can be used to indicate various states of operation of the Ethernet Controller. 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 hardware function. See "General-Purpose Input/Outputs (GPIOs)" on page 169 for additional details. The function of these pins is programmable via the PHY layer **MR23** register. Refer to page 500 for additional details on how to program these LED functions.

17.2.3 MAC Configuration/Operation

17.2.3.1 Ethernet Frame Format

Ethernet data is carried by Ethernet frames. The basic frame format is shown in Figure 17-3 on page 460.

Figure 17-3. Ethernet Frame

Preamble	SFD	Destination Address	Source Address	Length/ Type	Data	FCS
7	1	6	6	2	46 - 1500	4
Bytes	Byte	Bytes	Bytes	Bytes	Bytes	Bytes

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 by the physical layer signaling circuitry 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 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. The first of two octets is most significant. 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 standard. The MAC module assumes type interpretation if the value of the Length/Type field is greater than 1500 decimal.

Data

The data field is a sequence of 0 to 1500 octets. Full data transparency is provided so any values can appear in this field. A minimum frame size is required to properly meet the IEEE standard. If necessary, the data field is extended by appending extra bits (a pad). The pad field can have a size of 0 to 46 octets. The sum of the data and pad lengths must be a minimum of 46 octets. The MAC module automatically inserts pads if required, though it can be disabled by a register write. For the MAC module core, data sent/received can be larger than 1500 bytes, and no Frame Too Long error is reported. Instead, a FIFO Overrun error is reported when the frame received is too large to fit into the Ethernet Controller's RAM.

Frame Check Sequence (FCS)

The frame check sequence carries the cyclic redundancy check (CRC) value. The value of this field is computed over destination address, source address, length/type, data, and pad fields using the CRC-32 algorithm. The MAC module computes the FCS value one nibble at a time. For transmitted frames, this field is automatically inserted by the MAC layer, unless disabled by 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 the BADCRC bit in the **MACRCTL** register.

17.2.3.2 MAC Layer FIFOs

For Ethernet frame transmission, a 2 KB TX 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.

For Ethernet frame reception, a 2-KB RX 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.

For details regarding the TX and RX FIFO layout, refer to Table 17-1 on page 461. 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 FCS and Frame Length bytes. Also note that if FCS generation is disabled with the CRC bit in the **MACTCTL** register, the last word in the FIFO must be 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 overlaps words in the FIFO. However, for the RX FIFO, the beginning of the next frame is always on a word boundary.

Table 17-1. TX & RX FIFO Organization

FIFO Word Read/Write Sequence	Word Bit Fields	TX FIFO (Write)	RX FIFO (Read)
1st	7:0	Data Length LSB	Frame Length LSB
	15:8	Data Length MSB	Frame Length MSB
	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

FIFO Word Read/Write Sequence			RX FIFO (Read)				
3rd	7:0	5	SA oct 1				
	15:8		SA oct 2				
	23:16	5	SA oct 3				
	31:24	5	SA oct 4				
4th	7:0	7:0 SA oct 5					
	15:8 SA oct 6						
	23:16	ype MSB					
	31:24	31:24 Len/Type LSB					
5th to nth	7:0	d	data oct n				
	15:8	da	data oct n+1				
	23:16	da	ta oct n+2				
	31:24	da	ta oct n+3				
last	7:0	FCS 1 (if the CRC bit in MACCTL is 0)	FCS 1				
	15:8	FCS 2 (if the CRC bit in MACCTL is 0)	FCS 2				
	23:16	FCS 3 (if the CRC bit in MACCTL is 0)	FCS 3				
	31:24	FCS 4 (if the CRC bit in MACCTL is 0)	FCS 4				

17.2.3.3 Ethernet Transmission Options

The Ethernet Controller can automatically generate and insert the Frame Check Sequence (FCS) at the end of the transmit frame. This is controlled by the CRC bit in the **MACTCTL** register. For test purposes, in order to generate a frame with an invalid CRC, this feature can be disabled.

The *IEEE 802.3 specification* requires that the Ethernet frame payload section be a minimum of 46 bytes. The Ethernet Controller can be configured to automatically pad the data section if the payload data section loaded into the FIFO is less than the minimum 46 bytes. This feature is controlled by the PADEN bit in the **MACTCTL** register.

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.

17.2.3.4 Ethernet Reception Options

Using the BADCRC bit in the **MACRCTL** register, the Ethernet Controller can be configured to reject incoming Ethernet frames with an invalid FCS field.

The Ethernet receiver can also be configured for Promiscuous and Multicast modes using the PRMS and AMUL fields in the **MACRCTL** register. If these modes are not enabled, only Ethernet frames with a broadcast address, or frames matching the MAC address programmed into the **MACIA0** and **MACIA1** register is placed into the RX FIFO.

17.2.3.5 Packet Timestamps

Using the TSEN bit in the **MACTS** register, the MAC transmit and receive interrupts can be used to trigger edge capture events on General-Purpose Timer 3. The transmit interrupt is routed to the CCP (even) input of General-Purpose Timer 3, while the receive interrupt is routed to the CCP (odd) input of General-Purpose Timer 3. This timer can then be configured in 16-bit edge capture mode

and be used with a third 16-bit free-running timer to capture a more accurate timestamp for the transmit or receive packet. This feature can be used with a protocol such as IEEE-1588 to provide more accurate timestamps of the synchronization packets, improving the overall accuracy of the protocol.

17.2.4 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 no 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

17.3 Initialization and Configuration

To use the Ethernet Controller, the peripheral must be enabled by setting the EPHY0 and EMAC0 bits in the **RCGC2** register. 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 would be 4.
- 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.
- Program the MACRCTL register to reject frames with bad FCS using a value of 0x08.

- 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 **MACDATA** register. Then set the NEWTX bit in the **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 **MACNP** register to be non-zero. Then begin reading the frame from the RX FIFO by using the **MACDATA** register. When the frame (including the FCS field) has been read, the NPR field should decrement by one. When there are no more frames in the RX FIFO, the NPR field reads 0.

17.4 Ethernet Register Map

Table 17-2 on page 464 lists the Ethernet MAC registers. All addresses given are relative to the Ethernet MAC base address of 0x4004.8000.

The IEEE 802.3 standard specifies a register set for controlling and gathering status from the PHY. 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 17-2 on page 464 also lists these MII Management registers. All addresses given are absolute and are written directly to the REGADR field of the MACMCTL register. The format of registers 0 to 15 are defined by the IEEE specification and are common to all PHY implementations. The only variance allowed is for features that may or may not be supported by a specific PHY. Registers 16 to 31 are vendor-specific registers, used to support features that are specific to a vendors PHY implementation. Vendor-specific registers not listed are reserved.

Table 17-2. Ethernet Register Map

Offset	Name	Type	Reset	Description	See page
Ethernet	MAC				
0x000	MACRIS	RO	0x0000.0000	Ethernet MAC Raw Interrupt Status	466
0x000	MACIACK	W1C	0x0000.0000	Ethernet MAC Interrupt Acknowledge	468
0x004	MACIM	R/W	0x0000.007F	Ethernet MAC Interrupt Mask	469
800x0	MACRCTL	R/W	0x0000.0008	Ethernet MAC Receive Control	470
0x00C	MACTCTL	R/W	0x0000.0000	Ethernet MAC Transmit Control	471
0x010	MACDATA	R/W	0x0000.0000	Ethernet MAC Data	472
0x014	MACIA0	R/W	0x0000.0000	Ethernet MAC Individual Address 0	474
0x018	MACIA1	R/W	0x0000.0000	Ethernet MAC Individual Address 1	475
0x01C	MACTHR	R/W	0x0000.003F	Ethernet MAC Threshold	476
0x020	MACMCTL	R/W	0x0000.0000	Ethernet MAC Management Control	477
0x024	MACMDV	R/W	0x0000.0080	Ethernet MAC Management Divider	478
0x02C	MACMTXD	R/W	0x0000.0000	Ethernet MAC Management Transmit Data	479
0x030	MACMRXD	R/W	0x0000.0000	Ethernet MAC Management Receive Data	480
0x034	MACNP	RO	0x0000.0000	Ethernet MAC Number of Packets	481

Offset	Name	Туре	Reset	Description	See page
0x038	MACTR	R/W	0x0000.0000	Ethernet MAC Transmission Request	482
0x03C	MACTS	R/W	0x0000.0000	Ethernet MAC Timer Support	483
MII Manag	gement				
-	MR0	R/W	0x3100	Ethernet PHY Management Register 0 – Control	484
-	MR1	RO	0x7849	Ethernet PHY Management Register 1 – Status	486
-	MR2	RO	0x000E	Ethernet PHY Management Register 2 – PHY Identifier 1	488
-	MR3	RO	0x7237	Ethernet PHY Management Register 3 – PHY Identifier 2	489
-	MR4	R/W	0x01E1	Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement	490
-	MR5	RO	0x0000	Ethernet PHY Management Register 5 – Auto-Negotiation Link Partner Base Page Ability	492
-	MR6	RO	0x0000	Ethernet PHY Management Register 6 – Auto-Negotiation Expansion	493
-	MR16	R/W	0x0140	Ethernet PHY Management Register 16 – Vendor-Specific	494
-	MR17	R/W	0x0000	Ethernet PHY Management Register 17 – Interrupt Control/Status	496
-	MR18	RO	0x0000	Ethernet PHY Management Register 18 – Diagnostic	498
-	MR19	R/W	0x4000	Ethernet PHY Management Register 19 – Transceiver Control	499
-	MR23	R/W	0x0010	Ethernet PHY Management Register 23 – LED Configuration	500
-	MR24	R/W	0x00C0	Ethernet PHY Management Register 24 –MDI/MDIX Control	501

17.5 Ethernet MAC Register Descriptions

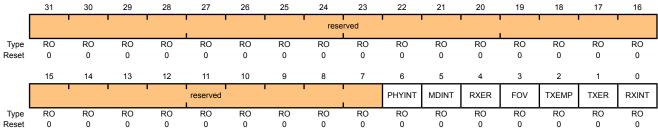
The remainder of this section lists and describes the Ethernet MAC registers, in numerical order by address offset. Also see "MII Management Register Descriptions" on page 483.

Register 1: Ethernet MAC Raw Interrupt Status (MACRIS), offset 0x000

The MACRIS register is the interrupt status register. On a read, this register gives the current status value of the corresponding interrupt prior to masking.

Ethernet MAC Raw Interrupt Status (MACRIS)

Base 0x4004.8000 Offset 0x000 Type RO, reset 0x0000.0000



Type eset	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
В	Bit/Field		Nam	ne Type		Reset	Desc	cription								
	31:7		reserv	ved	RO		0x0	Software should not rely on the value of a rescompatibility with future products, the value of preserved across a read-modify-write operation.				value of	a reserv			
	6		PHYII	NT	R	0	0x0	PHY	' Interrup	ot						
								When set, indicates that an enabled interrupt in the occured. MR17 in the PHY must be read to determine event that triggered this interrupt.								
	5		MDIN	NT	R	0	0x0	MII	Γransact	ion Com	plete					
								When set, indicates that a transacture has completed successfully.				saction	saction (read or write) on the MII interface			
	4		RXE	R	R	0	0x0	Rece	eive Erro	or						
													countere			r. The
									A receive only).	e error o	ccurs du	ring the	receptio	n of a fra	me (100) Mb/s
									The fram alignmer		an integ	er numbe	er of byte	es (dribble	e bits) d	ue to an
									The CRO	of the f	rame do	es not p	ass the F	CS che	ck.	
									The leng				nt with th	ne frame	data siz	e when
	3		FO	/	R	0	0x0	FIFC	Overrrı	un						
								Whe FIFC	-	dicates t	hat an o	verrun w	as enco	untered (on the re	eceive
	2		TXEN	ЛР	R	0	0x0	Tran	smit FIF	O Empty	/					
								Whe	n set, in	dicates t	hat the p	acket w	as transı	mitted ar	nd that th	ne TX

FIFO is empty.

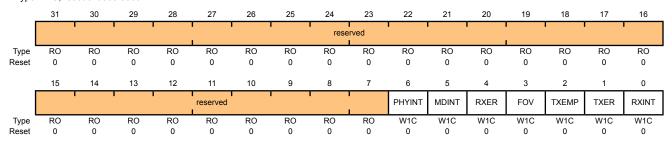
Bit/Field	Name	Туре	Reset	Description
1	TXER	RO	0x0	Transmit Error
				When set, indicates that an error was encountered on the transmitter. The possible errors that can cause this interrupt bit to be set are:
				The data length field stored in the TX FIFO exceeds 2032. The frame is not sent when this error occurs.
				The retransmission attempts during the backoff process have exceeded the maximum limit of 16.
0	RXINT	RO	0x0	Packet Received
				When set, indicates that at least one packet has been received and is stored in the receiver FIFO.

Register 2: Ethernet MAC Interrupt Acknowledge (MACIACK), offset 0x000

A write of a 1 to any bit position of this register clears the corresponding interrupt bit in the Ethernet MAC Raw Interrupt Status (MACRIS) register.

Ethernet MAC Interrupt Acknowledge (MACIACK)

Base 0x4004.8000 Offset 0x000 Type W1C, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:7	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	PHYINT	W1C	0x0	Clear PHY Interrupt
				A write of a 1 clears the PHYINT interrupt read from the MACRIS register.
5	MDINT	W1C	0x0	Clear MII Transaction Complete
				A write of a 1 clears the ${\tt MDINT}$ interrupt read from the \textbf{MACRIS} register.
4	RXER	W1C	0x0	Clear Receive Error
				A write of a 1 clears the RXER interrupt read from the MACRIS register.
3	FOV	W1C	0x0	Clear FIFO Overrun
				A write of a 1 clears the FOV interrupt read from the MACRIS register.
2	TXEMP	W1C	0x0	Clear Transmit FIFO Empty
				A write of a 1 clears the ${\tt TXEMP}$ interrupt read from the \textbf{MACRIS} register.
1	TXER	W1C	0x0	Clear Transmit Error
				A write of a 1 clears the ${\tt TXER}$ interrupt read from the \textbf{MACRIS} register and resets the TX FIFO write pointer.
0	RXINT	W1C	0x0	Clear Packet Received
				A write of a 1 clears the ${\tt RXINT}$ interrupt read from the \textbf{MACRIS} register.

Register 3: Ethernet MAC Interrupt Mask (MACIM), offset 0x004

This register allows software to enable/disable Ethernet MAC interrupts. Writing a 0 disables the interrupt, while writing a 1 enables it.

Ethernet MAC Interrupt Mask (MACIM)

Base 0x4004.8000 Offset 0x004 Type R/W, reset 0x0000.007F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			'	•			•	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
		1	'	1	reserved	1	1	'	1	PHYINTM	MDINTM	RXERM	FOVM	TXEMPM	TXERM	RXINTM
Туре	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	Ω	Λ	0	0	Ω	Λ	0	0	1	1	1	1	1	1	1

Bit/Field	Name	Туре	Reset	Description
31:7	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	PHYINTM	R/W	1	Mask PHY Interrupt
				This bit masks the ${\tt PHYINT}$ bit in the \textbf{MACRIS} register from being asserted.
5	MDINTM	R/W	1	Mask MII Transaction Complete
				This bit masks the ${\tt MDINT}$ bit in the \textbf{MACRIS} register from being asserted.
4	RXERM	R/W	1	Mask Receive Error
				This bit masks the ${\tt RXER}$ bit in the \textbf{MACRIS} register from being asserted.
3	FOVM	R/W	1	Mask FIFO Overrrun
				This bit masks the ${\tt FOV}$ bit in the \textbf{MACRIS} register from being asserted.
2	TXEMPM	R/W	1	Mask Transmit FIFO Empty
				This bit masks the ${\tt TXEMP}$ bit in the \textbf{MACRIS} register from being asserted.
1	TXERM	R/W	1	Mask Transmit Error
				This bit masks the ${\tt TXER}$ bit in the \textbf{MACRIS} register from being asserted.
0	RXINTM	R/W	1	Mask Packet Received
				This bit masks the RXINT bit in the MACRIS register from being

asserted.

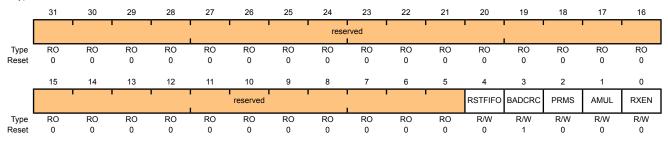
Register 4: Ethernet MAC Receive Control (MACRCTL), offset 0x008

This register enables software to configure the receive module and control the types of frames that are received from the physical medium. It is important to note that when the receive module is enabled, all valid frames with a broadcast address of FF-FF-FF-FF in the Destination Address field is 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	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.
4	RSTFIFO	R/W	0x0	Clear Receive FIFO
				When set, clears the receive FIFO. This should be done when software initialization is performed.
				It is recommended that the receiver be disabled (RXEN = 0), and then the reset initiated (RSTFIFO = 1). This sequence flushes and resets the RX FIFO.
3	BADCRC	R/W	0x1	Enable Reject Bad CRC
				The BADCRC bit enables the rejection of frames with an incorrectly calculated CRC.
2	PRMS	R/W	0x0	Enable Promiscuous Mode
				The PRMS bit enables Promiscuous mode, which accepts all valid frames, regardless of the Destination Address.
1	AMUL	R/W	0x0	Enable Multicast Frames
				The \mathtt{AMUL} bit enables the reception of multicast frames from the physical medium.
0	RXEN	R/W	0x0	Enable Receiver

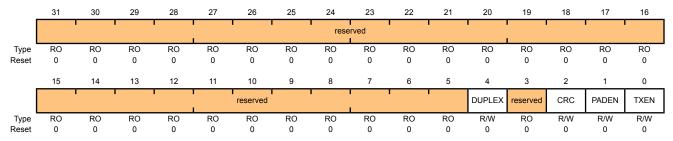
The RXEN bit enables the Ethernet receiver. When this bit is Low, the receiver is disabled and all frames on the physical medium are ignored.

Register 5: Ethernet MAC Transmit Control (MACTCTL), offset 0x00C

This register enables software to configure the transmit module, and control frames are placed onto the physical medium.

Ethernet MAC Transmit Control (MACTCTL)

Base 0x4004.8000 Offset 0x00C Type R/W, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:5	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.
4	DUPLEX	R/W	0x0	Enable Duplex Mode
				When set, enables Duplex mode, allowing simultaneous transmission and reception.
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	CRC	R/W	0x0	Enable CRC Generation
				When set, enables the automatic generation of the CRC and the placement at the end of the packet. If this bit is not set, the frames placed in the TX FIFO are sent exactly as they are written into the FIFO.
1	PADEN	R/W	0x0	Enable Packet Padding
				When set, enables the automatic padding of packets that do not meet the minimum frame size.
0	TXEN	R/W	0x0	Enable Transmitter

When set, enables the transmitter. When this bit is 0, the transmitter is disabled.

Register 6: Ethernet MAC Data (MACDATA), offset 0x010

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.

Writes to this register store the data in the TX FIFO at the location indicated by the write pointer. The write pointer is then auto-incremented to the next TX FIFO location.

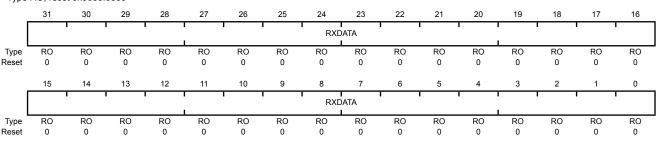
There is no mechanism for randomly accessing bytes in either the RX or TX FIFOs. Data must be read from the RX FIFO sequentially and stored in a buffer for further processing. Once a read has been performed, the data in the FIFO cannot be re-read. Data must be written to the TX FIFO sequentially. If 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.

Read-Only Register

Ethernet MAC Data (MACDATA)

Base 0x4004.8000 Offset 0x010

Type RO, reset 0x0000.0000



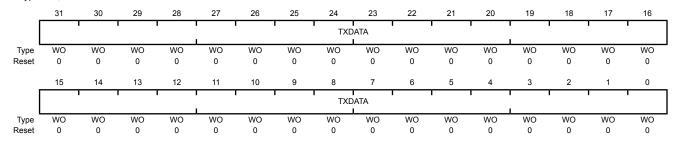
Bit/Field	Name	Туре	Reset	Description
31.0	RXDATA	RO	0x0	Receive FIFO Data

The ${\tt RXDATA}$ bits represent the next four bytes of data stored in the RX FIFO.

Write-Only Register

Ethernet MAC Data (MACDATA)

Base 0x4004.8000 Offset 0x010 Type WO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:0	TXDATA	WO	0x0	Transmit FIFO Data
				The ${\tt TXDATA}$ bits represent the next four bytes of data to place in the TX FIFO for transmission.

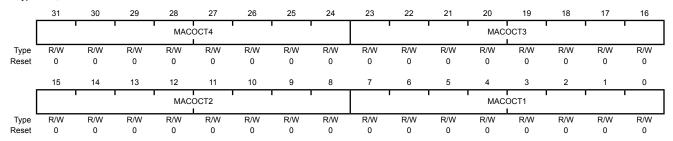
Register 7: Ethernet MAC Individual Address 0 (MACIA0), offset 0x014

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 IAR 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	Type	Reset	Description
31:24	MACOCT4	R/W	0x0	MAC Address Octet 4
				The ${\tt MACOCT4}$ bits represent the fourth octet of the MAC address used to uniquely identify each Ethernet Controller.
23:16	MACOCT3	R/W	0x0	MAC Address Octet 3
				The MACOCT3 bits represent the third octet of the MAC address used to uniquely identify each Ethernet Controller.
15:8	MACOCT2	R/W	0x0	MAC Address Octet 2
				The ${\tt MACOCT2}$ bits represent the second octet of the MAC address used to uniquely identify each Ethernet Controller.
7:0	MACOCT1	R/W	0x0	MAC Address Octet 1

The MACOCT1 bits represent the first octet of the MAC address used to uniquely identify each Ethernet Controller.

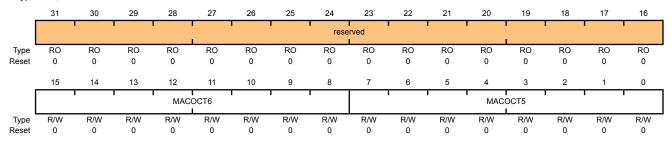
Register 8: Ethernet MAC Individual Address 1 (MACIA1), offset 0x018

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	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.
15:8	MACOCT6	R/W	0x0	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	0x0	MAC Address Octet 5

The MACOCT5 bits represent the fifth octet of the MAC address used to uniquely identify each Ethernet Controller.

Register 9: Ethernet MAC Threshold (MACTHR), offset 0x01C

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, transmission does not start until the NEWTX bit is set in the MACTR register. This effectively disables the early transmission feature.

Writing the THRESH bits to any value besides all 1s enables the early transmission feature. Once the byte count of data in the TX FIFO reaches this level, 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 would wait for 36 bytes of data to be written while a value of 0x02 would wait for 68 bytes to be written. 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 sent out on the physical medium. Because under-run checking is not performed, it is possible that the tail pointer may reach and pass the write pointer in the TX FIFO. This causes indeterminate values to be written to the physical medium rather than the end of the frame. Therefore, sufficient bus bandwidth for writing to the TX FIFO must be guaranteed by the software.

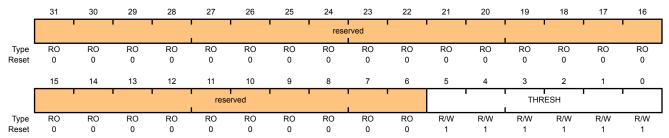
If a frame smaller than the threshold level needs to be sent, the NEWTX bit in the **MACTR** register must be set with an explicit write. This initiates the transmission of the frame even though the threshold limit has not been reached.

