DATASHEET

Apollo2 SoC

Includes Apollo2 Thin SoC

A-SOCAP2-DSGA01EN v1.3

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. “TYPICAL” PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING “TYPICALS” MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER’S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Revision History

Revision	Date	Description
0.67	May 2016	Updates from design team. General cleanup. Fix to Flash Control register tagging Electrical Spec update
0.71	August 2016	Refreshed register source file contents. Updated electrical spec section based on latest review feedback. Few updates in reset section description
0.8	December 2016	Pre-Production Release Several bug fixes
0.90	March 2017	Cleanup, corrections and additions for initial public release
0.91	March 2017	Updated temp range, I^2S Slave Interface section
0.92	May 2017	Initial open (non-NDA) release
0.93	June 2017	Corrected Pin List and Function Table Corrected Interrupt Vector Table Corrected Alarm RPT Function table in CLKGEN Updated Deep-Sleep to Buck Run mode transition time
0.94	September 2017	Electrical: Many TBDs filled in Pin List and Functions Table: Noted analog inputs IOS: I^2C/SPI Slave Module LRAM Addressing figure corrected and text in section 5.3 added for clarity. GPIO: Corrected 4-wire SPI loopback mode table STTIMER: Clarified STTMR register operation. CLKGEN: Made CCTRL reg visible and added instructions for initializing. Package: PCB land pattern and solder stencil added for CSP package.
0.95	December 2017	GPIO: Clarification of pad drive strengths CTIMER: Clarification of HCLK_DIV4 as a timer clock source Electrical: Max junction temperature extended; max PDM output clock frequency updated
1.0	February 2018	Electrical: ▪ Added SPI master and slave timings ▪ Added I^2C timings for Standard Mode and Fast Mode Plus ▪ Updated SPI Slave FSCLK max frequency ▪ Added timings for ADC sampling dynamics (trigger to start of scan) ▪ Updated ADC conversion rates ▪ Updated CTIMER external clock limits
1.0.1	March 2018	Updated IOS specification T_{CE_SDZ} and T_{SU_SI}

Revision	Date	Description
1.1	June 2019	System Core (flash): Updated note about allowable number of program cycles between erase cycles. Electrical: ▪ Updated VESDCDM - ESD Charged Device Model (CDM) ▪ Updated VESDHBM - ESD Human Body Model (HBM) ▪ Noted wafer package exceptions ▪ Added test conditions for temp sensor accuracy spec ▪ Added test conditions in General Purpose Input/Output (GPIO) table ▪ GPIO VIH and VIL corrected Package: Note about light susceptibility with CSP package Appendix 1 added
1.2	September 2020	General: Apollo2 Thin package added ▪ GPIO: Corrected PADnINPEN settings for I^2S Connections ▪ VREG: Clarified need for VDDC and VDDF capacitors for internal LDOs. ▪ Package Mechanical: Dimension Drawings added for Apollo2 Thin SoC ▪ Ordering Information: Apollo2 Thin SoC added
1.3	March 13, 2023	Updated document template

Reference Documents

Document ID	Description
A-SOCAP2-PBGA01EN	Apollo2 SoC Product Brief
A-SOCA2T-PBGA01EN	Apollo2 Thin SoC Product Brief

A-SOCAP2-DSGA01EN v1.3Table of Contents

List of Tables

List of Figures

SECTION

1

Introduction