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.

Ethernet MAC Threshold (MACTHR)

Base 0x4004.8000 Offset 0x01C

Type R/W, reset 0x0000.003F



Bit/Field	Name	Type	Reset	Description
31:6	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	THRESH	R/W	0x3F	Threshold Value

The $\verb|Thresh|$ bits represent the early transmit threshold. Once the amount of data in the TX FIFO exceeds this value, transmission of the packet begins.

Register 10: 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. The address, name, type, reset configuration, and functional description of each of these registers can be found in Table 17-2 on page 464 and in "MII Management Register Descriptions" on page 483.

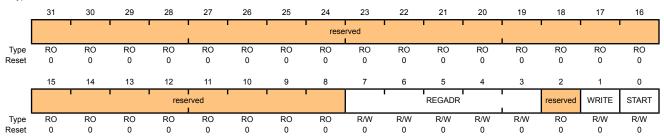
In order to initiate a *read* transaction from the MII Management registers, the WRITE bit must be written with a 0 during the same cycle that the START bit is written with a 1.

In order to initiate a *write* transaction to the MII Management registers, the WRITE bit must be written with a 1 during the same cycle that the START bit is written with a 1.

Ethernet MAC Management Control (MACMCTL)

Base 0x4004.8000 Offset 0x020

Type R/W, reset 0x0000.0000



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.
7:3	REGADR	R/W	0x0	MII Register Address
				The REGADR bit field represents the MII Management register address for the next MII management interface transaction.
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	WRITE	R/W	0x0	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; otherwise, it is a read.
0	START	R/W	0x0	MII Register Transaction Enable

The START bit represents the initiation of the next MII management interface transaction. When a 1 is written to this bit, the MII register located at <code>REGADR</code> is read (<code>WRITE=0</code>) or written (<code>WRITE=1</code>).

Register 11: 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:

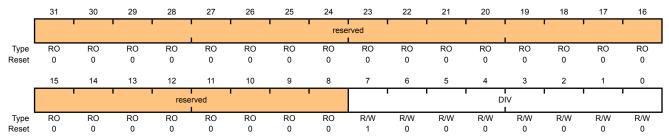
$$F_{mdc} = F_{ipclk} / (2 * (MACMDVR + 1))$$

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	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	DIV	R/W	0x80	Clock Divider

The DIV bits are used to set the clock divider for the MDC clock used to transmit data between the MAC and PHY over the serial MII interface.

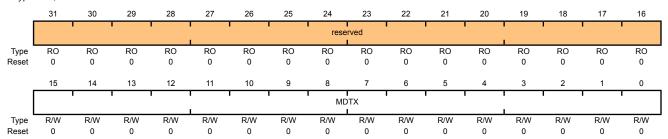
Register 12: Ethernet MAC Management Transmit Data (MACMTXD), offset 0x02C

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	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.
15:0	MDTX	R/W	0x0	MII Register Transmit Data

The \mathtt{MDTX} bits represent the data that will be written in the next MII management transaction.

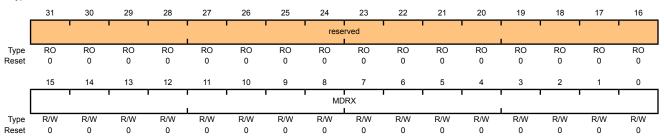
Register 13: Ethernet MAC Management Receive Data (MACMRXD), offset 0x030

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	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.
15:0	MDRX	R/W	0x0	MII Register Receive Data

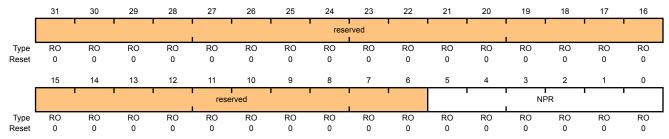
The ${\tt MDRX}$ bits represent the data that was read in the previous MII management transaction.

Register 14: 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 not set. When NPR is any other value, there is at least one frame in the RX FIFO and the RXINT bit in the MACRIS register is set.

Ethernet MAC Number of Packets (MACNP)

Base 0x4004.8000 Offset 0x034 Type RO, reset 0x0000.0000



Bit/Field	Name	Type	Reset	Description
31:6	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	NPR	RO	0x0	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 MACRISregister is asserted.

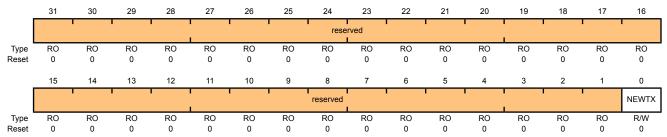
Register 15: Ethernet MAC Transmission Request (MACTR), offset 0x038

This register enables software to initiate the transmission of the frame currently located in the TX FIFO to the physical medium. Once the frame has been transmitted to the medium from the TX FIFO or a transmission error has been encountered, the NEWTX bit is auto-cleared by the hardware.

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	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.
0	NEWTX	R/W	0x0	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.

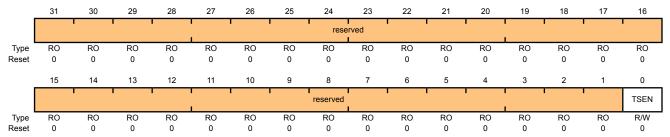
Register 16: Ethernet MAC Timer Support (MACTS), offset 0x03C

This register enables software to enable timer support on the transmission and reception of frames. This register is only applicable for devices that have 1588 hardware support; for all others, a read returns 0s.

Ethernet MAC Timer Support (MACTS)

Base 0x4004.8000 Offset 0x03C

Type R/W, reset 0x0000.0000



Bit/Field	Name	Туре	Reset	Description
31:1	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.
0	TSEN	R/W	0x0	Time Stamp Enable

When set, the ${ t TSEN}$ bit multiplexes the TX and RX interrupts to the CCP inputs of General-Purpose Timer 3.

17.6 MII Management Register Descriptions

The *IEEE 802.3 standard* specifies a register set for controlling and gathering status from the PHY. The registers are collectively known as the MII Management registers. All addresses given are absolute. Addresses not listed are reserved. Also see "Ethernet MAC Register Descriptions" on page 465.

Register 17: Ethernet PHY Management Register 0 – Control (MR0), address 0x00

This register enables software to configure the operation of the PHY. The default settings of these registers are designed to initialize the PHY to a normal operational mode without configuration.

Ethernet PHY Management Register 0 – Control (MR0)

Base 0x4004.8000 Address 0x00 Type R/W, reset 0x3100

. , pc	15	14	12	10	11	10	0	0	7	6	_	4	2	2	1	0
	15 RESET	14 LOOPBK	13 SPEEDSL	12 ANEGEN	11 PWRDN	10 ISO	9 RANEG	8 DUPLEX	7 COLT	6	5	4	3 reserved	2	1	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Type Reset	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0
E	Bit/Field		Nam	ne	Туј	ре	Reset	Des	cription							
	15		RES	ET	R/	W	0	Rese	et Regis	ters						
								inter	nal state		nes. Onc				d reinitia complet	
	14		LOOF	ВК	R/	W	0	Loop	oback M	ode						
								is is	olated fr	om the p	physical	medium		smissior	e receive ns are se	
	13		SPEE	DSL	R/	W	1	Spe	ed Seled	ct						
								Valu	ue Desc	cription						
								1	Enat	oles the	100 Mb/s	s mode o	of operati	on (100	BASE-TX	<).
								0	Enab	oles the	10 Mb/s	mode of	operatio	n (10BA	SE-T).	
	12		ANEG	SEN	R/	W	1	Auto	-Negotia	ation En	able					
								Whe	en set, e	nables tl	he Auto-	Negotiat	ion proce	ess.		
	11		PWR	DN	R/	W	0	Pow	er Dowr	า						
								Whe	en set, p	laces the	e PHY in	to a low-	power c	onsumin	g state.	
	10		ISC)	R/	W	0	Isola	ate							
										olates tr these b		and recei	ve data _l	oaths an	d ignores	s all
	9		RANI	ΞG	R/	W	0	Rest	tart Auto	-Negotia	ation					
												Negotiati hardwa		ess. Onc	e the res	tart has
	8		DUPL	.EX	R/	W	1	Set I	Duplex I	Mode						
								Valu	ue Desc	cription						
								1	set b	y softwa		nanual c			nis bit car ess or by	
								0		_			e of oper	ation.		_

Bit/Field	Name	Туре	Reset	Description
7	COLT	R/W	0	Collision Test
				When set, enables the Collision Test mode of operation. The ${\tt COLT}$ bit asserts after the initiation of a transmission and de-asserts once the transmission is halted.
6:0	reserved	R/W	0x00	Write as 0, ignore on read.

Register 18: Ethernet PHY Management Register 1 – Status (MR1), address 0x01

This register enables software to determine the capabilities of the PHY and perform its initialization and operation appropriately.

7

Ethernet PHY Management Register 1 – Status (MR1)

12

10

13

Base 0x4004.8000 Address 0x01 Type RO, reset 0x7849

15

	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 Name			ne	Ту	ре	Reset	Des	cription									
	15	5 reserved		/ed	RO		0	com	ware sho patibility served ac	with futu	ıre produ	ucts, the	value of	a reserv				
	14 100X_F		_F	R	0	1	100	BASE-TX	K Full-Dι	ıplex Mo	de							
				When set, indicates Full-Duplex mode.				nat the P	HY is ca	pable of	supporti	ng 100B	ASE-TX					
	13		100X	_H	R	С	1	100	BASE-TX	K Half-Dı	uplex Mo	ode						
								When set, indicates that the PHY is capa Half-Duplex mode.					pable of	supporti	ng 100B	ASE-TX		
	12		10T_	_F	R	Э	1		10BASE-T Full-Duplex Mode									
						Whe	en set, in le.	dicates t	hat the F	PHY is c	apable o	f 10BAS	E-T Full	-Duplex				
	11		10T_	_H	RO		1	10B	10BASE-T Half-Duplex Mode									
									en set, in -Duplex		hat the f	PHY is c	apable o	f suppor	ting 10B	ASE-T		
	10:7		reserv	/ed	R	0	0	com	ware shoupatibility served ac	with futu	ıre produ	ucts, the	value of	a reserv				
	6		MFP	'S	R	С	1	Man	nagemen	t Frames	s with Pr	eamble	Suppres	sed				
									en set, in eiving ma							of		
	5		ANE	ЭC	R	С	0	Auto	o-Negotia	ation Cor	mplete							
								com	en set, in ipleted ai o-Negotia	nd that tl	ne exten	ded regi				n		
	4		RFAL	JLT	R	С	0	Ren	note Fau	t								

When set, indicates that a remote fault condition has been detected. This bit remains set until it is read, even if the condition no longer exists.

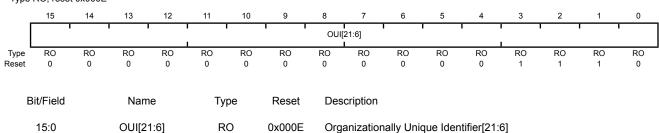
Bit/Field	Name	Туре	Reset	Description
3	ANEGA	RO	1	Auto-Negotiation When set, indicates that the PHY has the ability to perform Auto-Negotiation.
2	LINK	RO	0	Link Made When set, indicates that a valid link has been established by the PHY.
1	JAB	RC	0	Jabber Condition When set, indicates that a jabber condition has been detected by the PHY. This bit remains set until it is read, even if the jabber condition no longer exists.
0	EXTD	RO	1	Extended Capabilities When set, indicates that the PHY provides an extended set of capabilities that can be accessed through the extended register set.

Register 19: 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



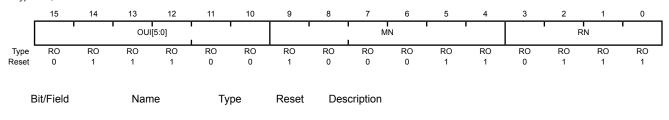
This field, along with the $\mathtt{OUI[5:0]}$ field in **MR3**, makes up the Organizationally Unique Identifier indicating the PHY manufacturer.

Register 20: 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



1	5:10	OUI[5:0]	RO	0x1C	Organizationally Unique Identifier[5:0]
					This field, along with the ${\tt OUI[21:6]}$ field in MR2 , makes up the Organizationally Unique Identifier indicating the PHY manufacturer.
	9:4	MN	RO	0x23	Model Number
					The ${\tt MN}$ 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.

Register 21: Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4), address 0x04

This register provides the advertised abilities of the PHY 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.

Ethernet PHY Management Register 4 – Auto-Negotiation Advertisement (MR4)

Base 0x4004.8000 Address 0x04 Type R/W, reset 0x01E1

ī	15	14	13	12	11	10	9	8	7	6	5	4 1	3	2	1	0	
	NP	reserved	RF		reser			A3	A2	A1	A0			S[4:0]			
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	
E	Bit/Field		Nan	ne	Тур	ре	Reset	Description									
	15		NF	•	R	0	0	Nex	Next Page								
								et, indicates the PHY is capable of Next Page exchanges to more detailed information on the PHY's capabilities.									
	14 reserved		ved	R	0	0	com	patibility	with futu	not rely on the value of a reserved bit. To provide a future products, the value of a reserved bit should be a read-modify-write operation.							
	13 RF			R/\	W	0	Ren	note Fau	It								
								en set, in been en			k partne	r that a F	Remote F	ault cor	ndition		
	12:9 reserved			ved	R	0	0	com	ware shoup apatibility served ac	with futu	ure prod	ucts, the	value of	a reserv			
	8		A3	3	R/	W	1	Tecl	hnology <i>i</i>	Ability Fi	eld[3]						
								sign this	When set, indicates that the PHY supports the 100Base-TX full-duplex signaling protocol. If software wants to ensure that this mode is not used, his bit can be written to 0 and Auto-Negotiation re-initiated with the RANEG bit in the MR0 register.								
	7		A2	2	R/	W	1	Tecl	hnology <i>i</i>	Ability Fi	eld[2]						
								sign	aling pro	on set, indicates that the PHY supports the 100Base-T half-dualing protocol. If software wants to ensure that this mode is not ubit can be written to 0 and Auto-Negotiation re-initiated.							
	6		A1		R/	W	1	Tecl	hnology <i>i</i>	Ability Fi	eld[1]						
								signaling pro		set, indicates that the PHY supports the 10Base-T full-duplex ng protocol. If software wants to ensure that this mode is not used can be written to 0 and Auto-Negotiation re-initiated.						•	
	5		AC)	R/	W	1	Tecl	hnology i	Ability Fi	eld[0]						
Ü								sign	en set, in aling pro bit can b	tocol. If s	oftware	wants to	ensure t	hat this n	node is n	•	

Bit/Field	Name	Type	Reset	Description
4:0	S[4:0]	RO	0x01	Selector Field

The ${\tt S[4:0]}$ field encodes 32 possible messages for communicating between PHYs. This field is hard-coded to 0x01, indicating that the Stellaris PHY is *IEEE 802.3* compliant.

Register 22: 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 PHY that are received and stored during Auto-Negotiation.

Ethernet PHY Management Register 5 - Auto-Negotiation Link Partner Base Page Ability (MR5)

10

Base 0x4004.8000 Address 0x05 Type RO, reset 0x0000

14

13

A[7:0]

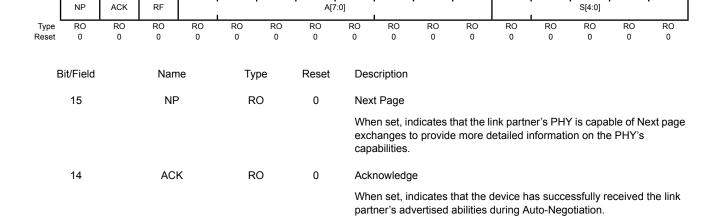
12

11

15

13

12:5



RF RO 0 Remote Fault

Used as a standard transport mechanism for transmitting simple fault information.

Technology Ability Field

The A[7:0] field encodes individual technologies that are supported by the PHY. See the **MR4** register.

4:0 S[4:0] RO 0x00 Selector Field

RO

0x00

The S[4:0] field encodes possible messages for communicating between PHYs.

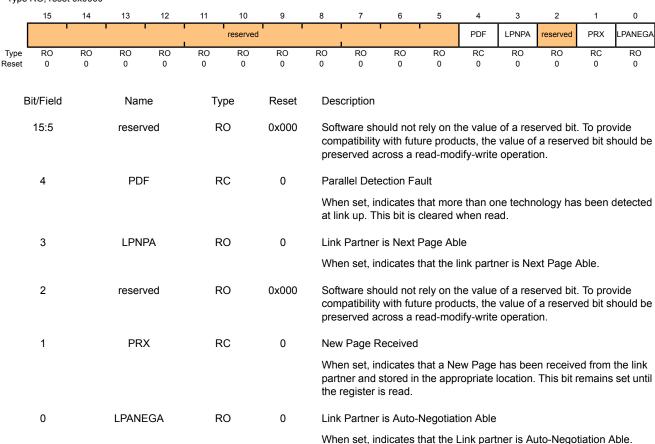
Value	Description
0x00	Reserved
0x01	IEEE Std 802.3
0x02	IEEE Std 802.9 ISLAN-16T
0x03	IEEE Std 802.5
0x04	IEEE Std 1394
0v05_0v1E	Peserved

Register 23: 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 PHY 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

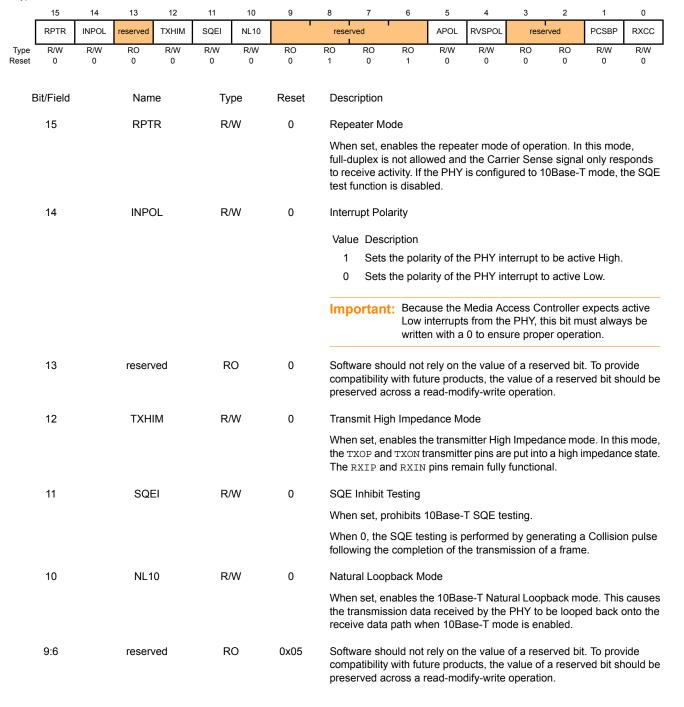


Register 24: Ethernet PHY Management Register 16 – Vendor-Specific (MR16), address 0x10

This register enables software to configure the operation of vendor-specific modes of the PHY.

Ethernet PHY Management Register 16 – Vendor-Specific (MR16)

Base 0x4004.8000 Address 0x10 Type R/W, reset 0x0140



Bit/Field	Name	Туре	Reset	Description
5	APOL	R/W	0	Auto-Polarity Disable
				When set, disables the PHY's auto-polarity function.
				If this bit is 0, the PHY automatically inverts the received signal due to a wrong polarity connection during Auto-Negotiation if the PHY is 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, a 1 in the RVSPOL bit indicates that the receive data is inverted while a 0 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 while a 0 does not invert the receive data.
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	PCSBP	R/W	0	PCS Bypass
				When set, 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-T mode is enabled.
0	RXCC	R/W	0	Receive Clock Control
				When set, enables the Receive Clock Control power saving mode if the PHY is configured in 100Base-TX mode. This mode shuts down the receive clock when no data is being received from the physical medium 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 25: 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 interrupt in the **MACRIS** register. This register can also be used in a polling mode via the MII Serial Interface as a means to observe key events within the PHY via one register address. Bits 0 through 7 are status bits, which are each set to logic 1 based on an event. These bits are cleared after the register is read. Bits 8 through 15 of this register, when set to logic 1, enable their corresponding bit in the lower byte to signal a PHY 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_E	JABBER_INT	RXER_INT	PRX_INT	PDF_INT	LPACK_INT	LSCHG_INT	RFAULT_INT	anegoomp_nt	
Type	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	Type	Reset	Description
15	JABBER_IE	R/W	0	Jabber Interrupt Enable
				When set, enables system interrupts when a Jabber condition is detected by the PHY.
14	RXER_IE	R/W	0	Receive Error Interrupt Enable
				When set, enables system interrupts when a receive error is detected by the PHY.
13	PRX_IE	R/W	0	Page Received Interrupt Enable
				When set, enables system interrupts when a new page is received by the PHY.
12	PDF_IE	R/W	0	Parallel Detection Fault Interrupt Enable
				When set, enables system interrupts when a Parallel Detection Fault is detected by the PHY.
11	LPACK_IE	R/W	0	LP Acknowledge Interrupt Enable
				When set, enables system interrupts when FLP bursts are received with the Acknowledge bit during Auto-Negotiation.
10	LSCHG_IE	R/W	0	Link Status Change Interrupt Enable
				When set, enables system interrupts when the Link Status changes from OK to FAIL.
9	RFAULT_IE	R/W	0	Remote Fault Interrupt Enable
				When set, 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, enables system interrupts when the Auto-Negotiation sequence has completed successfully.

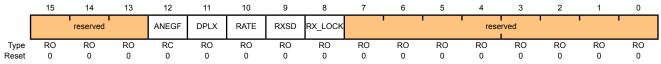
Bit/Field	Name	Туре	Reset	Description
7	JABBER_INT	RC	0	Jabber Event Interrupt
				When set, indicates that a Jabber event has been detected by the 10Base-T circuitry.
6	RXER_INT	RC	0	Receive Error Interrupt
				When set, indicates that a receive error has been detected by the PHY.
5	PRX_INT	RC	0	Page Receive Interrupt
				When set, 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, indicates that a Parallel Detection Fault has been detected by the PHY during the Auto-Negotiation process.
3	LPACK_INT	RC	0	LP Acknowledge Interrupt
				When set, indicates that an FLP burst has been received with the Acknowledge bit set during Auto-Negotiation.
2	LSCHG_INT	RC	0	Link Status Change Interrupt
				When set, indicates that the link status has changed from OK to FAIL.
1	RFAULT_INT	RC	0	Remote Fault Interrupt
				When set, indicates that a Remote Fault condition has been signaled by the link partner.
0	ANEGCOMP_INT	RC	0	Auto-Negotiation Complete Interrupt
				When set, indicates that the Auto-Negotiation sequence has completed successfully.

Register 26: 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	Type	Reset	Description
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	ANEGF	RC	0	Auto-Negotiation Failure
				When set, indicates that no common technology was found during Auto-Negotiation and has failed. This bit remains set until read.
11	DPLX	RO	0	Duplex Mode
				When set, 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, indicates that 100Base-TX was the highest common denominator found during the Auto-Negotiation process. Otherwise, 10Base-TX was the highest common denominator found.
9	RXSD	RO	0	Receive Detection
				When set, 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, 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	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.

Register 27: 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)

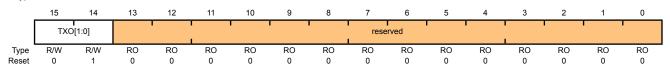
RO

reserved

0x0

Base 0x4004.8000 Address 0x13 Type R/W, reset 0x4000

13:0



Bit/Field	Name	Type	Reset	Description
15:14	TXO[1:0]	R/W	1	Transmit Amplitude Selection
				The TXO[1:0] field sets the transmit output amplitude to account for transmit transformer insertion loss.
				Value Description
				0x0 Gain set for 0.0dB of insertion loss
				0x1 Gain set for 0.4dB of insertion loss
				0x2 Gain set for 0.8dB of insertion loss
				0x3 Gain set for 1.2dB of insertion loss

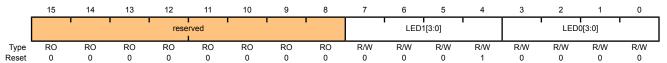
Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.

Register 28: Ethernet PHY Management Register 23 – LED Configuration (MR23), address 0x17

This register enables software to select the source that causes the LEDs 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	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.
7:4	LED1[3:0]	R/W	1	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 0 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

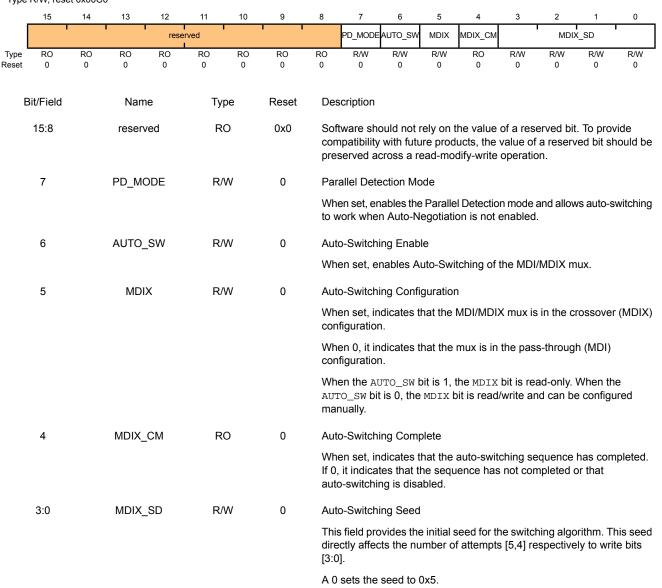
0x8 Link OK & Blink=RX or TX Activity

Register 29: 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



18 Analog Comparators

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

The LM3S8933 controller provides three independent integrated analog comparators that can be configured to drive an output or generate an interrupt or ADC event.

Note: Not all comparators have the option to drive an output pin. See the Comparator Operating Mode tables in "Functional Description" on page 503 for more information.

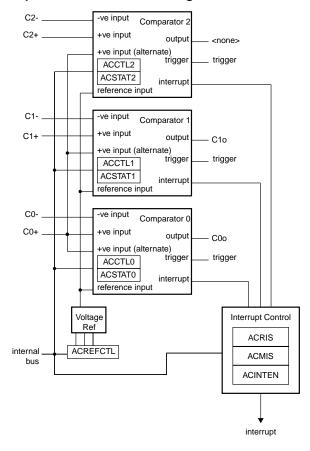
A comparator can compare a test voltage against any one of these voltages:

- 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.

18.1 Block Diagram

Figure 18-1. Analog Comparator Module Block Diagram



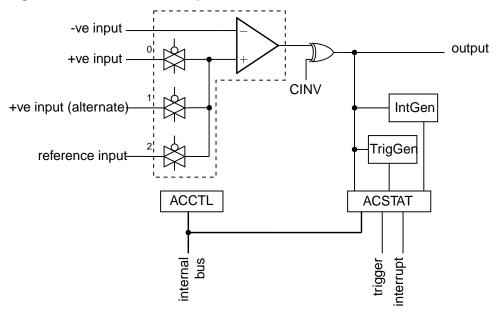
18.2 Functional Description

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.

As shown in Figure 18-2 on page 504, 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 18-2. 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). The operating modes of the comparators are shown in the Comparator Operating Mode tables.

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: Certain register bit values must be set before using the analog comparators. The proper pad configuration for the comparator input and output pins are described in the Comparator Operating Mode tables.

Table 18-1. Comparator 0 Operating Modes

ACCNTL0	Com	Comparator 0									
ASRCP	VIN-	VIN+	Output	Interrupt	ADCTrigger						
00	C0-	C0+	C0o	yes	yes						
01	C0-	C0+	C0o	yes	yes						
10	C0-	Vref	C0o	yes	yes						
11	C0-	reserved	C0o	yes	yes						

Table 18-2. Comparator 1 Operating Modes

ACCNTL1	Com	Comparator 1									
ASRCP	VIN-	VIN+	Output	Interrupt	ADCTrigger						
00	C1-	C1o/C1+	C1o/C1+	yes	yes						
01	C1-	C0+	C1o/C1+	yes	yes						
10	C1-	Vref	C1o/C1+	yes	yes						
11	C1-	reserved	C1o/C1+	yes	yes						

Table 18-3. Comparator 2 Operating Modes

ACCNTL2	Com	Comparator 2								
ASRCP	VIN-	VIN+	Output	Interrupt	ADCTrigger					
00	C2-	C2+	n/a	yes	yes					
01	C2-	C0+	n/a	yes	yes					
10	C2-	Vref	n/a	yes	yes					
11	C2-	reserved	n/a	yes	yes					

18.2.1 Internal Reference Programming

The structure of the internal reference is shown in Figure 18-3 on page 505. This is controlled by a single configuration register (**ACREFCTL**). Table 18-4 on page 505 shows the programming options to develop specific internal reference values, to compare an external voltage against a particular voltage generated internally.

Figure 18-3. Comparator Internal Reference Structure

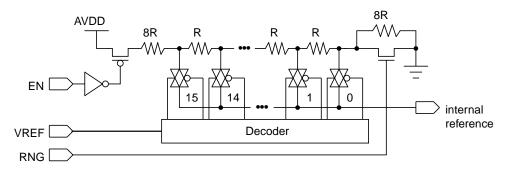


Table 18-4. Internal Reference Voltage and ACREFCTL Field Values

ACREFCTL R	Register	Output Reference Voltage Based on VREF Field Value
EN Bit Value	RNG Bit Value	
EN=0		0 V (GND) for any value of VREF; however, it is recommended that RNG=1 and VREF=0 for the least noisy ground reference.

ACREFCTL R	Register	Output Reference Voltage Based on VREF Field Value					
EN Bit Value	RNG Bit Value						
EN=1	RNG=0	Total resistance in ladder is 31 R. $V_{REF} = AV_{DD} \times \frac{R_{VREF}}{R_{T}}$					
		$V_{REF} = AV_{DD} \times \frac{(VREF + 8)}{31}$					
		$V_{RBF} = 0.85 + 0.106 \times VREF$					
		The range of internal reference in this mode is 0.85-2.448 V.					
	RNG=1	Total resistance in ladder is 23 R. $V_{REF} = AV_{DD} \times \frac{Rv_{REF}}{Rr}$					
		$V_{REF} = AV_{DD} \times \frac{VREF}{23}$ $V_{COST} = 0.143 \times VREF$					
		$V_{RBF} = 0.143 \times VREF$					
		The range of internal reference for this mode is 0-2.152 V.					

18.3 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 C0 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 on the C00 pin 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.

18.4 Register Map

Table 18-5 on page 507 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.

Table 18-5. Analog Comparators Register Map

Offset	Name	Туре	Reset	Description	See page
0x00	ACMIS	R/W1C	0x0000.0000	Analog Comparator Masked Interrupt Status	508
0x04	ACRIS	RO	0x0000.0000	Analog Comparator Raw Interrupt Status	509
0x08	ACINTEN	R/W	0x0000.0000	Analog Comparator Interrupt Enable	510
0x10	ACREFCTL	R/W	0x0000.0000	Analog Comparator Reference Voltage Control	511
0x20	ACSTAT0	RO	0x0000.0000	Analog Comparator Status 0	512
0x24	ACCTL0	R/W	0x0000.0000	Analog Comparator Control 0	513
0x40	ACSTAT1	RO	0x0000.0000	Analog Comparator Status 1	512
0x44	ACCTL1	R/W	0x0000.0000	Analog Comparator Control 1	513
0x60	ACSTAT2	RO	0x0000.0000	Analog Comparator Status 2	512
0x64	ACCTL2	R/W	0x0000.0000	Analog Comparator Control 2	513

18.5 Register Descriptions

The remainder of this section lists and describes the Analog Comparator registers, in numerical order by address offset.

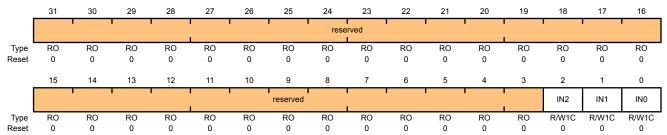
Register 1: Analog Comparator Masked Interrupt Status (ACMIS), offset 0x00

This register provides a summary of the interrupt status (masked) of the comparator.

Analog Comparator Masked Interrupt Status (ACMIS)

Base 0x4003.C000 Offset 0x00

Type R/W1C, 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	IN2	R/W1C	0	Comparator 2 Masked Interrupt Status
				Gives the masked interrupt state of this interrupt. Write 1 to this bit to clear the pending interrupt.
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.

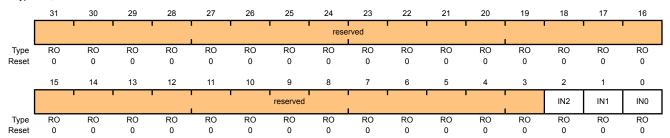
Register 2: Analog Comparator Raw Interrupt Status (ACRIS), offset 0x04

This register provides a summary of the interrupt status (raw) of the comparator.

Analog Comparator Raw Interrupt Status (ACRIS)

Base 0x4003.C000 Offset 0x04

Type RO, 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	IN2	RO	0	Comparator 2 Interrupt Status
				When set, indicates that an interrupt has been generated by comparator 2.
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 $\mathbf{0}.$

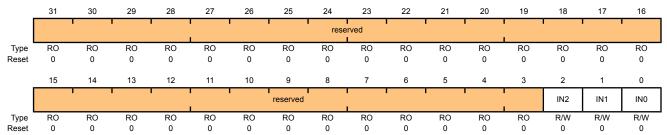
Register 3: Analog Comparator Interrupt Enable (ACINTEN), offset 0x08

This register provides the interrupt enable for the comparator.

Analog Comparator Interrupt Enable (ACINTEN)

Base 0x4003.C000

Offset 0x08
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	IN2	R/W	0	Comparator 2 Interrupt Enable When set, enables the controller interrupt from the comparator 2 output
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.

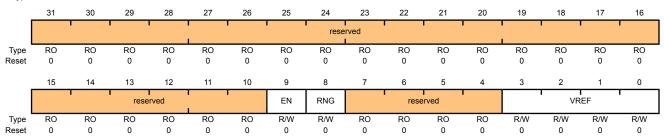
Register 4: Analog Comparator Reference Voltage Control (ACREFCTL), offset 0x10

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 0x10
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 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_{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 \mathtt{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 18-4 on page 505 for some output reference voltage examples.

Register 5: Analog Comparator Status 0 (ACSTAT0), offset 0x20

Register 6: Analog Comparator Status 1 (ACSTAT1), offset 0x40

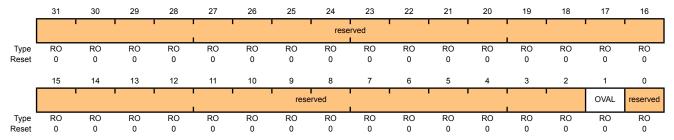
Register 7: Analog Comparator Status 2 (ACSTAT2), offset 0x60

These registers specify the current output value of the comparator.

Analog Comparator Status 0 (ACSTAT0)

Base 0x4003.C000 Offset 0x20

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 8: Analog Comparator Control 0 (ACCTL0), offset 0x24 Register 9: Analog Comparator Control 1 (ACCTL1), offset 0x44 Register 10: Analog Comparator Control 2 (ACCTL2), offset 0x64

These registers configure the comparator's input and output.

Analog Comparator Control 0 (ACCTL0)

Base 0x4003.C000

Offset 0x24
Type R/W, reset 0x0000.0000

_	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					i			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
		rese	rved		TOEN	ASF	RCP	reserved	TSLVAL	TS	EN	ISLVAL	ISI	ΞN	CINV	reserved
Туре	RO	RO	RO	RO	R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Туре	Reset	Description
31:12	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 ${ t TOEN}$ 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	Type	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 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.		

19 Pin Diagram

The LM3S8933 microcontroller pin diagrams are shown below.

Figure 19-1. 100-Pin LQFP Package Pin Diagram

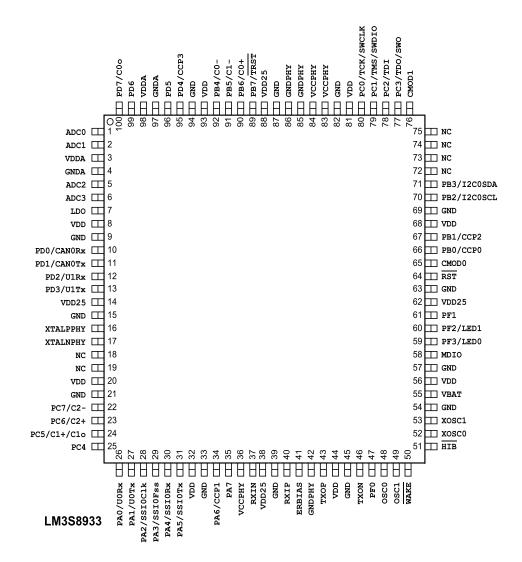


Figure 19-2. 108-Ball BGA Package Pin Diagram (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12
A	ADC1	NC	NC	NC	GNDA	PB4/ C0-	PB6/ C0+	PB7/ TRST	PCO/ TCK/ SWCLK	PC3/ TDO/ SWO	NC	NC
В	ADC0	ADC3	ADC2	NC	GNDA	GND	PB5/ C1-	PC2/ TDI	PC1/ TMS/ SWDIO	CMOD1	NC	NC
С	NC	NC	VDD25	GND	GND	VDDA	VDDA	GNDPHY	GNDPHY	VCCPHY	PB2/ I2COSCI	PB3/ I2C0SDA
D	NC	NC	VDD25							VCCPHY	VCCPHY	PB1/ CCP2
E	PD4/ CCP3	PD5	LDO							VDD33	CMOD0	PB0/ CCP0
F	PD7/ C0o	PD6	VDD25							GND	GND	GND
G	PD0/ CANORx	PD1/CAN0Tx	VDD25							VDD33	VDD33	VDD33
н	PD3/ UlTx	PD2/ U1Rx	GND							VDD33	RST	PF1
J	KTALNPHY	KTALPPHY	GND							GND	PF2/ LED1	PF3/ LED0
κ	NC	NC	ERBIAS	GNDPHY	GND	GND	VDD33	VDD33	VDD33	GND	xosc0	xosc1
L	PC4	PC7/ C2-	PAO/ UORx	PA3/ SSIOFss	PA4/ SSIORx	PA6/ CCP1	RXIN	TXON	MDIO	GND	osco	VBAT
М	PC5/ C1+/ C1o	PC6/ C2+	PA1/ UOTx	PA2/ SSIOCIL	PA5/ SSIOTx	PA7	RXIP	TXOP	PF0	WAKE	osc1	HIB

LM3S8933

20 Signal Tables

The following tables list the signals available for each pin. Functionality is enabled by software with the **GPIOAFSEL** register.

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.

Table 20-1 on page 517 shows the pin-to-signal-name mapping, including functional characteristics of the signals. Table 20-2 on page 521 lists the signals in alphabetical order by signal name.

Table 20-3 on page 525 groups the signals by functionality, except for GPIOs. Table 20-4 on page 528 lists the GPIO pins and their alternate functionality.

20.1 100-Pin LQFP Package Pin Tables

Table 20-1. Signals by Pin Number

Pin Number	Pin Name	Pin Type	Buffer Type	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 (3.3 V) 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.
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	ADC3	I	Analog	Analog-to-digital converter input 3.
7	LDO	-	Power	Low drop-out regulator output voltage. This pin requires an external capacitor between the pin and GND of 1 µF 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
	CAN0Rx	I	TTL	CAN module 0 receive
11	PD1	I/O	TTL	GPIO port D bit 1
	CAN0Tx	0	TTL	CAN module 0 transmit
12	PD2	I/O	TTL	GPIO port D bit 2
	U1Rx	I	TTL	UART module 1 receive. When in IrDA mode, this signal has IrDA modulation.
13	PD3	I/O	TTL	GPIO port D bit 3
	U1Tx	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

Pin Number	Pin Name	Pin Type	Buffer Type	Description
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	XTALP of the Ethernet PHY
17	XTALNPHY	0	TTL	XTALN of the Ethernet PHY
18	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
19	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
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
	C2-	I	Analog	Analog comparator 2 negative input
23	PC6	I/O	TTL	GPIO port C bit 6
	C2+	I	Analog	Analog comparator positive input
24	PC5	I/O	TTL	GPIO port C bit 5
	C1+	I	Analog	Analog comparator positive input
	Clo	0	TTL	Analog comparator 1 output
25	PC4	I/O	TTL	GPIO port C bit 4
26	PA0	I/O	TTL	GPIO port A bit 0
	U0Rx	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
27	PA1	I/O	TTL	GPIO port A bit 1
	UOTx	0	TTL	UART module 0 transmit. When in IrDA mode, this signal has IrDA modulation.
28	PA2	I/O	TTL	GPIO port A bit 2
	SSIOClk	I/O	TTL	SSI module 0 clock
29	PA3	I/O	TTL	GPIO port A bit 3
	SSI0Fss	I/O	TTL	SSI module 0 frame
30	PA4	I/O	TTL	GPIO port A bit 4
	SSIORx	I	TTL	SSI module 0 receive
31	PA5	I/O	TTL	GPIO port A bit 5
	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.
34	PA6	I/O	TTL	GPIO port A bit 6
	CCP1	I/O	TTL	Capture/Compare/PWM 1
35	PA7	I/O	TTL	GPIO port A bit 7
36	VCCPHY	I	TTL	VCC of the Ethernet PHY
37	RXIN	I	Analog	RXIN of the Ethernet PHY
38	VDD25	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
39	GND	-	Power	Ground reference for logic and I/O pins.

Pin Number	Pin Name	Pin Type	Buffer Type	Description
40	RXIP	I	Analog	RXIP of the Ethernet PHY
41	ERBIAS	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.
42	GNDPHY	I	TTL	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.
50	WAKE	I	-	An external input that brings the processor out of hibernate mode when asserted.
51	ĦIB	0	TTL	An output that indicates the processor is in hibernate mode.
52	xosc0	l	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
53	XOSC1	0	Analog	Hibernation Module oscillator crystal output.
54	GND	-	Power	Ground reference for logic and I/O pins.
55	VBAT	-	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
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
59	PF3	I/O	TTL	GPIO port F bit 3
	LED0	0	TTL	MII LED 0
60	PF2	I/O	TTL	GPIO port F bit 2
	LED1	0	TTL	MII 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/O	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

Pin Number	Pin Name	Pin Type	Buffer Type	Description
67	PB1	I/O	TTL	GPIO port B bit 1
	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
	I2C0SCL	I/O	OD	I2C module 0 clock
71	PB3	I/O	TTL	GPIO port B bit 3
	I2C0SDA	I/O	OD	I2C module 0 data
72	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
73	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
74	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
75	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
76	CMOD1	I/O	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
77	PC3	I/O	TTL	GPIO port C bit 3
	TDO	0	TTL	JTAG TDO and SWO
	SWO	0	TTL	JTAG TDO and SWO
78	PC2	I/O	TTL	GPIO port C bit 2
	TDI	I	TTL	JTAG TDI
79	PC1	I/O	TTL	GPIO port C bit 1
	TMS	I/O	TTL	JTAG TMS and SWDIO
	SWDIO	I/O	TTL	JTAG TMS and SWDIO
80	PC0	I/O	TTL	GPIO port C bit 0
	TCK	I	TTL	JTAG/SWD CLK
	SWCLK	1	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	I	TTL	VCC of the Ethernet PHY
84	VCCPHY	1	TTL	VCC of the Ethernet PHY
85	GNDPHY	1	TTL	GND of the Ethernet PHY
86	GNDPHY	1	TTL	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
	TRST	ı	TTL	JTAG TRSTn
90	PB6	I/O	TTL	GPIO port B bit 6
	C0+	ı	Analog	Analog comparator 0 positive input
91	PB5	I/O	TTL	GPIO port B bit 5
	C1-	I	Analog	Analog comparator 1 negative input

Pin Number	Pin Name	Pin Type	Buffer Type	Description
92	PB4	I/O	TTL	GPIO port B bit 4
	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
	CCP3	I/O	TTL	Capture/Compare/PWM 3
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.
98	VDDA	-	Power	The positive supply (3.3 V) 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.
99	PD6	I/O	TTL	GPIO port D bit 6
100	PD7	I/O	TTL	GPIO port D bit 7
	C0o	0	TTL	Analog comparator 0 output

Table 20-2. Signals by Signal Name

Pin Name	Pin Number	Pin Type	Buffer Type	Description
ADC0	1	1	Analog	Analog-to-digital converter input 0.
ADC1	2	1	Analog	Analog-to-digital converter input 1.
ADC2	5	1	Analog	Analog-to-digital converter input 2.
ADC3	6	I	Analog	Analog-to-digital converter input 3.
C0+	90	1	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	1	Analog	Analog comparator positive input
C1-	91	I	Analog	Analog comparator 1 negative input
C1o	24	0	TTL	Analog comparator 1 output
C2+	23	I	Analog	Analog comparator positive input
C2-	22	I	Analog	Analog comparator 2 negative input
CAN0Rx	10	I	TTL	CAN module 0 receive
CAN0Tx	11	0	TTL	CAN module 0 transmit
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	95	I/O	TTL	Capture/Compare/PWM 3
CMOD0	65	I/O	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	76	I/O	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
ERBIAS	41	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.
GND	9	-	Power	Ground reference for logic and I/O pins.
GND	15	-	Power	Ground reference for logic and I/O pins.
GND	21	-	Power	Ground reference for logic and I/O pins.
GND	33	-	Power	Ground reference for logic and I/O pins.
GND	39	-	Power	Ground reference for logic and I/O pins.
GND	45	-	Power	Ground reference for logic and I/O pins.
GND	54	-	Power	Ground reference for logic and I/O pins.
GND	57	-	Power	Ground reference for logic and I/O pins.
GND	63	-	Power	Ground reference for logic and I/O pins.
GND	69	-	Power	Ground reference for logic and I/O pins.
GND	82	-	Power	Ground reference for logic and I/O pins.
GND	87	-	Power	Ground reference for logic and I/O pins.
GND	94	-	Power	Ground reference for logic and I/O pins.
GNDA	4	-	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.
GNDA	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	Į	TTL	GND of the Ethernet PHY
GNDPHY	85	I	TTL	GND of the Ethernet PHY
GNDPHY	86	Į	TTL	GND of the Ethernet PHY
HIB	51	0	TTL	An output that indicates the processor is in hibernate mode.
I2C0SCL	70	I/O	OD	I2C module 0 clock
I2C0SDA	71	I/O	OD	I2C 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 µF 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	MII LED 0
LED1	60	0	TTL	MII LED 1
MDIO	58	I/O	TTL	MDIO of the Ethernet PHY
NC	18	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	19	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	72	-	-	No connect. Leave the pin electrically unconnected/isolated.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
NC	73	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	74	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	75	-	-	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.
PAO	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
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
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

Pin Name	Pin Number	Pin Type	Buffer Type	Description
PF3	59	I/O	TTL	GPIO port F bit 3
RST	64	I	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
SSIOFss	29	I/O	TTL	SSI module 0 frame
SSI0Rx	30	I	TTL	SSI module 0 receive
SSI0Tx	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 TRSTn
TXON	46	0	Analog	TXON of the Ethernet PHY
TXOP	43	0	Analog	TXOP of the Ethernet PHY
UORx	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.
VBAT	55	1	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
VCCPHY	36	I	TTL	VCC of the Ethernet PHY
VCCPHY	83	I	TTL	VCC of the Ethernet PHY
VCCPHY	84	I	TTL	VCC of the Ethernet PHY
VDD	8	-	Power	Positive supply for I/O and some logic.
VDD	20	-	Power	Positive supply for I/O and some logic.
VDD	32	-	Power	Positive supply for I/O and some logic.
VDD	44	-	Power	Positive supply for I/O and some logic.
VDD	56	-	Power	Positive supply for I/O and some logic.
VDD	68	-	Power	Positive supply for I/O and some logic.
VDD	81	-	Power	Positive supply for I/O and some logic.
VDD	93	-	Power	Positive supply for I/O and some logic.
VDD25	14	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
VDD25	38	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD25	62	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD25	88	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDDA	3	-	Power	The positive supply (3.3 V) 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	98	-	Power	The positive supply (3.3 V) 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.
WAKE	50	I	-	An external input that brings the processor out of hibernate mode when asserted.
xosc0	52	I	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
XOSC1	53	0	Analog	Hibernation Module oscillator crystal output.
XTALNPHY	17	0	TTL	XTALN of the Ethernet PHY
XTALPPHY	16	I	TTL	XTALP of the Ethernet PHY

Table 20-3. Signals by Function, Except for GPIO

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
ADC	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.
	ADC3	6	I	Analog	Analog-to-digital converter input 3.
Analog	C0+	90	Į	Analog	Analog comparator 0 positive input
Comparators	C0-	92	I	Analog	Analog comparator 0 negative input
	C0o	100	0	TTL	Analog comparator 0 output
	C1+	24	I	Analog	Analog comparator positive input
	C1-	91	Į	Analog	Analog comparator 1 negative input
	C1o	24	0	TTL	Analog comparator 1 output
	C2+	23	Į	Analog	Analog comparator positive input
	C2-	22	I	Analog	Analog comparator 2 negative input
Controller Area	CAN0Rx	10	ı	TTL	CAN module 0 receive
Network	CAN0Tx	11	0	TTL	CAN module 0 transmit

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
Ethernet PHY	ERBIAS	41	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.
	GNDPHY	42	I	TTL	GND of the Ethernet PHY
	GNDPHY	85	I	TTL	GND of the Ethernet PHY
	GNDPHY	86	I	TTL	GND of the Ethernet PHY
	LED0	59	0	TTL	MII LED 0
	LED1	60	0	TTL	MII 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
	TXON	46	0	Analog	TXON of the Ethernet PHY
	TXOP	43	0	Analog	TXOP of the Ethernet PHY
	VCCPHY	36	I	TTL	VCC of the Ethernet PHY
	VCCPHY	83	I	TTL	VCC of the Ethernet PHY
	VCCPHY	84	I	TTL	VCC of the Ethernet PHY
	XTALNPHY	17	0	TTL	XTALN of the Ethernet PHY
	XTALPPHY	16	I	TTL	XTALP of the Ethernet PHY
General-Purpose	CCP0	66	I/O	TTL	Capture/Compare/PWM 0
Timers	CCP1	34	I/O	TTL	Capture/Compare/PWM 1
	CCP2	67	I/O	TTL	Capture/Compare/PWM 2
	CCP3	95	I/O	TTL	Capture/Compare/PWM 3
I2C	I2C0SCL	70	I/O	OD	I2C module 0 clock
	I2C0SDA	71	I/O	OD	I2C module 0 data
JTAG/SWD/SWO	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
Power	GND	9	-	Power	Ground reference for logic and I/O pins.
	GND	15	-	Power	Ground reference for logic and I/O pins.
	GND	21	-	Power	Ground reference for logic and I/O pins.
	GND	33	-	Power	Ground reference for logic and I/O pins.
	GND	39	-	Power	Ground reference for logic and I/O pins.
	GND	45	-	Power	Ground reference for logic and I/O pins.
	GND	54	-	Power	Ground reference for logic and I/O pins.
	GND	57	-	Power	Ground reference for logic and I/O pins.
	GND	63	-	Power	Ground reference for logic and I/O pins.
	GND	69	-	Power	Ground reference for logic and I/O pins.
	GND	82	-	Power	Ground reference for logic and I/O pins.
	GND	87	-	Power	Ground reference for logic and I/O pins.

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
	GND	94	-	Power	Ground reference for logic and I/O pins.
	GNDA	4	-	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.
	GNDA	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.
	HIB	51	0	TTL	An output that indicates the processor is in hibernate mode.
	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).
	VBAT	55	-	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
	VDD	8	-	Power	Positive supply for I/O and some logic.
	VDD	20	-	Power	Positive supply for I/O and some logic.
	VDD	32	-	Power	Positive supply for I/O and some logic.
	VDD	44	-	Power	Positive supply for I/O and some logic.
	VDD	56	-	Power	Positive supply for I/O and some logic.
	VDD	68	-	Power	Positive supply for I/O and some logic.
	VDD	81	-	Power	Positive supply for I/O and some logic.
	VDD	93	-	Power	Positive supply for I/O and some logic.
	VDD25	14	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	38	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	62	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	88	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDDA	3	-	Power	The positive supply (3.3 V) 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	98	-	Power	The positive supply (3.3 V) 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.
	WAKE	50	I	-	An external input that brings the processor out of hibernate mode when asserted.
SSI	SSI0Clk	28	I/O	TTL	SSI module 0 clock
	SSI0Fss	29	I/O	TTL	SSI module 0 frame

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
	SSI0Rx	30	I	TTL	SSI module 0 receive
	SSIOTx	31	0	TTL	SSI module 0 transmit
System Control & Clocks	CMOD0	65	I/O	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
	CMOD1	76	I/O	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.
	RST	64	I	TTL	System reset input.
	TRST	89	I	TTL	JTAG TRSTn
	xosc0	52	I	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
	XOSC1	53	0	Analog	Hibernation Module oscillator crystal output.
UART	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.
	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.

Table 20-4. GPIO Pins and Alternate Functions

GPIO Pin	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

GPIO Pin	Pin Number	Multiplexed Function	Multiplexed Function
PC1	79	TMS	SWDIO
PC2	78	TDI	
PC3	77	TDO	SWO
PC4	25		
PC5	24	C1+	Clo
PC6	23	C2+	
PC7	22	C2-	
PD0	10	CAN0Rx	
PD1	11	CAN0Tx	
PD2	12	U1Rx	
PD3	13	U1Tx	
PD4	95	CCP3	
PD5	96		
PD6	99		
PD7	100	C0o	
PF0	47		
PF1	61		
PF2	60	LED1	
PF3	59	LED0	

20.2 108-Pin BGA Package Pin Tables

Table 20-5. Signals by Pin Number

Pin Number	Pin Name	Pin Type	Buffer Type	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
	C0-	Į	Analog	Analog comparator 0 negative input
A7	PB6	I/O	TTL	GPIO port B bit 6
	C0+	I	Analog	Analog comparator 0 positive input
A8	PB7	I/O	TTL	GPIO port B bit 7
	TRST	I	TTL	JTAG TRSTn
A9	PC0	I/O	TTL	GPIO port C bit 0
	TCK	I	TTL	JTAG/SWD CLK
	SWCLK	I	TTL	JTAG/SWD CLK

Pin Number	Pin Name	Pin Type	Buffer Type	Description
A10	PC3	I/O	TTL	GPIO port C bit 3
	TDO	0	TTL	JTAG TDO and SWO
	SWO	0	TTL	JTAG TDO and SWO
A11	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
A12	NC	ı	-	No connect. Leave the pin electrically unconnected/isolated.
B1	ADC0	I	Analog	Analog-to-digital converter input 0.
B2	ADC3	I	Analog	Analog-to-digital converter input 3.
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.
В7	PB5	I/O	TTL	GPIO port B bit 5
	C1-	I	Analog	Analog comparator 1 negative input
B8	PC2	I/O	TTL	GPIO port C bit 2
	TDI	I	TTL	JTAG TDI
В9	PC1	I/O	TTL	GPIO port C bit 1
	TMS	I/O	TTL	JTAG TMS and SWDIO
	SWDIO	I/O	TTL	JTAG TMS and SWDIO
B10	CMOD1	I/O	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
B11	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
B12	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
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 (3.3 V) 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.

Pin Number	Pin Name	Pin Type	Buffer Type	Description
C7	VDDA	-	Power	The positive supply (3.3 V) 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.
C8	GNDPHY	1	TTL	GND of the Ethernet PHY
C9	GNDPHY	I	TTL	GND of the Ethernet PHY
C10	VCCPHY	I	TTL	VCC of the Ethernet PHY
C11	PB2	I/O	TTL	GPIO port B bit 2
	I2C0SCL	I/O	OD	I2C module 0 clock
C12	PB3	I/O	TTL	GPIO port B bit 3
	I2C0SDA	I/O	OD	I2C module 0 data
D1	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
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	I	TTL	VCC of the Ethernet PHY
D11	VCCPHY	I	TTL	VCC of the Ethernet PHY
D12	PB1	I/O	TTL	GPIO port B bit 1
	CCP2	I/O	TTL	Capture/Compare/PWM 2
E1	PD4	I/O	TTL	GPIO port D bit 4
	CCP3	I/O	TTL	Capture/Compare/PWM 3
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/O	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
	CCPO	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
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 G2	PD0 CAN0Rx	I/O	TTL	0010 101110
G2	CAN0Rx		IIL	GPIO port D bit 0
G2		I	TTL	CAN module 0 receive
	PD1	I/O	TTL	GPIO port D bit 1
	CANOTx	0	TTL	CAN module 0 transmit
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.
G12	VDD33	-	Power	Positive supply for I/O and some logic.
H1	PD3	I/O	TTL	GPIO port D bit 3
	UlTx	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.
H2	PD2	I/O	TTL	GPIO port D bit 2
	UlRx	I	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	XTALN of the Ethernet PHY
J2	XTALPPHY	I	TTL	XTALP of the Ethernet PHY
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
	LED1	0	TTL	MII LED 1
J12	PF3	I/O	TTL	GPIO port F bit 3
	LED0	0	TTL	MII LED 0
K1	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
K2	NC	-	-	No connect. Leave the pin electrically unconnected/isolated.
K3	ERBIAS	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.
K4	GNDPHY	I	TTL	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.

Pin Number	Pin Name	Pin Type	Buffer Type	Description
K11	xosc0	I	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
K12	XOSC1	0	Analog	Hibernation Module oscillator crystal output.
L1	PC4	I/O	TTL	GPIO port C bit 4
L2	PC7	I/O	TTL	GPIO port C bit 7
	C2-	I	Analog	Analog comparator 2 negative input
L3	PA0	I/O	TTL	GPIO port A bit 0
	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
L5	PA4	I/O	TTL	GPIO port A bit 4
	SSI0Rx	I	TTL	SSI module 0 receive
L6	PA6	I/O	TTL	GPIO port A bit 6
	CCP1	I/O	TTL	Capture/Compare/PWM 1
L7	RXIN	1	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	VBAT	-	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
M1	PC5	I/O	TTL	GPIO port C bit 5
	C1+	I	Analog	Analog comparator positive input
	Clo	0	TTL	Analog comparator 1 output
M2	PC6	I/O	TTL	GPIO port C bit 6
	C2+	1	Analog	Analog comparator positive input
M3	PA1	I/O	TTL	GPIO port A bit 1
	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
	SSI0Clk	I/O	TTL	SSI module 0 clock
M5	PA5	I/O	TTL	GPIO port A bit 5
	SSIOTx	0	TTL	SSI module 0 transmit
M6	PA7	I/O	TTL	GPIO port A bit 7
M7	RXIP	I	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

Pin Number	Pin Name	Pin Type	Buffer Type	Description
M10	WAKE	I	-	An external input that brings the processor out of hibernate mode when asserted.
M11	OSC1	0	Analog	Main oscillator crystal output.
M12	HIB	0	TTL	An output that indicates the processor is in hibernate mode.

Table 20-6. Signals by Signal Name

Pin Name	Pin Number	Pin Type	Buffer Type	Description
ADC0	B1	1	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.
ADC3	B2	I	Analog	Analog-to-digital converter input 3.
C0+	A7	I	Analog	Analog comparator 0 positive input
C0-	A6	I	Analog	Analog comparator 0 negative input
C0o	F1	0	TTL	Analog comparator 0 output
C1+	M1	I	Analog	Analog comparator positive input
C1-	B7	I	Analog	Analog comparator 1 negative input
Clo	M1	0	TTL	Analog comparator 1 output
C2+	M2	I	Analog	Analog comparator positive input
C2-	L2	I	Analog	Analog comparator 2 negative input
CAN0Rx	G1	I	TTL	CAN module 0 receive
CANOTX	G2	0	TTL	CAN module 0 transmit
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	E1	I/O	TTL	Capture/Compare/PWM 3
CMOD0	E11	I/O	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
CMOD1	B10	I/O	TTL	CPU Mode bit 1. Input must be set to logic 0 (grounded); other encodings reserved.
ERBIAS	К3	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.
GND	C4	-	Power	Ground reference for logic and I/O pins.
GND	C5	-	Power	Ground reference for logic and I/O pins.
GND	H3	-	Power	Ground reference for logic and I/O pins.
GND	J3	-	Power	Ground reference for logic and I/O pins.
GND	K5	-	Power	Ground reference for logic and I/O pins.
GND	K6	-	Power	Ground reference for logic and I/O pins.
GND	L10	-	Power	Ground reference for logic and I/O pins.
GND	K10	-	Power	Ground reference for logic and I/O pins.
GND	J10	-	Power	Ground reference for logic and I/O pins.
GND	F10	-	Power	Ground reference for logic and I/O pins.
GND	F11	-	Power	Ground reference for logic and I/O pins.
GND	В6	-	Power	Ground reference for logic and I/O pins.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
GND	F12	-	Power	Ground reference for logic and I/O pins.
GNDA	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.
GNDA	A5	-	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	K4	I	TTL	GND of the Ethernet PHY
GNDPHY	C8	I	TTL	GND of the Ethernet PHY
GNDPHY	C9	ļ	TTL	GND of the Ethernet PHY
HIB	M12	0	TTL	An output that indicates the processor is in hibernate mode.
I2C0SCL	C11	I/O	OD	I2C module 0 clock
I2C0SDA	C12	I/O	OD	I2C 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 µF 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	MII LED 0
LED1	J11	0	TTL	MII LED 1
MDIO	L9	I/O	TTL	MDIO of the Ethernet PHY
NC	A2	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	A3	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	B4	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	A4	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	A11	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	B12	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	B11	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	A12	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	D1	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	D2	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	C2	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	C1	-	-	No connect. Leave the pin electrically unconnected/isolated.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
NC	K1	-	-	No connect. Leave the pin electrically unconnected/isolated.
NC	K2	-	-	No connect. Leave the pin electrically unconnected/isolated.
osc0	L11	I	Analog	Main oscillator crystal input or an external clock reference input.
OSC1	M11	0	Analog	Main oscillator crystal output.
PA0	L3	I/O	TTL	GPIO port A bit 0
PA1	М3	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	В7	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	В9	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
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

Pin Name	Pin Number	Pin Type	Buffer Type	Description
RST	H11	I	TTL	System reset input.
RXIN	L7	I	Analog	RXIN of the Ethernet PHY
RXIP	M7	Į	Analog	RXIP of the Ethernet PHY
SSIOClk	M4	I/O	TTL	SSI module 0 clock
SSI0Fss	L4	I/O	TTL	SSI module 0 frame
SSI0Rx	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 TRSTn
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.
VBAT	L12	-	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
VCCPHY	C10	I	TTL	VCC of the Ethernet PHY
VCCPHY	D10	I	TTL	VCC of the Ethernet PHY
VCCPHY	D11	I	TTL	VCC of the Ethernet PHY
VDD25	C3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD25	D3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD25	F3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD25	G3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
VDD33	K7	-	Power	Positive supply for I/O and some logic.
VDD33	G12	-	Power	Positive supply for I/O and some logic.

Pin Name	Pin Number	Pin Type	Buffer Type	Description
VDD33	K8	-	Power	Positive supply for I/O and some logic.
VDD33	K9	-	Power	Positive supply for I/O and some logic.
VDD33	H10	-	Power	Positive supply for I/O and some logic.
VDD33	G10	-	Power	Positive supply for I/O and some logic.
VDD33	E10	-	Power	Positive supply for I/O and some logic.
VDD33	G11	-	Power	Positive supply for I/O and some logic.
VDDA	C6	-	Power	The positive supply (3.3 V) 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	C7	-	Power	The positive supply (3.3 V) 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.
WAKE	M10	I	-	An external input that brings the processor out of hibernate mode when asserted.
XOSC0	K11	I	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
XOSC1	K12	0	Analog	Hibernation Module oscillator crystal output.
XTALNPHY	J1	0	TTL	XTALN of the Ethernet PHY
XTALPPHY	J2	ı	TTL	XTALP of the Ethernet PHY

Table 20-7. Signals by Function, Except for GPIO

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
ADC	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.
	ADC3	B2	I	Analog	Analog-to-digital converter input 3.
Analog	C0+	A7	I	Analog	Analog comparator 0 positive input
Comparators	C0-	A6	I	Analog	Analog comparator 0 negative input
	C0o	F1	0	TTL	Analog comparator 0 output
	C1+	M1	I	Analog	Analog comparator positive input
	C1-	В7	I	Analog	Analog comparator 1 negative input
	Clo	M1	0	TTL	Analog comparator 1 output
	C2+	M2	I	Analog	Analog comparator positive input
	C2-	L2	I	Analog	Analog comparator 2 negative input
Controller Area	CAN0Rx	G1	I	TTL	CAN module 0 receive
Network	CANOTX	G2	0	TTL	CAN module 0 transmit
Ethernet PHY	ERBIAS	K3	I	Analog	12.4 KOhm resistor (1% precision) used internally for Ethernet PHY.

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
	GNDPHY	K4	I	TTL	GND of the Ethernet PHY
	GNDPHY	C8	I	TTL	GND of the Ethernet PHY
	GNDPHY	C9	I	TTL	GND of the Ethernet PHY
	LED0	J12	0	TTL	MII LED 0
	LED1	J11	0	TTL	MII LED 1
	MDIO	L9	I/O	TTL	MDIO of the Ethernet PHY
	RXIN	L7	I	Analog	RXIN of the Ethernet PHY
	RXIP	M7	ļ	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	I	TTL	VCC of the Ethernet PHY
	VCCPHY	D10	I	TTL	VCC of the Ethernet PHY
	VCCPHY	D11	I	TTL	VCC of the Ethernet PHY
	XTALNPHY	J1	0	TTL	XTALN of the Ethernet PHY
	XTALPPHY	J2	I	TTL	XTALP of the Ethernet PHY
General-Purpose	CCP0	E12	I/O	TTL	Capture/Compare/PWM 0
Timers	CCP1	L6	I/O	TTL	Capture/Compare/PWM 1
	CCP2	D12	I/O	TTL	Capture/Compare/PWM 2
	CCP3	E1	I/O	TTL	Capture/Compare/PWM 3
I2C	I2C0SCL	C11	I/O	OD	I2C module 0 clock
	I2C0SDA	C12	I/O	OD	I2C module 0 data
JTAG/SWD/SWO	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
Power	GND	C4	-	Power	Ground reference for logic and I/O pins.
	GND	C5	-	Power	Ground reference for logic and I/O pins.
	GND	НЗ	-	Power	Ground reference for logic and I/O pins.
	GND	J3	-	Power	Ground reference for logic and I/O pins.
	GND	K5	-	Power	Ground reference for logic and I/O pins.
	GND	K6	-	Power	Ground reference for logic and I/O pins.
	GND	L10	-	Power	Ground reference for logic and I/O pins.
	GND	K10	-	Power	Ground reference for logic and I/O pins.
	GND	J10	-	Power	Ground reference for logic and I/O pins.
	GND	F10	-	Power	Ground reference for logic and I/O pins.
	GND	F11	-	Power	Ground reference for logic and I/O pins.
	GND	В6	-	Power	Ground reference for logic and I/O pins.
	GND	F12	-	Power	Ground reference for logic and I/O pins.

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
	GNDA	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.
	GNDA	A5	-	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.
	HIB	M12	0	TTL	An output that indicates the processor is in hibernate mode.
	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).
	VBAT	L12	-	Power	Power source for the Hibernation Module. It is normally connected to the positive terminal of a battery and serves as the battery backup/Hibernation Module power-source supply.
	VDD25	C3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	D3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	F3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD25	G3	-	Power	Positive supply for most of the logic function, including the processor core and most peripherals.
	VDD33	K7	-	Power	Positive supply for I/O and some logic.
	VDD33	G12	-	Power	Positive supply for I/O and some logic.
	VDD33	K8	-	Power	Positive supply for I/O and some logic.
	VDD33	K9	-	Power	Positive supply for I/O and some logic.
	VDD33	H10	-	Power	Positive supply for I/O and some logic.
	VDD33	G10	-	Power	Positive supply for I/O and some logic.
	VDD33	E10	-	Power	Positive supply for I/O and some logic.
	VDD33	G11	-	Power	Positive supply for I/O and some logic.
	VDDA	C6	-	Power	The positive supply (3.3 V) 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	C7	-	Power	The positive supply (3.3 V) 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.
	WAKE	M10	I	-	An external input that brings the processor out of hibernate mode when asserted.
SSI	SSI0Clk	M4	I/O	TTL	SSI module 0 clock
	SSI0Fss	L4	I/O	TTL	SSI module 0 frame
	SSI0Rx	L5	I	TTL	SSI module 0 receive

Function	Pin Name	Pin Number	Pin Type	Buffer Type	Description
	SSI0Tx	M5	0	TTL	SSI module 0 transmit
System Control & Clocks	CMOD0	E11	I/O	TTL	CPU Mode bit 0. Input must be set to logic 0 (grounded); other encodings reserved.
	CMOD1	B10	I/O	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.
	RST	H11	I	TTL	System reset input.
	TRST	A8	I	TTL	JTAG TRSTn
	xosc0	K11	I	Analog	Hibernation Module oscillator crystal input or an external clock reference input. Note that this is either a 4.19-MHz crystal or a 32.768-kHz oscillator for the Hibernation Module RTC. See the CLKSEL bit in the HIBCTL register.
	XOSC1	K12	0	Analog	Hibernation Module oscillator crystal output.
UART	U0Rx	L3	I	TTL	UART module 0 receive. When in IrDA mode, this signal has IrDA modulation.
	U0Tx	М3	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.
	U1Tx	H1	0	TTL	UART module 1 transmit. When in IrDA mode, this signal has IrDA modulation.

Table 20-8. GPIO Pins and Alternate Functions

GPIO Pin	Pin Number	Multiplexed Function	Multiplexed Function
PA0	L3	UORx	
PA1	M3	UOTx	
PA2	M4	SSI0Clk	
PA3	L4	SSI0Fss	
PA4	L5	SSIORx	
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

GPIO Pin	Pin Number	Multiplexed Function	Multiplexed Function
PC2	B8	TDI	
PC3	A10	TDO	SWO
PC4	L1		
PC5	M1	C1+	Clo
PC6	M2	C2+	
PC7	L2	C2-	
PD0	G1	CAN0Rx	
PD1	G2	CAN0Tx	
PD2	H2	U1Rx	
PD3	H1	UlTx	
PD4	E1	CCP3	
PD5	E2		
PD6	F2		
PD7	F1	C0o	
PF0	M9		
PF1	H12		
PF2	J11	LED1	
PF3	J12	LED0	

21 Operating Characteristics

Table 21-1. Temperature Characteristics

Characteristic ^a	Symbol	Value	Unit
Industrial operating temperature range	T _A	-40 to +85	°C
Extended operating temperature range	T _A	-40 to +105	°C

a. Maximum storage temperature is 150°C.

Table 21-2. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance (junction to ambient) ^a	Θ_{JA}	34	°C/W
Average junction temperature ^b	T _J	$T_A + (P_{AVG} \cdot \Theta_{JA})$	°C

- a. Junction to ambient thermal resistance θ_{JA} numbers are determined by a package simulator.
- b. Power dissipation is a function of temperature.

22 Electrical Characteristics

22.1 DC Characteristics

22.1.1 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 22-1. Maximum Ratings

Characteristic	Symbol	Va	lue	Unit
		Min	Max	
I/O supply voltage (V _{DD})	V _{DD}	0	4	٧
Core supply voltage (V _{DD25})	V _{DD25}	0	3	٧
Analog supply voltage (V _{DDA})	V_{DDA}	0	4	٧
Battery supply voltage (V _{BAT})	V _{BAT}	0	4	٧
Ethernet PHY supply voltage (V _{CCPHY})	V _{CCPHY}	0	4	٧
Input voltage	V _{IN}	-0.3	5.5	٧
Maximum current per output pins	I	-	25	mA

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).

22.1.2 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 22-2. 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 _{BAT}	Battery supply voltage	2.3	3.0	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

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{SIH}	High-level input voltage for Schmitt trigger inputs	0.8 * V _{DD}	-	V_{DD}	V
V _{SIL}	Low-level input voltage for Schmitt trigger inputs	0	-	0.2 * V _{DD}	V
V _{OH} ^a	High-level output voltage	2.4	-	-	V
V _{OL} ^a	Low-level output voltage	-	-	0.4	V
I _{OH}	High-level source current, V _{OH} =2.4 V				
	2-mA Drive	2.0	-	-	mA
	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	ı	-	mA
I _{OL}	Low-level sink current, V_{OL} =0.4 V				
	2-mA Drive	2.0	-	-	mA
	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.

22.1.3 On-Chip Low Drop-Out (LDO) Regulator Characteristics

Table 22-3. 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

22.1.4 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_{DD} = 3.3 V
- $V_{DD25} = 2.50 \text{ V}$
- V_{BAT} = 3.0 V
- 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 22-4. Detailed Power Specifications

Parameter Parameter Name		Conditions		V _{DD} , V _{DDA} ,	2.5 V V _{DD25}		3.0 V V _{BAT}		Unit
			Nom	Max	Nom	Max	Nom	Max	
I _{DD_RUN}	Run mode 1	V _{DD25} = 2.50 V	48	pending ^a	108	pending ^a	0	pending ^a	mA
	(Flash loop)	Code= while(1){} executed in Flash							
		Peripherals = All ON							
		System Clock = 50 MHz (with PLL)							
	Run mode 2	V _{DD25} = 2.50 V	5	pendinga	52	pendinga	0	pendinga	mA
	(Flash loop)	Code= while(1){} executed in Flash							
		Peripherals = All OFF							
	Run mode 1 (SRAM loop)	System Clock = 50 MHz (with PLL)							
		V _{DD25} = 2.50 V	48	pendinga	100	pendinga	0	pendinga	mA
		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	0	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	pendinga	0	pending ^a	mA
		Peripherals = All OFF							
		System Clock = 50 MHz (with PLL)							
I _{DD_DEEPSLEEP}	Deep-Sleep mode	LDO = 2.25 V	4.6	pendinga	0.21	pendinga	0	pending ^a	mA
	mode	Peripherals = All OFF							
		System Clock = IOSC30KHZ/64							
I _{DD_HIBERNATE}	Hibernate mode	V _{BAT} = 3.0 V	0	0	0	0	16	pending ^a	μA
		$V_{DD} = 0 V$							
		V _{DD25} = 0 V							
		$V_{DDA} = 0 V$							
		V _{DDPHY} = 0 V							
		Peripherals = All OFF							
		System Clock = OFF							
		Hibernate Module = 32 kHz							

a. Pending characterization completion.

22.1.5 Flash Memory Characteristics

Table 22-5. 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
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	200	-	-	ms

a. A program/erase cycle is defined as switching the bits from 1 -> 0 -> 1.

22.1.6 Hibernation

Table 22-6. Hibernation Module DC Characteristics

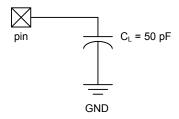
Parameter	Parameter Name	Value	Unit	
V_{LOWBAT}	Low battery detect voltage	2.35	V	

22.2 AC Characteristics

22.2.1 Load Conditions

Unless otherwise specified, the following conditions are true for all timing measurements. Timing measurements are for 4-mA drive strength.

Figure 22-1. Load Conditions



22.2.2 Clocks

Table 22-7. 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.

b. PLL frequency is automatically calculated by the hardware based on the XTAL field of the RCC register.

Table 22-8. 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	21	30	39	KHz
f _{XOSC}	Hibernation module oscillator frequency	-	4.194304	-	MHz
f _{XOSC_XTAL}	Crystal reference for hibernation oscillator	-	4.194304	-	MHz
f _{XOSC_EXT}	External clock reference for hibernation module	-	32.768	-	KHz
f _{MOSC}	Main oscillator frequency	1	-	8	MHz
t _{MOSC_per}	Main oscillator period	125	-	1000	ns
f _{ref_crystal_bypass}	Crystal reference using the main oscillator (PLL in BYPASS mode)	1	-	8	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 14-MHz to 18-MHz clock source to operate properly.

Table 22-9. Crystal Characteristics

Parameter Name		Units			
Frequency	8	6	4	3.5	MHz
Frequency tolerance	±50	±50	±50	±50	ppm
Aging	±5	±5	±5	±5	ppm/yr
Oscillation mode	Parallel	Parallel	Parallel	Parallel	-
Temperature stability (-40°C to 85°C)	±25	±25	±25	±25	ppm
Temperature stability (-40°C to 105°C)	±25	±25	±25	±25	ppm
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

22.2.3 Analog-to-Digital Converter

Table 22-10. ADC Characteristics^a

Parameter	Parameter Name	Min	Nom	Max	Unit
V _{ADCIN}	Maximum single-ended, full-scale analog input voltage	-	-	3.0	V
	Minimum single-ended, full-scale analog input voltage	-	-	0	V
	Maximum differential, full-scale analog input voltage	-	-	1.5	V
	Minimum differential, full-scale analog input voltage	-	-	-1.5	V
C _{ADCIN}	Equivalent input capacitance	-	1	-	pF
N	Resolution	-	10	-	bits
f _{ADC}	ADC internal clock frequency	14	16	18	MHz
t _{ADCCONV}	Conversion time	-	-	16	t _{ADC} cycles ^b
f _{ADCCONV}	Conversion rate	875	1000	1125	k samples/s
INL	Integral nonlinearity	-	-	±1	LSB

Parameter	Parameter Name	Min	Nom	Max	Unit
DNL	Differential nonlinearity	-	-	±1	LSB
OFF	Offset	-	-	±1	LSB
GAIN	Gain	-	-	±1	LSB

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.

22.2.4 Analog Comparator

Table 22-11. 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 22-12. Analog Comparator Voltage Reference Characteristics

Parameter	Parameter Name	Min	Nom	Max	Unit
R _{HR}	Resolution high range	-	V _{DD} /32	-	LSB
R _{LR}	Resolution low range	-	V _{DD} /24	-	LSB
A _{HR}	Absolute accuracy high range	-	-	±1/2	LSB
A _{LR}	Absolute accuracy low range	-	-	±1/4	LSB

22.2.5 I²C

Table 22-13, I²C Characteristics

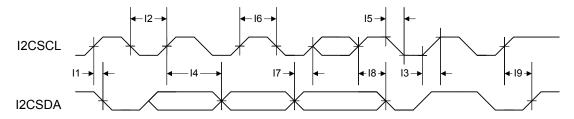
Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
I1 ^a	t _{SCH}	Start condition hold time	36	-	-	system clocks
I2 ^a	t _{LP}	Clock Low period	36	-	-	system clocks
I3 ^b	t _{SRT}	<code>I2CSCL/I2CSDA</code> rise time (V $_{IL}$ =0.5 V to V $_{IH}$ =2.4 V)	-	-	(see note b)	ns
I4 ^a	t _{DH}	Data hold time	2	-	-	system clocks
I5 ^c	t _{SFT}	<code>I2CSCL/I2CSDA</code> 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
19 ^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.

b. t_{ADC}= 1/f_{ADC clock}

c. Specified at a nominal 50 pF load.

Figure 22-2. I²C Timing



22.2.6 Ethernet Controller

Table 22-14. 100BASE-TX Transmitter Characteristics^a

Parameter Name	Min	Nom	Max	Unit
Peak output amplitude	950	-	1050	mVpk
Output amplitude symmetry	0.98	-	1.02	mVpk
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 22-15. 100BASE-TX Transmitter Characteristics (informative)^a

Parameter Name	Min	Nom	Max	Unit
Return loss	16	-	-	dB
Open-circuit inductance	350	-	-	μs

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 22-16. 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 22-17. 10BASE-T Transmitter Characteristics^a

Parameter Name	Min	Nom	Max	Unit
Peak differential output signal	2.2	-	2.8	٧
Harmonic content	27	-	-	dB
Link pulse width	-	100	-	ns

Parameter Name	Min	Nom	Max	Unit
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 22-18. 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 22-19. 10BASE-T Receiver Characteristics

Parameter Name	Min	Nom	Max	Unit
DLL phase acquisition time	-	10	-	ВТ
Jitter tolerance (pk-pk)	30	-	-	ns
Input squelched threshold	500	600	700	mVppd
Input unsquelched threshold	275	350	425	mVppd
Differential input resistance	-	20	-	kΩ
Bit error ratio	-	10 ⁻¹⁰	-	-
Common-mode rejection	25	-	-	V

Table 22-20. 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)	
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 22-21. Ethernet Reference Crystal^a

Name	Value	Condition
Frequency	25.00000	MHz
Frequency tolerance	±50	PPM
Aging	±2	PPM/yr
Temperature stability (-40° to 85°)	±5	PPM
Temperature stability (-40° to 105°)	±5	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
Serious 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 with the following characteristics.

Figure 22-3. External XTLP Oscillator Characteristics

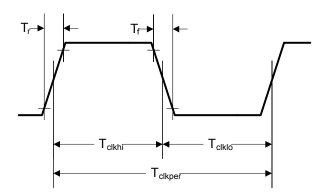


Table 22-22. External XTLP Oscillator Characteristics

Parameter Name	Symbol	Min	Nom	Max	Unit
XTLN Input Low Voltage	XTLN _{ILV}	-	-	8.0	-
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		-	-	0.1	ns

a. IEEE 802.3 frequency tolerance ± 50 ppm.

b. IEEE 802.3 frequency tolerance ± 50 ppm.

22.2.7 Hibernation Module

The Hibernation Module requires special system implementation considerations since it is intended to power-down all other sections of its host device. The system power-supply distribution and interfaces to the device must be driven to 0 V_{DC} or powered down with the same external voltage regulator controlled by $\overline{\text{HIB}}$.

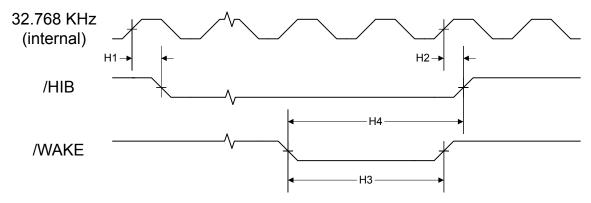
The external voltage regulators controlled by HIB must have a settling time of 250 µs or less.

Table 22-23. Hibernation Module AC Characteristics

Parameter No	Parameter	Parameter Name	Min	Nom	Max	Unit
H1	t _{HIB_LOW}	Internal 32.768 KHz clock reference rising edge to /HIB asserted	-	200	-	μs
H2	t _{HIB_HIGH}	Internal 32.768 KHz clock reference rising edge to /HIB deasserted	-	30	-	μs
H3	t _{WAKE_ASSERT}	/WAKE assertion time	62	-	-	μs
H4	t _{WAKETOHIB}	/WAKE assert to /HIB desassert	62	-	124	μs
H5	t _{XOSC_SETTLE}	XOSC settling time ^a	20	-	-	ms
H6	t _{HIB_REG_WRITE}	Time for a write to non-volatile registers in HIB module to complete	92	-	-	μs
H7	t _{HIB_TO_VDD}	HIB deassert to VDD and VDD25 at minimum operational level	-	-	250	μs

a. This parameter is highly sensitive to PCB layout and trace lengths, which may make this parameter time longer. Care must be taken in PCB design to minimize trace lengths and RLC (resistance, inductance, capacitance).

Figure 22-4. Hibernation Module Timing



22.2.8 Synchronous Serial Interface (SSI)

Table 22-24. SSI Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
S1	t _{clk_per}	SSIC1k cycle time	2	-	65024	system clocks
S2	t _{clk_high}	SSIC1k high time	-	1/2	-	t clk_per
S3	t _{clk_low}	SSIC1k low time	-	1/2	-	t clk_per
S4	t _{clkrf}	SSIC1k rise/fall time	-	7.4	26	ns
S5	t _{DMd}	Data from master valid delay time	0	-	20	ns
S6	t _{DMs}	Data from master setup time	20	-	-	ns
S7	t _{DMh}	Data from master hold time	40	-	-	ns
S8	t _{DSs}	Data from slave setup time	20	-	-	ns
S9	t _{DSh}	Data from slave hold time	40	-	-	ns

Figure 22-5. SSI Timing for TI Frame Format (FRF=01), Single Transfer Timing Measurement

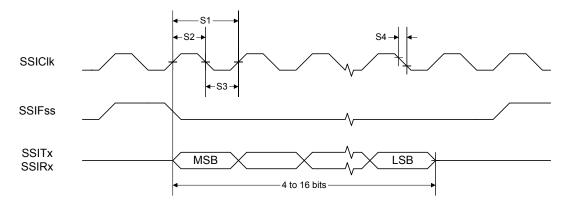
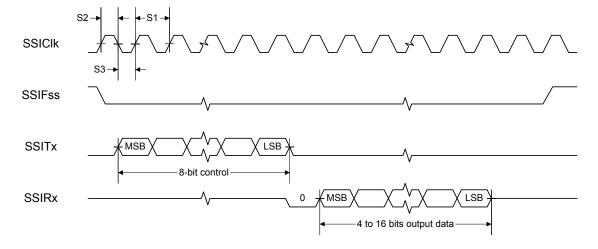


Figure 22-6. SSI Timing for MICROWIRE Frame Format (FRF=10), Single Transfer



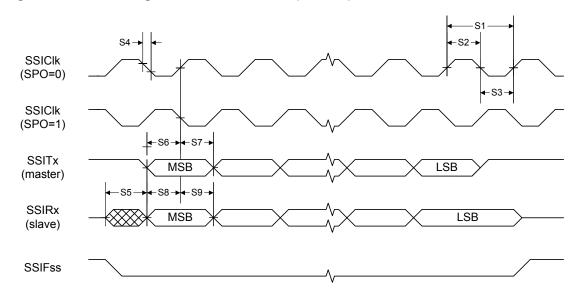


Figure 22-7. SSI Timing for SPI Frame Format (FRF=00), with SPH=1

22.2.9 JTAG and Boundary Scan

Table 22-25. 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}	-	ns
J4	t _{TCK_HIGH}	TCK clock High time	-	t _{TCK}	-	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
J11	TCK fall to Data Valid from High-Z	2-mA drive	-	23	35	ns
t _{TDO_ZDV}		4-mA drive		15	26	ns
_		8-mA drive		14	25	ns
		8-mA drive with slew rate control		18	29	ns
J12	TCK fall to Data Valid from Data Valid	2-mA drive	-	21	35	ns
t _{TDO_DV}		4-mA drive		14	25	ns
		8-mA drive		13	24	ns
		8-mA drive with slew rate control		18	28	ns

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
J13	TCK fall to High-Z from Data Valid	2-mA drive	-	9	11	ns
t TDO DVZ		4-mA drive		7	9	ns
_		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 22-8. JTAG Test Clock Input Timing

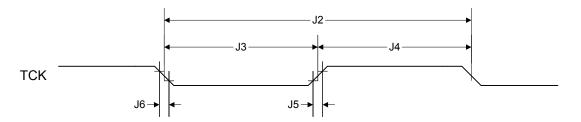


Figure 22-9. JTAG Test Access Port (TAP) Timing

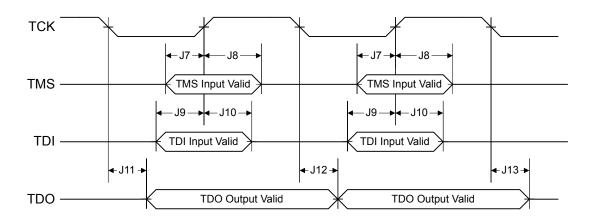
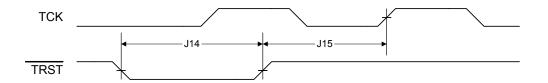


Figure 22-10. JTAG TRST Timing



22.2.10 General-Purpose I/O

Note: All GPIOs are 5 V-tolerant.

Table 22-26. GPIO Characteristics

Parameter	Parameter Name	Condition	Min	Nom	Max	Unit
t _{GPIOR}	GPIO Rise Time (from 20% to 80% of V_{DD})	2-mA drive	-	17	26	ns
		4-mA drive		9	13	ns
		8-mA drive		6	9	ns
		8-mA drive with slew rate control		10	12	ns
t _{GPIOF}	GPIO Fall Time (from 80% to 20% of V _{DD})	2-mA drive	-	17	25	ns
		4-mA drive		8	12	ns
		8-mA drive		6	10	ns
		8-mA drive with slew rate control		11	13	ns

22.2.11 Reset

Table 22-27. Reset Characteristics

Parameter No.	Parameter	Parameter Name	Min	Nom	Max	Unit
R1	V _{TH}	Reset threshold	-	2.0	-	٧
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
R7	T _{IRHWR}	Internal reset timeout after hardware reset (RST 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)	-	-	250	ms
R11	T _{MIN}	Minimum RST pulse width	2	-	-	μs

a. 20 * t _{MOSC_per}

Figure 22-11. External Reset Timing (RST)

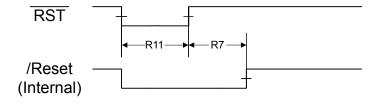


Figure 22-12. Power-On Reset Timing

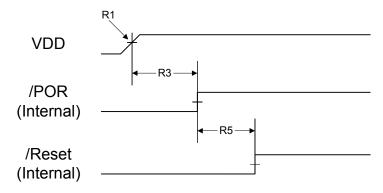


Figure 22-13. Brown-Out Reset Timing

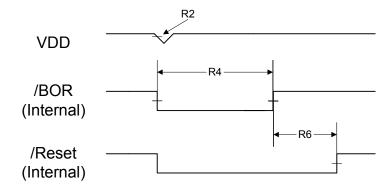


Figure 22-14. Software Reset Timing

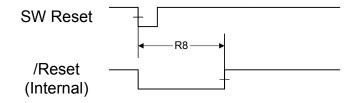
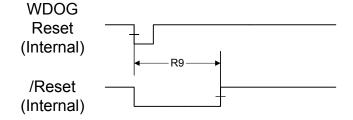
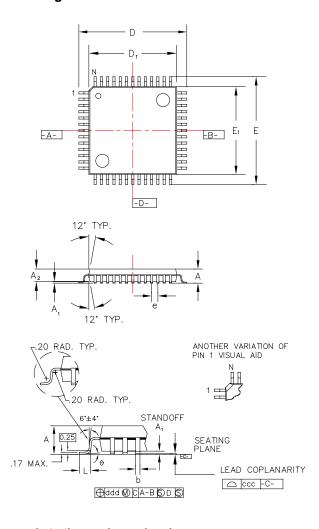


Figure 22-15. Watchdog Reset Timing



23 Package Information

Figure 23-1. 100-Pin LQFP Package

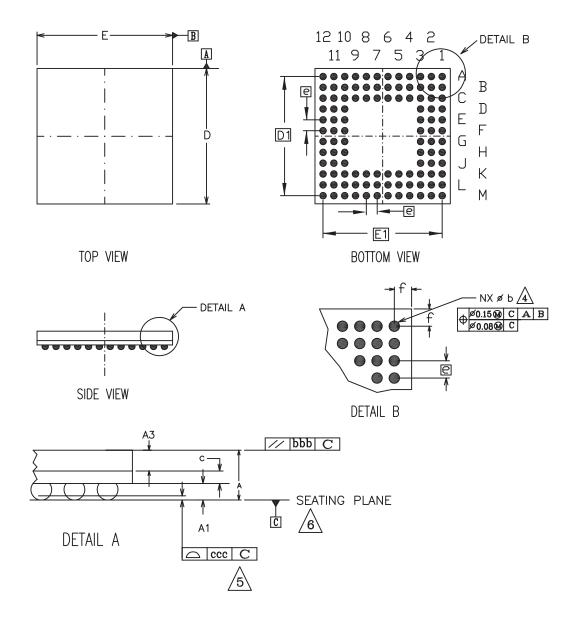


Note: The following notes apply to the package drawing.

- 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.

Body +2.00 mm	Footprint, 1.4 mm	package thickness
Symbols	Leads	100L
Symbols	Leaus	TOOL
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 Refer	ence Drawing	MS-026
Variation [Designator	BED

Figure 23-2. 108-Ball BGA Package



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.
- A EXCEPT DIMENSION b.

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 C 0.28 0.32 0.36 D 9.85 10.00 10.15 D1 8.80 BSC S E1 8.80 BSC S bb 0.43 0.48 0.53 bbb .20 3 ddd .12 3 3 f - 0.60 - M 12 1 1 n 108 1 1 REF: JEDEC MO-219F 1 1				
A1 0.29 0.34 0.39 A3 0.65 0.70 0.75 c 0.28 0.32 0.36 D 9.85 10.00 10.15 D1 8.80 BSC E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	Symbols	MIN	NOM	MAX
A3 0.65 0.70 0.75 c 0.28 0.32 0.36 D 9.85 10.00 10.15 D1 8.80 BSC E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	Α	1.22	1.36	1.50
c 0.28 0.32 0.36 D 9.85 10.00 10.15 D1 8.80 BSC E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	A1	0.29	0.34	0.39
D 9.85 10.00 10.15 D1 8.80 BSC E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	A3	0.65	0.70	0.75
D1 8.80 BSC E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	С	0.28	0.32	0.36
E 9.85 10.00 10.15 E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	D	9.85	10.00	10.15
E1 8.80 BSC b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	D1	8	.80 BS	С
b 0.43 0.48 0.53 bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	Е	9.85	10.00	10.15
bbb .20 ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	E1	8	.80 BS	С
ddd .12 e 0.80 BSC f - 0.60 - M 12 n 108	b	0.43	0.48	0.53
e 0.80 BSC f - 0.60 - M 12 n 108	bbb		.20	
f - 0.60 - M 12 n 108	ddd		.12	
M 12 n 108	е	0	.80 BS	С
n 108	f	-	0.60	-
	М		12	
REF: JEDEC MO-219F	n		108	
	REF: J	EDEC	MO-2	19F

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 346 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 566).

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

								1							
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
	Control 400F.E000														
DID0, type	e RO, offset	0x000, res	set -												
		VER										ASS			
				JOR							MII	NOR			
PBORCTL	L, type R/W,	offset 0x0	30, reset 0:	x0000.7FFI)										
														DODIOD	
LDODOTI	5 DAM	- 55 4 0 - 0	044 0-											BORIOR	
LDOPCIL	., type R/W,	onset uxu	34, reset ux	XUUUU.UUUU 											
												\/4	\DJ		
RIS. type	RO, offset (0x050, rese	et 0x0000.0	000								V	100		
rtio, type	Ito, onoct	7,000,1000													
									PLLLRIS					BORRIS	
IMC, type	R/W, offset	0x054, res	set 0x0000.	0000											
		,													
									PLLLIM					BORIM	
MISC, typ	e R/W1C, o	ffset 0x058	, reset 0x0	000.000											
									PLLLMIS					BORMIS	
RESC, typ	oe R/W, offs	et 0x05C,	reset -												
										LDO	SW	WDT	BOR	POR	EXT
RCC, type	R/W, offse	t 0x060, re	set 0x0780												
				ACG		SYS	SDIV		USESYSDIV						
		PWRDN		BYPASS			X	ΓAL		OSC	SRC			IOSCDIS	MOSCDIS
PLLCFG,	type RO, of	fset 0x064	, reset -												
						_									
DCC2 for	na DAN affa	-4.0×070 =		0.2040		F							R		
USERCC2	oe R/W, offs	et uxu7u, r	eset uxu76	0.2010	eve	DIV2									
USERCCZ		PWRDN2		BYPASS2		DIVZ				OSCSRC2					
DSLPCI K	CFG, type		0x144 res							J0001102					
JOLI OLI	o, type	, 511361		0.0100.		ORIDE									
					SOBIV					DSOSCSRO	;				
DID1, type	e RO, offset	0x004, res	set -												
	VE				F/	AM					PAF	RTNO			
	PINCOUNT								TEMP			KG	ROHS	QL	JAL
DC0, type	RO, offset	0x008, res	et 0x00FF.0	007F											
							SRA	AMSZ							
							FLA	SHSZ		_		_			
DC1, type	RO, offset	0x010, res	et 0x0101.3	33FF											
							CAN0								ADC
	MINS					MAXAI	DCSPD	MPU	HIB	TEMPSNS	PLL	WDT	SWO	SWD	JTAG
DC2, type	RO, offset	0x014, res	et 0x070F.1	1013											
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0
			I2C0								SSI0			UART1	UART0
	RO, offset	0x018, res	et 0x8F0F.3												
32KHZ				CCP3	CCP2	CCP1	CCP0					ADC3	ADC2	ADC1	ADC0
		C2PLUS	C2MINUS	C10	C1PLUS	C1MINUS	C00	COPLUS	COMINUS						

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
DC4, type	e RO, offset	0x01C, res	set 0x5100.	007F											
	EPHY0		EMAC0				E1588								
									GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
RCGC0.1	type R/W, of	fset 0x100	reset 0x00	000040											
	.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		, 10001 0201				CAN0								ADC
						MANYA			LUD			MDT			ADC
						MAXA	DCSPD		HIB			WDT			
SCGC0, 1	type R/W, off	set 0x110	, reset 0x00	000040											
							CAN0								ADC
						MAXA	DCSPD		HIB			WDT			
DCGC0,	type R/W, of	fset 0x120	, reset 0x00	000040											
							CAN0								ADC
						MAXA	DCSPD		HIB			WDT			
PCGC1	type R/W, of	Feat Ov104	rocat OvOC	2000000											
	., ρο 1./ νν , ΟΠ	361 VA 104	, 16361 0400		COMPS	COMPA	COMPA					TIMEDO	TIMEDO	TIMED4	TIMEDO
			1000		COMP2	COMP1	COMP0				0010	TIMER3	TIMER2	TIMER1	TIMER0
			12C0								SSI0			UART1	UART0
SCGC1, 1	type R/W, off	set 0x114	reset 0x00	000000											
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0
			I2C0								SSI0			UART1	UART0
DCGC1,	type R/W, of	fset 0x124	, reset 0x00	000000											
,					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0
			I2C0				22 3				SSI0	2.13		UART1	UART0
DCCC2	tura D/M af	Fact 0x400		1000000							0010			O/ II ()	0711110
RUGUZ,	type R/W, of	Set ux tuo		1000000				1				I			
	EPHY0		EMAC0												
									GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
SCGC2, 1	type R/W, off	set 0x118	, reset 0x00	000000											
	EPHY0		EMAC0												
									GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
DCGC2,	type R/W, of	set 0x128	, reset 0x00	000000											
	EPHY0		EMAC0												
	Little		LIVINGO						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
20000									01100	01 101	GITOL	GI IOD	01 100	OI IOB	OFTOA
SKCKU, I	type R/W, off	Set uxu4u	, reset uxuu	000000											
							CAN0								ADC
									HIB			WDT			
SRCR1, t	type R/W, off	set 0x044	reset 0x00	000000											
					COMP2	COMP1	COMP0					TIMER3	TIMER2	TIMER1	TIMER0
			I2C0								SSI0			UART1	UART0
SRCR2 1	type R/W, off	set 0x048	reset 0x00	000000							1			1	
, . ,,	EPHY0		EMAC0												
	LITTO		LIVIACO						GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
									GFIUG	GFIUF	GFIUE	GFIOD	GFIUC	GFIUB	GPIUA
	ation Mo														
Base 0x	400F.C000														
HIBRTCO	C, type RO, o	ffset 0x00	0, reset 0x0	0000.0000											
							RT	CC							
							RT	CC							
HIBRTON	M0, type R/W	. offset 0×	004. reset 0	xFFFFFF	FF										
	, .,pc 10 W	,	, 10361 0				DT	CM0							
							KI	CM0							
HIBRTC	M1, type R/W	, offset 0x	008, reset 0	xFFFF.FF	FF										
							RTO	CM1							
							RTO	CM1							
HIBRTCL	D, type R/W	, offset 0x	00C, reset ()xFFFF.FF	FF										
							RT	CLD							
								CLD							

0.4	20	00	00	0.7	00	05	0.4	T 00	00	04	00	10	40	47	40
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 3	18	17	16 0
			, reset 0x0		10	3	0	1 '	0	3				'	
1112012, ()	, pc 1011, 0		, 10001 020												
								VABORT	CLK32EN	LOWBATEN	PINWEN	RTCWEN	CLKSEL	HIBREQ	RTCEN
HIBIM. tvp	e R/W. offs	set 0x014.	reset 0x000	00.0000								_		-	
, .,,		,													
												EXTW	LOWBAT	RTCALT1	RTCALT
HIBRIS, tv	pe RO. off	set 0x018.	reset 0x000	00.000								l		_	
-,,	, .	,													
												EXTW	LOWBAT	RTCALT1	RTCALT
HIBMIS, ty	pe RO, off	set 0x01C,	reset 0x00	00.0000				-				1			
												EXTW	LOWBAT	RTCALT1	RTCALT
HIBIC, typ	e R/W1C, c	offset 0x02	0, reset 0x0	0000.0000											
												EXTW	LOWBAT	RTCALT1	RTCALT
HIBRTCT,	type R/W,	offset 0x02	24, reset 0x	0000.7FFF											
							Т	RIM							
HIBDATA,	type R/W,	offset 0x03	30-0x12C, r	eset 0x0000	0.0000										
							F	RTD							
							F	RTD							
Internal	Memor	v													
			Control	Offset)											
Base 0x4	_			,											
FMA, type	R/W, offse	t 0x000, re	set 0x0000	.0000											
	,	,												OFF	SET
							OF	FSET				I			
FMD, type	R/W, offse	t 0x004, re	set 0x0000	.0000											
							D	ATA							
							D	ATA							
FMC, type	R/W, offse	t 0x008, re	set 0x0000	.0000											
							WF	RKEY							
												COMT	MERASE	ERASE	WRITE
FCRIS, typ	e RO, offs	et 0x00C, i	reset 0x000	0.0000											
														PRIS	ARIS
FCIM, type	R/W, offs	et 0x010, r	eset 0x0000	0.0000											
														PMASK	AMASK
FCMISC, ty	ype R/W10	, offset 0x	014, reset (0x0000.0000)										
														PMISC	AMISC
Internal	Memor	v													
			n Contro	ol Offset))										
Base 0x4				2.1000)											
			0, reset 0x3	31											
,	-, ,														
											US	EC			
FMPRE0 1	type R/W	offset 0x13	0 and 0x20	0, reset 0xF	FFF.FFFF			1				-			
, •	,			,			READ	ENABLE							
								ENABLE							

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
						9	0		0	5	4	1 3			U
FMPPEU,	type R/W, o	onset ux134	4 and ux4u	u, reset uxi	·FFF.FFFF		PPOO	ENIADLE							
								ENABLE							
IISED DI	BG, type R/\	N offect Ox	1D0 rocot	0.000	·		FROG_	ENABLE							
NW	BG, type K/	v, onset o	CIDO, Teset	UXFFFF.FF	FE			DATA							
INVV						D4	ATA	DAIA						DBG1	DBG0
IISED DI	EG0, type R	/M offect ()v1E0_roso	+ 0×EEEE E		<i>DF</i>	NA .							DBG1	DBG0
NW	EGU, type K	vv, onset c	JX IEU, IESE	UXFFFF.F	FFF			DATA							
INVV							DA	ATA							
USER RI	EG1, type R	/W offset ()y1F4 rese	t OxFFFF F	FFF										
NW		, 0001	.x.=.,	. •				DATA							
							DA	ATA							
FMPRE1.	, type R/W, o	offset 0x20	4. reset 0xF	FFF.FFFF											
	, , , ,		,				READ	ENABLE							
								ENABLE							
FMPRE2,	, type R/W, o	offset 0x20	8, reset 0xF	FFF.FFFF											
	,						READ	ENABLE							
								ENABLE							
FMPRE3,	, type R/W, o	offset 0x20	C, reset 0x	FFFF.FFFF											
							READ_	ENABLE							
							READ_	ENABLE							
FMPPE1,	type R/W, c	offset 0x404	4, reset 0xF	FFF.FFFF											
							PROG_	ENABLE							
							PROG_	ENABLE							
FMPPE2,	type R/W, c	ffset 0x40	8, reset 0xF	FFF.FFFF											
							PROG_	ENABLE							
							PROG_	ENABLE							
FMPPE3,	type R/W, c	ffset 0x40	C, reset 0xl	FFFF.FFFF											
							PROG_	ENABLE							
							PROG_	ENABLE							
GPIO PO GPIO PO GPIO PO GPIO PO GPIO PO GPIO PO	al-Purpos ort A base: ort B base: ort C base: ort D base: ort E base: ort F base: ort G base:	0x4000.4 0x4000.5 0x4000.6 0x4000.7 0x4002.4 0x4002.5 0x4002.6	000 000 000 000 000 000												
GPIODAT	ΓA, type R/W	, offset 0x	000, reset 0	0x0000.000	0										
											D	ATA			
GPIODIR	, type R/W,	offset 0x40	0, reset 0x	0000.0000							Г	DIR			
GPIOIS 1	type R/W, of	fset 0x404	reset flyfin	000.0000				1							
J. 1010, I	., po 1411, 01														
												S			
GPIOIBF	, type R/W,	offset 0x40	8. reset 0×1	0000,0000				1							
	, ., .,		.,												
											II	I BE			
GPIOIFV	type R/W, o	offset 0x40	C. reset 0x1	0000.0000				1				-			
	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		, 1 2 3 3 4 3 4									_ ≣V			
											ll .	_ v			

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
GPIOIM, ty	pe R/W, of	fset 0x410	, reset 0x0	000.0000											
											II.	1E			
GPIORIS, 1	type RO, o	ffset 0x414	l, reset 0x0	0000.0000				I							
											R	ls.			
GPIOMIS	type RO. o	ffset 0x418	3, reset 0x0	0000.0000											
,	J														
											M	IS			
GPIOICR,	type W1C,	offset 0x4	1C, reset 0	x0000.0000											
											I	C			
GPIOAFSE	EL, type R/	W, offset 0	x420, reset	t -											
											AFS	251			
GDIUDBar	R type R/M	/ offeet no	500 reset (0x0000.00F	F						AFS	JEL			
C. IODINZI	., type 100	., 511361 01	-50, 16361												
											DR	l 2V2			
GPIODR4F	R, type R/W	, offset 0x	504, reset (0x0000.000)										
											DR	2V4			
GPIODR8F	R, type R/W	/, offset 0x	508, reset (0x0000.000)										
CRICORR	tuno B/M	offoot OvE	00 reset 0	x0000.0000							DR				
GPIOODK,	, type K/vv,	onset uxo	oc, reset o	.0000.0000											
											10	DE			
GPIOPUR,	type R/W,	offset 0x5	10, reset -												
											Pl	JE			
GPIOPDR,	type R/W,	offset 0x5	14, reset 0	x0000.0000											
0010010											P[DE			
GPIOSLR,	type R/W,	Offset UX5	18, reset ux	k0000.0000				1							
											SF	 RL			
GPIODEN.	type R/W,	offset 0x5	1C. reset -					1							
											DE	ΞN			
GPIOLOCI	K, type R/V	V, offset 0x	520, reset	0x0000.000	1										
								OCK							
							LC	OCK							
GPIOCR, t	ype -, offse	et 0x524, re	eset -												
											C	R			
GPIOPerin	hID4 type	RO offset	0xFD0 re-	set 0x0000.	0000							11			
1 611p	Б, гуре	. 10, 011361	. JAI DO, 16:												
											PII	I D4			
GPIOPerip	hID5, type	RO, offset	0xFD4, res	set 0x0000.	0000										
											PII	D5			

24	20	20	00	07	00	05	0.4	20	20	24	20	40	10	47	40
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 3	18	17	16 0
			t 0xFD8, res			9	0	/	0	5	4	3		!	0
эгіогепр	піре, туре	KO, Olisei	TOXEDO, Tes	Set uxuuuu											
											PII	l D6			
GPIOPerip	hID7, type	RO, offset	t 0xFDC, res	set 0x0000	0.0000										
											PII	D7			
GPIOPerip	hID0, type	RO, offset	t 0xFE0, res	et 0x0000	.0061		-								
											PII	D0			
GPIOPerip	hID1, type	RO, offset	t 0xFE4, res	et 0x0000	.0000										
											PII	D1			
GPIOPerip	hID2, type	RO, offset	t 0xFE8, res	set 0x0000	.0018										
											PI	D2			
GPIOPerip	hID3, type	RO, offset	t 0xFEC, res	set 0x0000	0.0001										
											DII	D2			
CDIODOCU	IIDO tuno D	O offeet	0xFF0, rese	t 0v0000 ()00D						PII	D3			
GPIOPCEII	ibu, type K	o, onset	JXFFU, rese	it uxuuuu.t	J00D										
											CI	D0			
GPIOPCell	IID1, type R	O. offset (0xFF4, rese	t 0x0000.0	00F0										
00. 00		, 0001	1,1000												
											CI	I D1			
GPIOPCell	IID2, type R	O, offset (0xFF8, rese	t 0x0000.0	0005										
											CI	D2	1		
GPIOPCell	ID3, type R	O, offset (0xFFC, rese	et 0x0000.	00B1										
											CI	D3			
Timer0 ba Timer1 ba Timer2 ba	I-Purpos ase: 0x400 ase: 0x400 ase: 0x400 ase: 0x400	03.0000 03.1000 03.2000	S												
GPTMCFG	, type R/W,	offset 0x0	000, reset 0	x0000.000	0										
ODT	D 4 =	u - ee		0000	100									GPTMCFG	
GPIMTAM	к, type R/V	v, offset 0	x004, reset	UXU000.00)UU										
												TAAMS	TACMR	Τ^	MR
GPTMTRM	R type P/	N offeet o	x008, reset	0×0000 00	200							IAANO	IACIVIK	IA	IVIIX
C1 11411111	it, type K/V	·, onset 0	, reset	JA0000.00											
												TBAMS	TBCMR	TR	MR
GPTMCTL.	, type R/W.	offset 0x0	00C, reset 0:	x0000.000	0							ı .			
,	. ,,		, ,,,,,,												
	TBPWML	TBOTE		TBE	VENT	TBSTALL	TBEN		TAPWML	TAOTE	RTCEN	TAE	VENT	TASTALL	TAEN
)18, reset 0x			1	1					1			
					CBEIM	СВМІМ	твтоім					RTCIM	CAEIM	CAMIM	TATOIN
GPTMRIS,	type RO, o	ffset 0x01	C, reset 0x0	0000.0000									-		
					CBERIS	CBMRIS	TBTORIS					RTCRIS	CAERIS	CAMRIS	TATORI

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
GPTMMIS	S, type RO,	offset 0x02	20, reset 0x	0000.0000											
					CBEMIS	CBMMIS	TBTOMIS					RTCMIS	CAEMIS	CAMMIS	TATOMIS
GPTMICE	R, type W1C	offset Oxi	124 reset (×0000 000											
	, ,,po	, 011001 071													
					CBECINT	CBMCINT	TRTOCINT					RTCCINT	CAFCINT	CAMCINT	TATOCINT
GPTMTA	II P type P/	W offeet 0	v028 reset	0×0000 FF				 FF (32-hit	mode)			111.0011	071201111	G/ 41101111	
OI TIMITA	iLit, type it	v, onset o	X020, 1636	0.00000.11	11 (10-511	illoue, allu			illoue)						
GPTMTR	II P type P/	W offeet f	v02C rese	t 0×0000 F	FFF		1741								
OI IMITE	ILIX, type IV	vv, onset o	7,020,1636												
							TRI	l Di							
CDTMTA	MATCHE 6	no B/M of	foot 0v020	rooot OvO	000 EEEE /4	6 hit mada			2 hit made						
GPTWIA	MAICHK, ty	pe K/W, O	iset uxusu,	reset uxut	JUU.FFFF (1	6-DIL MOGE			2-bit mode)					
							IAI	VIRL							
GPIMIB	MAICHR, ty	/pe K/vv, o	riset uxu34	, reset uxu	UUU.FFFF			I							
								<u> </u>							
							IBI	VIRL							
GPTMTA	PR, type R/\	v, offset 0	xu38, reset	UX0000.00	UÜ										
											IA	PSR			
GPTMTB	PR, type R/	W, offset 0	x03C, reset	0x0000.00	000			1				1			
											ТВ	PSR			
GPTMTA	PMR, type F	R/W, offset	0x040, res	et 0x0000.0	0000										
											TAP	SMR			
GPTMTB	PMR, type F	R/W, offset	0x044, res	et 0x0000.0	0000										
											TBF	SMR			
GPTMTA	R, type RO,	offset 0x0	48, reset 0>	(0000.FFFF	(16-bit mo	de) and 0x	FFFF.FFFF	(32-bit mo	de)						
							TA	RH							
							TA	RL							
GPTMTB	R, type RO,	offset 0x0	4C, reset 0	x0000.FFF	F										
							TB	RL							
Watch	dog Time	r													
Base 0x	4000.0000														
WDTLOA	D, type R/W	/, offset 0x	000, reset (xFFFF.FFI	FF										
	CABCINT CABACINT STROCKIT ALR. type RW, offset 0x025, reset 0x0000 FFFF (16-bit mode) and 0xFFFFFFF (32-bit mode) TALER! TA														
							WDT	Load							
WDTVAL	UE, type RC), offset 0x	004, reset	0xFFFF.FF	FF										
							WDT	Value							
							WDT	Value							
WDTCTL	, type R/W,	offset 0x00	08, reset 0x	0000.0000											
	,														
														RESEN	INTEN
WDTICR.	type WO. o	ffset 0x00	C, reset -											1	1
			, -				WDT	IntClr							
WDTRIS	type RO of	ffset Oy010	. reset flyfi	000.000											
	, , , , , , , , , , , , , , , , , ,		,												
															WDTRIS
															MPILIO

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 RO, of							l							
															WDTMIS
WDTTES1	T, type R/W,	offset 0x4	118, reset 0	x0000.0000											
							STALL								
WDTLOC	K, type R/W	, offset 0x	C00, reset (0x0000.0000)										
								Lock							
WDTDawin	ahID4 tuma	DO -#4	0×FD0 ===		.000		WDI	Lock							
vvDireii	ohID4, type	KO, oliset	UXFDU, IES	et uxuuuu.u	000										
											P	I ID4			
WDTPerin	phID5, type	RO. offset	0xFD4. res	et 0x0000.0	000						· ·				
		,													
											P	ID5			
WDTPerip	phID6, type	RO, offset	0xFD8, res	et 0x0000.0	000										
											Р	ID6			
WDTPerip	phID7, type	RO, offset	0xFDC, res	set 0x0000.0	0000										
											P	ID7			
WDTPerip	phID0, type	RO, offset	0xFE0, res	et 0x0000.0	005			1							
											_				
											Р	ID0			
WDTPerip	phID1, type	RO, offset	UXFE4, res	et uxuuuu.u	018			I				1			
											P	l ID1			
WDTPerin	phID2, type	RO. offset	0xFF8, res	et 0x0000.0	018						· ·				
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	, 000	- CAL 20, 100		• . •										
											P	I ID2			
WDTPerip	phID3, type	RO, offset	0xFEC, res	set 0x0000.0	001										
											Р	ID3			
WDTPCel	IID0, type R	O, offset 0	0xFF0, rese	t 0x0000.00	0D										
											С	ID0			
WDTPCel	IIID1, type R	O, offset 0	0xFF4, rese	t 0x0000.00	F0			1							
WDTDO-I	IIID2, type R	0 -554.6	D. FF0	4.00000.00	^-							ID1			
WDIPCel	IIID2, type K	O, onset C	JXFF8, rese	t 0x0000.00	U5										
											C	ID2			
WDTPCel	IID3, type R	O. offset (0xFFC, rese	et 0x0000.00	B1										
		,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,												
											C	ID3			
Analog	-to-Digita	al Conv	erter (AD	OC)											
	4003.8000			-,											
ADCACTS	SS, type R/V	V, offset 0:	x000, reset	0x0000.000	0										
												ASEN3	ASEN2	ASEN1	ASEN0
ADCRIS, 1	type RO, off	fset 0x004	, reset 0x00	000.000											
												INR3	INR2	INR1	INR0

												_			
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
ADCIM, ty	ype R/W, off	set 0x008,	reset 0x00	00.0000											
												MACKS	MACKO	MACKA	MASK0
ADCISC	type R/W1C	affaat Ou	000 =====	00000.000								MASK3	MASK2	MASK1	IVIASKU
ADCISC,	type R/W/IC	, onset ux	Juc, reset												
												IN3	IN2	IN1	IN0
ADCOST	AT, type R/W	/1C offset	0x010 res	et OxOOOO (2000							1			
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	, ., ро	,	- CAC 10, 100												
												OV3	OV2	OV1	OV0
ADCEMU	X, type R/W	offset 0x	014, reset 0	x0000.000	0							1			
			,												
	EN	13			Е	M2			EI	M1			EI	M0	
ADCUSTA	AT, type R/W	1C, offset	0x018, res	et 0x0000.0	0000										
												UV3	UV2	UV1	UV0
ADCSSPI	RI, type R/W	, offset 0x	020, reset (0x0000.321	0										
		S	S3			S	S2			S	S1			S	30
ADCPSSI	l, type WO, o	offset 0x02	28, reset -												
												SS3	SS2	SS1	SS0
ADCSAC,	, type R/W, o	offset 0x03	30, reset 0x	0000.0000											
														AVG	
ADCSSM	UX0, type R			et 0x0000.0	000										
			JX7				JX6				JX5				JX4
			JX3			М	JX2			М	JX1			MU	JX0
	TL0, type R/											I =a.			
TS7 TS3	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
						ENDZ	DZ	131	IEI	ENDI	Di	130	IEU	ENDO	DU
ADCSSFI	FO0, type R	O, onset t	JXU40, 1656		100										
										D	ATA				
ADCSSEL	FO1, type R	O offset (NYO68 reset	1 020000 00	100										
7.2000	, . , po	-,													
										D/	ATA				
ADCSSFI	FO2, type R	O, offset 0)x088, reset	t 0x0000.00	000										
									1	D	ATA	1		ı	
ADCSSFI	FO3, type R	O, offset 0	0x0A8, rese	t 0x0000.00	000										
										D	ATA				
ADCSSFS	STAT0, type	RO, offset	t 0x04C, res	set 0x0000.	0100										
			FULL				EMPTY		HF	TR			TP	TR	
ADCSSFS	STAT1, type	RO, offset	t 0x06C, res	set 0x0000.	0100										
			FULL				EMPTY		HF	TR			TP	TR	
ADCSSFS	STAT2, type	RO, offset	t 0x08C, res	set 0x0000.	0100										
			FULL				EMPTY		HF	TR			TP	TR	

31	30	29			26	25	24	23	22	21				17	16
15	14	13	28 12	27	10	9	8	7	6	5	20 4	19	18	1	0
ADCSSFS	STAT3, type	RO, offset	0x0AC, re:	set 0x0000.	.0100										
			-												
			FULL				EMPTY		HF	PTR			TF	TR	
ADCSSM	UX1, type F	R/W, offset 0)x060, rese	et 0x0000.0	000										
		MU				MU	JX2			MU	IX1			ML	JX0
ADCSSM	UX2, type F	R/W, offset (0x080, rese	et 0x0000.00	000							1			
		MU	IX3			MI	JX2			ML	IX1			ML	IXU
ADCSSC	TL1, type R			t 0x0000.00	100	1410	J/L			1410				IVIC	
	, ., ,,	,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,												
TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
ADCSSC	TL2, type R	/W, offset 0	x084, rese	t 0x0000.00	000										
TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
ADCSSM	UX3, type F	R/W, offset (0x0A0, res	et 0x0000.0	000										
ADCCCC	TI 2 4	NAI -554 5	w0.4.4	4 020000 00	202									ML	JXU
ADCSSC	TL3, type R	/vv, offset u	XUA4, rese	T UXUUUU.UU	JU2										
												TS0	IE0	END0	D0
ADCTML	R type R/W	55 4 0 - 4													
		. omset ux1	00. reset 0	X0000.000U)										
ADCIMIL	, type latt	, onset ux1	00, reset 0	 											
Univers	sal Asyn	chronou				rs (UAR)	Гѕ)								LB
Univers UARTO b UART1 b		chronou 000.C000 000.D000	s Recei	vers/Tra		rs (UAR	Гѕ)								LB
Univers UARTO b UART1 b	sal Asyn base: 0x40 base: 0x40	chronou 000.C000 000.D000	s Recei	vers/Tra		rs (UAR	ſs)								LB
Univers UARTO b UART1 b	sal Asyn base: 0x40 base: 0x40	chronou 000.C000 000.D000	s Recei	vers/Tra		rs (UAR)	FE				DA	ATA			LB
Univers UARTO E UART1 E UARTDR,	sal Asyn base: 0x40 base: 0x40	chronou 000.C000 000.D000 offset 0x000	s Recei	0000.0000 OE	nsmitte BE	PE					D/	ATA			LB
Univers UARTO E UART1 E UARTDR,	sal Asyn base: 0x40 base: 0x40 , type R/W,	chronou 000.C000 000.D000 offset 0x000	s Recei	0000.0000 OE	nsmitte BE	PE					D/				
Univers UARTO E UART1 E UARTDR,	sal Asyn base: 0x40 base: 0x40 , type R/W,	chronou 000.C000 000.D000 offset 0x000	s Receivalle (1986), reset 0x0	0000.0000 OE 04, reset 0x	BE	PE					D/	ATA OE	BE	PE	LB
Univers UARTO E UART1 E UARTDR,	sal Asyn base: 0x40 base: 0x40 , type R/W,	chronou 000.C000 000.D000 offset 0x000	s Receivalle (1986), reset 0x0	0000.0000 OE 04, reset 0x	BE	PE					D/		BE	PE	
Univers UARTO E UART1 E UARTDR,	sal Asyn base: 0x40 base: 0x40 , type R/W,	chronou 000.C000 000.D000 offset 0x000	s Receivalle (1986), reset 0x0	0000.0000 OE 04, reset 0x	BE	PE							BE	PE	
Univers UARTO E UARTT E UARTTOR, UARTRSE	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 , type R/W,	chronou 00.C000 000.D000 offset 0x000 R, type RO,	s Receivence of the control of the c	0000.0000 OE 04, reset 0x	BE	PE						OE	BE	PE	
Univers UARTO E UARTO E UARTOR, UARTRSE	sal Asyn base: 0x40 base: 0x40 , type R/W,	chronou 00.C000 000.D000 offset 0x000 R, type RO,	s Receivence of the control of the c	0000.0000 OE 04, reset 0x	BE	PE						OE	BE	PE	
Univers UARTO E UARTT E UARTTOR, UARTRSE	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 , type R/W,	chronou 00.C000 000.D000 offset 0x000 R, type RO,	s Receivence of the control of the c	0000.0000 OE 04, reset 0x	BE	PE		TXFE	RXFF	TXFF		OE	BE	PE	
Univers UARTO E UARTO E UARTO E UARTOR, UARTRSE	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 , type R/W,	chronou 000.C000 000.D000 offset 0x000 R, type RO, R, type WO,	s Received 0x000 offset 0x000 o	0000.0000 OE 04, reset 0x	BE x0000.0000	PE		TXFE	RXFF	TXFF	D/	OE ATA	BE	PE	
Univers UARTO E UARTO E UARTO E UARTOR, UARTRSE	sal Asyn base: 0x40 base: 0x40 type R/W, r/UARTECH	chronou 000.C000 000.D000 offset 0x000 R, type RO, R, type WO,	s Received 0x000 offset 0x000 o	0000.0000 OE 04, reset 0x	BE x0000.0000	PE		TXFE	RXFF	TXFF	D/	OE ATA BUSY	BE	PE	
Univers UARTO E UARTO E UARTOR, UARTRSE UARTRSE UARTRSE	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 type R/W, R/UARTECH type RO, o	chronou 00.C000 000.D000 offset 0x000 R, type RO, R, type WO, ffset 0x018	s Receiv O, reset 0x0 offset 0x0 offset 0x0 , reset 0x0	0000.0000 OE 04, reset 0x 000.0090	BE x0000.0000	PE		TXFE	RXFF	TXFF	D/	OE ATA	BE	PE	
Univers UARTO E UARTO E UARTOR, UARTRSE UARTRSE UARTRSE	sal Asyn base: 0x40 base: 0x40 type R/W, r/UARTECH	chronou 00.C000 000.D000 offset 0x000 R, type RO, R, type WO, ffset 0x018	s Receiv O, reset 0x0 offset 0x0 offset 0x0 , reset 0x0	0000.0000 OE 04, reset 0x 000.0090	BE x0000.0000	PE		TXFE	RXFF	TXFF	D/	OE ATA BUSY	BE	PE	
Univers UARTO E UARTO E UARTOR, UARTRSE UARTRSE UARTRSE	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 type R/W, R/UARTECH type RO, o	chronou 00.C000 000.D000 offset 0x000 R, type RO, R, type WO, ffset 0x018	s Receiv O, reset 0x0 offset 0x0 offset 0x0 , reset 0x0	0000.0000 OE 04, reset 0x 000.0090	BE x0000.0000	PE	FE		RXFF	TXFF	D/	OE ATA BUSY	BE	PE	
UARTO E UARTO E UARTO E UARTO E UARTOR, UARTRSF UARTRSF UARTICPI UARTILPI UARTIBR	sal Asyn base: 0x40 base: 0x40 base: 0x40 type R/W, R/UARTECH type RO, o	chronou 000.C000 000.D000 offset 0x00 R, type RO, R, type WO,	s Receit 0, reset 0x0 offset 0x0 offset 0x0 i, reset 0x0	0000.0000 OE 04, reset 0x 0000.0090 0x0000.0000	BE x0000.0000	PE	FE	TXFE	RXFF	TXFF	D/	OE ATA BUSY	BE	PE	
UARTO E UARTO E UARTO E UARTO E UARTOR, UARTRSF UARTRSF UARTICPI UARTILPI UARTIBR	sal Asyn pase: 0x40 pase: 0x40 pase: 0x40 type R/W, R/UARTECH type RO, o	chronou 000.C000 000.D000 offset 0x00 R, type RO, R, type WO,	s Receit 0, reset 0x0 offset 0x0 offset 0x0 i, reset 0x0	0000.0000 OE 04, reset 0x 0000.0090 0x0000.0000	BE x0000.0000	PE	FE		RXFF	TXFF	D/	OE ATA BUSY	BE	PE	
UNIVERSE UARTO E UARTO E UARTRE UARTRE UARTER, UARTILPI UARTILPI UARTIBR	sal Asyn base: 0x40 base: 0x40 base: 0x40 type R/W, R/UARTECH type RO, o	chronou 000.C000 000.D000 offset 0x00 R, type RO, R, type WO,	s Receit 0, reset 0x0 offset 0x0 offset 0x0 i, reset 0x0	0000.0000 OE 04, reset 0x 0000.0090 0x0000.0000	BE x0000.0000	PE	FE		RXFF	TXFF	D/	OE ATA BUSY DVSR		PE	
UARTO E UARTO E UARTO E UARTO E UARTOR, UARTRSF UARTRSF UARTRSF	sal Asyn base: 0x40 base: 0x40 type R/W, r/UARTECH type RO, o	chronou 000.C000 000.D000 offset 0x000 R, type RO, R, type WO, ffset 0x018 V, offset 0x0	s Receit 0, reset 0x0 offset 0x00 offset 0x00 , reset 0x0 220, reset 0	0000.0000 OE 04, reset 0x 0000.0000 0x0000.0000 0x0000.0000	BE x0000.0000	PE	FE		RXFF	TXFF	D/	OE ATA BUSY DVSR	BE	PE	
UARTO E UARTO E UARTO E UARTO E UARTOR, UARTRSF UARTRSF UARTRSF	sal Asyn base: 0x40 base: 0x40 base: 0x40 type R/W, R/UARTECH type RO, o	chronou 000.C000 000.D000 offset 0x000 R, type RO, R, type WO, ffset 0x018 V, offset 0x0	s Receit 0, reset 0x0 offset 0x00 offset 0x00 , reset 0x0 220, reset 0	0000.0000 OE 04, reset 0x 0000.0000 0x0000.0000 0x0000.0000	BE x0000.0000	PE	FE		RXFF	TXFF	D/	OE ATA BUSY DVSR		PE	

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
UARTCTL	, type R/W,	offset 0x0	30, reset 0	×0000.0300)										
						RXE	TXE	LBE					SIRLP	SIREN	UARTEN
UARTIFLS	S, type R/W	offset 0x0	034, reset 0	x0000.001	2			ı							
											RXIFLSEL			TXIFLSEL	
IIARTIM :	type R/W, o	ffeat NyN38	reset OvO	000 0000							KAIFLSEL			TAIFLSEL	-
OAIXTIIII,	type (att, o	11361 02000	, 16361 020												
					OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM				
UARTRIS	, type RO, o	ffset 0x03	C, reset 0x0	0000.000F											
					OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS				
UARTMIS	, type RO, c	offset 0x04	0, reset 0x0	0000.0000								_			
					OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS				
UARTICR	, type W1C,	offset 0x0	44, reset 0	x0000.0000)										
					OEIC	BEIC	PEIC	FEIC	DTIC	TXIC	DVIC				
ΠΑΡΤΡΑ×	iphID4, type	PO offee	t 0vED0 ===	set Ovene		DEIC	FEIC	FEIC	RTIC	IAIC	RXIC				
UARTPEL	іріпід4, туре	KO, onse	t uxrbu, re		7.0000										
											PI] D4			
UARTPer	iphID5, type	RO, offse	t 0xFD4, re	set 0x0000	0.0000			l							
											PI	D5		1	
UARTPer	iphID6, type	RO, offse	t 0xFD8, re	set 0x0000	0.0000										
											PI	D6			
UARTPer	iphID7, type	RO, offse	t 0xFDC, re	eset 0x000	0.0000										
											PI	D7			
UARTPer	iphID0, type	RO, offse	t 0xFE0, re	set 0x0000	0.0011			I				1			
											PI	D0			
UARTPer	iphID1, type	RO offse	t 0xFF4. re	set 0x0000	0.0000										
	.,,,,,,														
											PI	D1			
UARTPer	iphID2, type	RO, offse	t 0xFE8, re	set 0x0000	0.0018		-								
											PI	D2			
UARTPer	iphID3, type	RO, offse	t 0xFEC, re	set 0x0000	0.0001										
											PI	D3			
UARTPC	ellID0, type	KO, offset	uxFF0, res	et 0x0000.0 	JUOD										
											CI	D0			
UARTEC	ellID1, type	RO. offset	OxFF4 res	et OxOOOO	00F0			<u> </u>			Ci	20			
JAN1 F 06	s i, type	, 011361			-3. 0										
											CI	D1			
UARTPC	ellID2, type	RO, offset	0xFF8, res	et 0x0000.0	0005			l							
											CI	D2			

												1			
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
UARTPCE	IID3, type i	RO, offset	0xFFC, res	et uxuuuu.	00B1										
												ID3			
											C	נטו			
			erface (S	SI)											
	e: 0x4000.														
SSICR0, ty	/pe R/W, of	fset 0x000	, reset 0x00	100.0000											
			00					OPLI	ODO		DE			20	
201001			SC					SPH	SPO	F	RF			SS	
SSICR1, ty	/pe R/W, of	rset uxuu4	, reset 0x00	100.0000								1			
												000	MO	005	1.014
20122	Dan 6											SOD	MS	SSE	LBM
SSIDR, typ	DE R/W, Offs	set uxuu8,	reset 0x000	10.0000											
								1							
20122 /							D.	ATA							
SSISK, typ	ie KU, ottse	et uxuuc, i	reset 0x000	0.0003											
											DOV	DEE	DNE	TNE	TEE
2010222											BSY	RFF	RNE	TNF	TFE
SSICPSR,	type R/W, c	offset UXU	10, reset 0x	J000.0000											
											CDC	DVSR			
00004 6	- DAM -#-	-4.004.4		0.000							CPS	DVSK			
SSIIM, type	e R/W, offse	et UXU14, i	eset 0x000	J.0000											
												TXIM	RXIM	DTIM	RORIM
CCIDIC 6	na DO affa	-4.0×040		0.000								I Alivi	KAIIVI	RTIM	KOKIIVI
SSIKIS, IY	pe KO, ons	et uxu 16,	reset 0x000	0.0008											
												TXRIS	RXRIS	RTRIS	RORRIS
CCIMIC 64	no BO offe	ot 0v01C	reset 0x000	00.000								TARIS	IXXIXIO	KIKIS	KOKKIO
SSIIVIIS, LY	pe KO, ons	et uxu ic,	Teset uxuut	70.0000											
												TXMIS	RXMIS	RTMIS	RORMIS
CCUCD 4.	== W4C =f	f4 0×020		200 0000								TAIVIIO	IXXIVIIO	KTIVIIO	KOKWIC
SSIICK, ty	pe w ic, oi	ISEL UXUZU	, reset 0x00	,00.000											
														RTIC	RORIC
CCIDorinhi	ID4 tupo B	O offeet (VEDO rocci	t 0×0000 0	200									KIIC	KOKIC
SSIPERIPRI	D4, type K	O, onset u	xFD0, reset	L OXOGOO.OC	J00										
											D	ID4			
SSIDorinhi	IDE type P	O offect (xFD4, reset	t 0×0000 0	200						•	104			
33ir eripiii	Do, type K	O, Oliset u	AI D4, 1636		J00										
											P	I ID5			
SSIPerinhl	ID6 type R	O offset (xFD8, reset	t 0×0000 00	200			1				.50			
COII CIIPIII	Do, type it	0, 011001 0	JAI 20, 1000												
											P	I ID6			
SSIPerinhl	ID7 type R	O offset (xFDC, rese	t 0x0000 0	000			1				.50			
- C Cripin	, .ypc it.	_,													
											P	I ID7			
SSIPerinhl	ID0. tvne R	O. offset f	xFE0, reset	t 0x0000 nr)22			1			•				
	, ., po 10	_, 5551 0			_										
											P	I ID0			
SSIPerinhl	ID1, type R	O. offset f	xFE4, reset	t 0x0000 nr	000			1			•	-			
Sen onpili	, type K	_, 511361 0													
											P	l ID1			
SSIPerinhi	ID2 type P	O offeet o	xFE8, reset	t OxOOOO O	118			1							
Jon empili	, type K	., J.11361 U	=0, 16361	. 5,5550.00											
											D	ID2			
											Р	104			

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
SSIPerip	hID3, type F	RO, offset (0xFEC, rese	et 0x0000.00	001										
20120 11					_						PI	D3			
SSIPCell	ID0, type R0	D, offset 03	(FFU, reset	UXUUUU.000	טו										
											CI	D0			
SSIPCell	ID1, type R0	O offset ()	FF4 reset	0x0000 005	:n						01				
0011 0011	ID 1, type itt	, 011301 07	11 4, 10001												
											CI	D1			
SSIPCell	ID2, type R0	D, offset 0	rF8, reset	0x0000.000	5										
											CI	D2			
SSIPCell	ID3, type R0	O, offset 0	(FFC, reset	0x0000.00E	31										
											CI	D3			
	ntegrated	I Circuit	(I ² C) Int	erface											
I ² C Ma	ster														
I2C Mas	ster 0 base	: 0x4002.	0000												
I2CMSA,	type R/W, o	offset 0x00	0, reset 0x0	0000.0000											
											SA				R/S
I2CMCS,	type RO, of	ffset 0x004	, reset 0x00	000.0000											
									DUODOV	IDLE	ADDLOT	DATAOK	ADDAOK	EDDOD	DUIOV
IDOMOS	ture WO e	ffe et 0×00	1 ==== 1 0×0	000 0000					BUSBSY	IDLE	ARBLST	DATACK	ADRACK	ERRUR	BUSY
IZCIVICS,	type WO, o	iiset uxuuz	, reset uxu	1											
												ACK	STOP	START	RUN
I2CMDR,	type R/W, o	offset 0x00	8, reset 0x0	0000.0000								1			
,															
											DA	ATA			
12CMTPF	R, type R/W,	offset 0x0	0C, reset 0	x0000.0001											
											TI	PR			
12CMIMR	, type R/W,	offset 0x0	10, reset 0x	0000.0000											
															IM
I2CMRIS	, type RO, o	ffset 0x014	4, reset 0x0	0000.0000											
100044410		£54-004	0 4 0 0												RIS
IZCIVINIS	, type RO, o	DIISET UXU1	o, reset ux0	0000.0000											
															MIS
12CMICP	, type WO, o	offset 0x01	C. reset fly	0000.0000											IVIIO
LOWINGK	, .,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		_, 1036t UX												
															IC
I2CMCR.	type R/W, o	offset 0x02	0, reset 0x0	0000.0000											
,	,														
										SFE	MFE				LPBK

			1												
31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	6	21 5	20 4	19 3	18	17	16 0
	itegrated				10	9	0	,	0	5	4	3		'	U
Inter-in	_	Circuit	(I-C) Into	егтасе											
	ve re 0 base: (กรุสกกร ก	800												
	type R/W,			20000 0000											
IZCSUAR	t, type R/vv,	onset uxu	ou, reset ux												
												OAR			
ISCSCSB	, type RO, c	offeet OvOC	M reset fly(0000 0000								OAIT			
IZOOOOIK	, type ito, c	mset oxot	74, 16361 070												
													FBR	TREQ	RREQ
12CSCSR	, type WO,	offset OxO	N4 reset Ny	0000 0000									TBIX	TILLE	TITLE
12000011	, , , , , , ,	JIIGOT OXO	U-1, 10001 UX												
															DA
I2CSDR. 1	type R/W, o	ffset 0x00	8. reset 0x0	000.0000											
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		.,												
											DA	I ATA			
I2CSIMR.	type R/W, o	offset 0x00	OC, reset 0x	0000.0000				1							
															DATAIM
I2CSRIS,	type RO, of	fset 0x010), reset 0x0	000.000											
															DATARIS
I2CSMIS,	type RO, of	ffset 0x01	4, reset 0x0	000.0000											
															DATAMIS
I2CSICR,	type WO, o	ffset 0x01	8, reset 0x0	0000.0000											
															DATAIC
Contro	ller Area	Netwo	rk (CAN)	Module											
CAN0 ba	ase: 0x400	4.0000													
CANCTL,	type R/W,	offset 0x0	00, reset 0x	0000.0001											
								Test	CCE	DAR		EIE	SIE	IE	INIT
CANSTS,	type R/W,	offset 0x0	04, reset 0x	0000.0000											
								BOff	EWarn	EPass	RxOK	TxOK		LEC	
CANERR,	, type RO, o	ffset 0x00	08, reset 0x0	0000.0000											
RP				REC							TI	EC			
CANBIT, 1	type R/W, o	ffset 0x00	C, reset 0x0	0000.2301											
		T00			то	4			DA/				20		
CANIDIT	tuno BC = C	TSeg2	. rocat 0:-00	100 0000	18	eg1		l S.	JW			ВІ	RP		
CANIN I, 1	type RO, of	iset uxu10	, reset uxuu	,00.0000											
							1.	ntld							
CANTET	type R/W, c	offeat None	A reset for	0000 0000			II	iuu							
JANIOI,	type rave, c	moet UXU	ı→, ıeset uXt	JJUU.UUUU											
								Rx	7	-x	LBack	Silent	Basic		
CANRED	F type R/M	offeet Nu	018 reset 0	0x0000.0000				1 1/4	<u> </u>	^	LDack	J GHEIR	Dasic		
SAMBINE	-, type rave	, Justi ux	o io, ieset u	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,											
													PE	RPE	
												L	DF.	VI L	

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	17	0
)x020, reset										_		
			,												
Busy												MN	IUM		
CANIF2CI	RQ, type R/	W, offset 0	x080, reset	0x0000.00	01										
Busy												MN	IUM		
CANIF1CI	MSK, type I	R/W, offset	0x024, res	et 0x0000.0	000										
								WRNRD	Mask	Arb	Control	ClrIntPnd	NewDat	DataA	DataE
CANIF2CI	VISK, type I	R/W, offset	0x084, res	et 0x0000.0	1000							1			
								WRNRD	Mask	Arb	Control	ClrIntPnd	NewDat	DataA	Datal
CANIF1CI	MSK type I	R/W offset	0x024, res	et OxOOOO (1000			WININD	IVIASK	Aib	Control	Cirilla IId	Newbat	DataA	Datat
		, 511361		. 0,0000.0											
								WRNRD	Mask	Arb	Control		TxRqst	DataA	Data
CANIF2CI	MSK, type I	R/W, offset	0x084, res	et 0x0000.0	0000			1							
								WRNRD	Mask	Arb	Control		TxRqst	DataA	Data
CANIF1M	SK1, type F	R/W, offset	0x028, rese	et 0x0000.F	FFF										
							N	1sk							
CANIF2M	SK1, type F	R/W, offset	0x088, rese	et 0x0000.F	FFF										
0 4 1 W = 4 1 4							IV.	1sk							
CANIF1M	SK2, type i	R/W, offset	0x02C, res	et 0x0000.F 	-FFF										
MXtd	MDir								Msk						
		R/W. offset	0x08C, res	et 0x0000.F	FFF				WOK						
-,	, , , , , , , , , , , ,	, 0001													
MXtd	MDir								Msk						
CANIF1A	RB1, type F	R/W, offset	0x030, rese	et 0x0000.0	000										
								ID					'		
CANIF2A	RB1, type F	R/W, offset	0x090, rese	et 0x0000.0	000										
							- 1	ID							
CANIF1AF	RB2, type F	R/W, offset	0x034, rese	et 0x0000.0	000										
	10.1														
MsgVal	Xtd	Dir	• •••		•••				ID						
CANIF2AI	KB2, type F	t/W, offset	0x094, rese	et 0x0000.0	υ00										
MsgVal	Xtd	Dir							ID						
			0x038, rese	ot 0v0000 0	000				טו						
OMINIT HIN	oı∟, type r	urv, onset	UAUJO, TESE	. 0.0000.0	000										
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	EoB					DI	_C	
			0x098, rese				401					<u> </u>		-	
	, ., po 1	, 551	3,												
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	EoB					DI	_C	
		W, offset 0	x03C, reset	0x0000.00	00			1				1			
							D	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
				0x0000.000							·		_		
-,	-, t yp 0 10 1	.,													
							D	ata							
CANIF1DE	31, type R/\	N, offset 0:	x044, reset	0x0000.000	10										
		•	,												
							D	ata							
CANIF1DE	32, type R/\	N, offset 0:	x048, reset	0x0000.000	10										
							D	ata				1			
CANIF2DA	\1, type R/\	N, offset 0:	x09C, reset	0x0000.000	00										
							D	ata							
CANIF2DA	\2, type R/\	N, offset 0:	x0A0, reset	0x0000.000	00										
							D	ata							
CANIF2DE	31, type R/\	N, offset 0:	x0A4, reset	0x0000.000	00										
							D	ata							
CANIF2DE	32, type R/\	N, offset 0:	x0A8, reset	0x0000.000	00										
							D	ata							
CANTXRO	1, type RO	, offset 0x	100, reset 0	x0000.0000)										
							Tx	Rqst				•			
CANTXRO	2, type RO	, offset 0x	104, reset 0	x0000.0000)										
							Tx	Rqst							
CANNWD	A1, type R0), offset 0	(120, reset	0x0000.000	0										
							Ne	wDat							
CANNWD	A2, type R0	O, offset 0x	(124, reset	0x0000.000	0										
							Ne	wDat							
CANMSG1	IINT, type F	RO, offset	0x140, rese	t 0x0000.00	00										
							Int	Pnd							
CANMSG2	ZINT, type F	RO, offset (0x144, rese	t 0x0000.00	00										
							Int	Pnd							
CANMSG1	IVAL, type	RO, offset	0x160, res	et 0x0000.0	000										
							Ms	gVal							
CANMSG2	VAL, type	RO, offset	0x164, res	et 0x0000.0	000										
							Ms	gVal							
	t Contro	oller													
Etherne															
	004.8000														
MACRIS, t	ype RO, of	fset 0x000	, reset 0x00	000.0000											
									PHYINT	MDINT	RXER	FOV	TXEMP	TXER	RXINT

								1							
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
MACIACK,	, type W1C	offset 0x0	000, reset 0	0x0000.0000				1							
									PHYINT	MDINT	RXER	FOV	TXEMP	TXER	RXINT
MACIM, ty	pe R/W, off	set 0x004,	reset 0x00	000.007F								ı			
									PHYINTM	MDINTM	RXERM	FOVM	TXEMPM	TXERM	RXINTM
MACRCTL	, type R/W,	offset 0x0	08, reset 0	x0000.0008								ı			
											RSTFIFO	BADCRC	PRMS	AMUL	RXEN
MACTCTL	, type R/W,	offset 0x0	0C, reset 0	x0000.0000								1			
											DUPLEX		CRC	PADEN	TXEN
MACDATA	, type RO,	offset 0x01	0, reset 0x	(0000.0000											
								DATA							
							RXI	DATA							
MACDATA	, type WO,	offset 0x0	10, reset 0:	x0000.0000											
								DATA							
							TXI	DATA							
MACIA0, t	ype R/W, of	fset 0x014													
				OCT4								OCT3			
				OCT2							MAC	OCT1			
MACIA1, t	ype R/W, of	fset 0x018	, reset 0x0	000.0000								ı			
				OCT6							MAC	OCT5			
MACTHR,	type R/W, o	offset 0x01	C, reset 0x	(0000.003F											
												THR	ESH		
MACMCTL	_, type R/W,	offset 0x0	20, reset 0	0x0000.0000	1										
										REGADR				WRITE	START
MACMDV,	type R/W, o	offset 0x02	4, reset 0x	0800.0000											
											D	IV			
MACMTXE), type R/W	, offset 0x0)2C, reset (0x0000.0000)										
							MI	DTX							
MACMRXI	D, type R/W	, offset 0x0	030, reset (0x0000.0000)										
							MI	DRX							
MACNP, ty	pe RO, offs	set 0x034,	reset 0x00	00.000											
												NI	PR		
MACTR, ty	pe R/W, of	fset 0x038,	reset 0x00	000.0000											
															NEWTX
MACTS, ty	pe R/W, of	set 0x03C	, reset 0x0	000.000											
															TSEN
Etherne	t Contro	ller													
MII Man	agemen	t													
	R/W, addre		eset Oy310	0											
	LOOPBK				ISO	PANEC	DUPLEX	COLT							
KESEI	LOOPBK	OFEEDSL	ANEGEN	FWKDN	130	RAINEG	DUPLEX	LOLI							

100 100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR7, type R0, address 0x17, reset 0x7549 MR2, type R0, address 0x2, reset 0x7527 OUI[\$1:9] MR3, type R0, address 0x2, reset 0x7287 OUI[\$1:9] MR4, type R0, address 0x2, reset 0x7287 OUI[\$1:9] MR5, type R0, address 0x2, reset 0x7287 OUI[\$1:9] MR5, type R0, address 0x2, reset 0x7287 OUI[\$1:9] MR5, type R0, address 0x16, reset 0x7287 MR6, type R0, address 0x16, reset 0x7287 MR5, type R0, address																
MRZ, type RQ, address 0x02, reset 0x00000 STATE MRZ, type RQ, address 0x01, reset 0x00140 STATE MRZ, type RQ, address 0x01, reset 0x00140 STATE MRZ, type RQ, address 0x11, reset 0x00140 STATE MRZ, type RQ, address 0x11, reset 0x0000 STATE MRZ, type RQ, address 0x11, reset 0x00100 STATE MRZ, type RQ, address 0x12, reset 0x0000 STATE MRZ, type RQ, address 0x12, reset 0x						10						-				
MR2, type RO, address 0x82, reset 0x309E CUI[21:8] MR3, type RO, address 0x83, reset 0x7237 MR4, type RW, address 0x84, reset 0x96E NF RF	unti, typ				10T H					MEPS	ANEGO	RFAULT	ANEGA	LINK	JAB	FXTD
MR3, type R0, address 0x03, reset 0x02000	MR2. tvn									1411 1 0	744200	TUTTOLI	71112071	Liitit	O/ LD	EXID
MRS, type RO, address 0x04, reset 0x0500	(2, (3)	,	,					OUII	21:61							
MR4, type R0, address 0x16, reset 0x0000 S[4:0] MR5, type R0, address 0x17, reset 0x0000 S[4:0]	MR3. tvp	e RO. addre	ss 0x03. re	set 0x7237												
MR4, type RW, address 0x04, reset 0x0000 MR5, type RO, address 0x05, reset 0x0000 MR6, type RO, address 0x05, reset 0x0000 MR6, type RO, address 0x06, reset 0x0000 MR6, type RO, address 0x16, reset 0x0000 MR6, type RO, address 0x11, reset 0x00000 MR17, type RW, address 0x11, reset 0x00000 MR18, type RO, address 0x11, reset 0x00000 MR18, type RO, address 0x12, reset 0x00000 MR18, type RO, address 0x12, reset 0x00000 MR18, type RO, address 0x12, reset 0x00000 MR18, type RW, address 0x13, reset 0x0000 MR19, type RW, address 0x13, reset 0x0000 MR23, type RW, address 0x13, reset 0x0000 MR24, type RW, address 0x13, reset 0x0000 MR25, type RW, address 0x13, reset 0x0000 MR26, type RW, address 0x13, reset 0x0000 MR27, type RW, address 0x13, reset 0x0000 MR28, type RW, address 0x13, reset 0x0000 MR29, type RW, address 0x13, reset 0x0000 MR20, type RW, address 0x13, reset 0x0000 MR21, type RW, address 0x13, reset 0x0000 MR21, type RW, address 0x13, reset 0x0000 0x000 MR22, type RW, address 0x13, reset 0x0000 0x000 MR23, type RW, address 0x13, reset 0x0000 0x000 MR24, type RW, address 0x13, reset 0x0000 0x000 MR25, type RW, address 0x13, reset 0x0000 0x000 MR26, type RW, address 0x13, reset 0x0000 0x000 MR27, type RW, address 0x13, reset 0x0000 0x000 MR28, type RW, address 0x13, reset 0x0000 0x000 MR27, type RW, address 0x13, reset 0x0000 0x000 MR28, type RW, address 0x13, reset 0x0000 0x000 MR27, type RW, address 0x13, reset 0x0000 0x000 MR28, type RW, address 0x13, reset 0x0000 0x000 MR29, type RW, address 0x13, reset 0x0000 0x000 MR29, type RW, address 0x13, reset 0x0000 0x000 MR29, type RW,									M	IN				R	N	
NP	MR4. tvp	e R/W. addr			1											
MR5, type RO, address 0x05, reset 0x0000 NP ACK RF A(70) S(4.0) NR16, type RO, address 0x06, reset 0x0000 NR16, type RO, address 0x10, reset 0x0100 NR17, type RW, address 0x11, reset 0x0100 NR17, type RW, address 0x11, reset 0x0000 NR17, type RW, address 0x11, reset 0x0000 NR17, type RW, address 0x12, reset 0x0000 NR19, type RW, address 0x13, reset 0x0000 0x000 NR19, type RW, address 0x13, reset 0x0000 0x000 NR24, type RW, address 0x13, reset 0x0000 0x000 NR19, type RW, address 0x13, reset 0x0000 0x000 NR		,,,,,,						A3	A2	A1	A0			S[4:0]		
NR5, type RN, address 0x10, reset 0x0000		e RO, addre		set 0x0000										-1 -1		
MR16, type RN, address 0x10, reset 0x001 0		_					Αſ	7:0]						S[4:0]		
MRT46, type RIW, address 0x10, reset 0x0140 MRT46, type RIW, address 0x10, reset 0x0140 MRT47, type RIW, address 0x11, reset 0x00000 MRT47, type RIW, address 0x11, reset 0x00000 MRT47, type RIW, address 0x11, reset 0x00000 MRT48, type RIW, address 0x12, reset 0x00000 MRT48, type RIW, address 0x12, reset 0x00000 MRT48, type RIW, address 0x13, reset 0x440000 MRT48, type RIW, address 0x13, reset 0x440000 MRT49, type RIW, address 0x13, reset 0x440000 MRT49, type RIW, address 0x13, reset 0x00000 MRT49, type RIW, address 0x18, reset 0x00000 MRT49, type RIW, address 0x18, reset 0x0000000 MRT49, type RIW, address 0x18, reset 0x00000000000000000000000000000000000	MR6, typ	e RO, addre	ss 0x06, re	set 0x0000												
RPTR NPOL	, ,,		,									PDF	LPNPA		PRX	LPANEGA
RPTR NPOL	MR16, ty	pe R/W, add	lress 0x10,	reset 0x014	10											
ARBER_IE RXER_IE PRX_IE PRX_IE PDF_IE LPACK_IE LSCHG_E RAULT_IE ARBODAES ARBER_NT RAED_NT RAED	-	-				NL10					APOL	RVSPOL			PCSBP	RXCC
MR19, type RW, address 0x12, reset 0x00000 ANEGF DPLX RATE RXSD RX_LOCK MR19, type RW, address 0x13, reset 0x40000 TXO(1-0) LED(13-0)	MR17, ty	pe R/W, add	lress 0x11,	reset 0x000	00							I				
MR19, type RW, address 0x12, reset 0x00000 ANEGF DPLX RATE RXSD RX_LOCK MR19, type RW, address 0x13, reset 0x40000 TXO(1-0) LED(13-0)		-				LSCHG_IE	RFAULT_IE	ANEGCOMP_E	JABBER_INT	RXER_INT	PRX_INT	PDF_INT	LPACK_INT	LSCHG_INT	RFAULT_INT	ANEGCOMP_NT
MR19, type R/W, address 0x13, reset 0x0000 TXC(1:0) MR23, type R/W, address 0x17, reset 0x0000 MR24, type R/W, address 0x17, reset 0x00000 MR25, type R/W, address 0x18, reset 0x00000 ACMIS, type R/W, address 0x18, reset 0x00000 ACMIS, type R/W (. offset 0x00, reset 0x0000, 0000 ACMIS, type R/W, offset 0x04, reset 0x0000, 0000 ACRIS, type R/W, offset 0x04, reset 0x0000, 0000 ACRIS, type R/W, offset 0x04, reset 0x0000, 0000 EN R/S (. type R/W, offset 0x04, reset 0x0000, 0000) EN R/S (. type R/W, offset 0x04, reset 0x0000, 0000) EN R/S (. type R/W, offset 0x10, reset 0x0000, 0000) ACREFCTL, type R/W, offset 0x10, reset 0x0000, 0000 EN R/S (. type R/W, offset 0x10, reset 0x0000, 0000) ACREFCTL, type R/W, offset 0x10, reset 0x0000, 0000	MR18, ty	pe RO, addr	ess 0x12, r	eset 0x000	0		1			1	1			1		
TXC(1-0) MR24, type R/W, address 0x17, reset 0x0000 MR24, type R/W, address 0x18, reset 0x0000 MR24, type R/W, address 0x18, reset 0x0000 PD_MODE AUTO_SW MDIX MDIX_CM MDIX_SD Analog Comparators Base 0x4003.C000 ACMIS, type R/W1C, offset 0x00, reset 0x0000.0000 ACMIS, type R/W1C, offset 0x04, reset 0x0000.0000 ACRIS, type R/W, offset 0x04, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x08, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x40, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 ACCTLO, type R/W, offset 0x60, reset 0x0000.0000 OVAL ACCTLO, type R/W, offset 0x60, reset 0x0000.0000				ANEGF	DPLX	RATE	RXSD	RX_LOCK								
MR23, type R/W, address 0x17, reset 0x0000 MR24, type R/W, address 0x18, reset 0x00C0 PD_MODE ALTO_SW MDIX MDIX_CM MDIX_SD Analog Comparators Base 0x4003.C000 ACMIS, type R/W, offset 0x00, reset 0x0000.0000 ACMIS, type R/W, offset 0x00, reset 0x0000.0000 ACRIS, type R/W, offset 0x08, reset 0x0000.0000 ACRIFECTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTATO, type RO, offset 0x20, reset 0x0000.0000 ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 ACCREATED, type RO, offset 0x40, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x40, reset 0x0000.0000	MR19, ty	pe R/W, add	lress 0x13,	reset 0x400	00											
MR24, type R/W, address 0x18, reset 0x000C0 MR24, type R/W, address 0x18, reset 0x000C0 Analog Comparators Base 0x4003 C0000 ACMIS, type R/WIC, offset 0x00, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type R/W, offset 0x08, reset 0x0000.0000 ACRIS, type R/W, offset 0x08, reset 0x0000.0000 EN RNG	TX	O[1:0]														
MR24, type R/W, address 0x18, reset 0x000C0 PD_MODE AUTO_SW MDIX MDIX_CM MDIX_SD	MR23, ty	pe R/W, add	lress 0x17,	reset 0x00°	10											
Analog Comparators Base 0x4003.C000 ACMIS, type RW1C, offset 0x00, reset 0x0000.0000 ACMIS, type RW, offset 0x04, reset 0x0000.0000 ACMIS, type RW, offset 0x04, reset 0x0000.0000 ACINTEN, type RW, offset 0x08, reset 0x0000.0000 EN RNG NEFF ACSTATO, type RW, offset 0x20, reset 0x0000.0000 EN RNG NEFF ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x24, reset 0x0000.0000										LED.	1[3:0]			LED	0[3:0]	
Analog Comparators Base 0x4003.C000 ACMIS, type RWIC, offset 0x00, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type RIW, offset 0x08, reset 0x0000.0000 ACREFCTL, type RIW, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTATO, type RO, offset 0x20, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000	MR24, ty	pe R/W, add	lress 0x18,	reset 0x000	C0											
ACRIS, type R/W1C, offset 0x00, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type RV, offset 0x08, reset 0x0000.0000 ACRIFECTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTATO, type RO, offset 0x20, reset 0x0000.0000 ACRIFECTL, type RV, offset 0x20, reset 0x0000.0000 ACSTATO, type RO, offset 0x40, reset 0x0000.0000 ACSTATO, type RO, offset 0x40, reset 0x0000.0000 ACSTATO, type RO, offset 0x40, reset 0x0000.0000 ACSTATO, type RV, offset 0x40, reset 0x0000.0000									PD_MODE	AUTO_SW	MDIX	MDIX_CM		MDIX	C_SD	
ACRIS, type R/W1C, offset 0x00, reset 0x0000.0000 ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACRIS, type R/W, offset 0x04, reset 0x0000.0000 ACRIS, type R/W, offset 0x08, reset 0x0000.0000 EN RNG VREF ACSTATO, type RO, offset 0x20, reset 0x0000.0000 ACSTAT1, type RO, offset 0x20, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000	Analog	Compai	rators													
ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACINTEN, type R/W, offset 0x08, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACRECTL, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000	Base 0x	4003.C000)													
ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACINTEN, type R/W, offset 0x08, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 ACREFCTL, type RO, offset 0x20, reset 0x0000.0000 EN RNG OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000	ACMIS, t	ype R/W1C,	offset 0x00	0, reset 0x0	000.0000											
ACRIS, type RO, offset 0x04, reset 0x0000.0000 ACINTEN, type R/W, offset 0x08, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 ACREFCTL, type RO, offset 0x20, reset 0x0000.0000 EN RNG OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000																
ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 ACREFCTL, type RO, offset 0x20, reset 0x0000.0000 EN RNG OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x24, reset 0x0000.0000														IN2	IN1	IN0
ACINTEN, type R/W, offset 0x08, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 COVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 COVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 COVAL ACCTL0, type RW, offset 0x24, reset 0x0000.0000	ACRIS, t	ype RO, offs	set 0x04, re	set 0x0000.	.0000											
ACINTEN, type R/W, offset 0x08, reset 0x0000.0000 ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 COVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 COVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 COVAL ACCTL0, type RW, offset 0x24, reset 0x0000.0000																
ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000														IN2	IN1	IN0
ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 COVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACINTEN	I, type R/W,	offset 0x08	3, reset 0x0	000.000				1							
ACREFCTL, type R/W, offset 0x10, reset 0x0000.0000 EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 COVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000																
EN RNG VREF ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000														IN2	IN1	IN0
ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACREFC	IL, type R/V	V, offset Ux	10, reset 0	(0000.0000)										
ACSTAT0, type RO, offset 0x20, reset 0x0000.0000 OVAL ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000							- FN	DNC						\/D		
ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACSTAT) type PO	offeet nyon	rosot Ovon	00 0000		EIN	KING						VK	.LF	
ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACCIAIL	, type KO, (J.1361 UXZU,	, . eset 0x00	00.0000											
ACSTAT1, type RO, offset 0x40, reset 0x0000.0000 OVAL ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000															OVAI	
ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACSTAT*	1. type RO	offset 0x40	reset 0x00	00.0000										OVAL	
ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000		, ., ,, , , , , , , , , , , , , , , , ,														
ACSTAT2, type RO, offset 0x60, reset 0x0000.0000 OVAL ACCTL0, type R/W, offset 0x24, reset 0x0000.0000															OVAL	
ACCTL0, type R/W, offset 0x24, reset 0x0000.0000	ACSTAT2	2, type RO. o	offset 0x60.	reset 0x00	00.000											
ACCTL0, type R/W, offset 0x24, reset 0x0000.0000																
															OVAL	
TOEN ASRCP TSLVAL TSEN ISLVAL ISEN CINV	ACCTL0,	type R/W, c	offset 0x24,	reset 0x00	00.0000											
TOEN ASRCP TSLVAL TSEN ISLVAL ISEN CINV																
					TOEN	ASF	RCP		TSLVAL	TS	EN	ISLVAL	IS	EN	CINV	

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
ACCTL1,	type R/W, o	offset 0x44,	reset 0x00	00.000											
				TOEN	ASF	RCP		TSLVAL	TS	EN	ISLVAL	ISI	EN	CINV	
ACCTL2,	type R/W, o	offset 0x64,	reset 0x00	00.0000											
				TOEN	ASF	RCP		TSLVAL	TS	EN	ISLVAL	ISI	EN	CINV	

C Ordering and Contact Information

C.1 Ordering Information

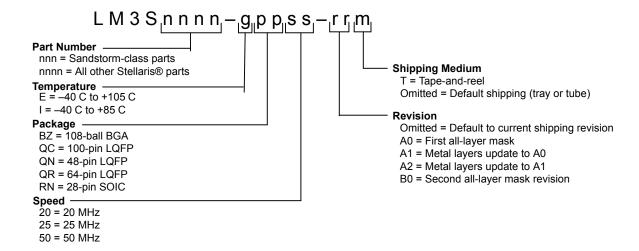


Table C-1. Part Ordering Information

Orderable Part Number	Description
LM3S8933-IBZ50	Stellaris® LM3S8933 Microcontroller
LM3S8933-IBZ50 (T)	Stellaris® LM3S8933 Microcontroller
LM3S8933-EQC50	Stellaris® LM3S8933 Microcontroller
LM3S8933-EQC50 (T)	Stellaris [®] LM3S8933 Microcontroller
LM3S8933-IQC50	Stellaris® LM3S8933 Microcontroller
LM3S8933-IQC50 (T)	Stellaris [®] LM3S8933 Microcontroller

C.2 Kits

The Luminary Micro 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:
 - http://www.luminarymicro.com/products/reference design kits/
- Evaluation Kits provide a low-cost and effective means of evaluating Stellaris[®] microcontrollers before purchase:
 - http://www.luminarymicro.com/products/kits.html
- Development Kits provide you with all the tools you need to develop and prototype embedded applications right out of the box:
 - http://www.luminarymicro.com/products/development_kits.html

See the Luminary Micro website for the latest tools available, or ask your Luminary Micro distributor.

C.3 Company Information

Luminary Micro, Inc. designs, markets, and sells ARM Cortex-M3-based microcontrollers (MCUs). Austin, Texas-based Luminary Micro is the lead partner for the Cortex-M3 processor, delivering the world's first silicon implementation of the Cortex-M3 processor. Luminary Micro's introduction of the Stellaris® family of products provides 32-bit performance for the same price as current 8- and 16-bit microcontroller designs. With entry-level pricing at \$1.00 for an ARM technology-based MCU, Luminary Micro's Stellaris product line allows for standardization that eliminates future architectural upgrades or software tool changes.

Luminary Micro, Inc. 108 Wild Basin, Suite 350 Austin, TX 78746 Main: +1-512-279-8800 Fax: +1-512-279-8879 http://www.luminarymicro.com sales@luminarymicro.com

C.4 Support Information

For support on Luminary Micro products, contact: support@luminarymicro.com +1-512-279-8800, ext. 3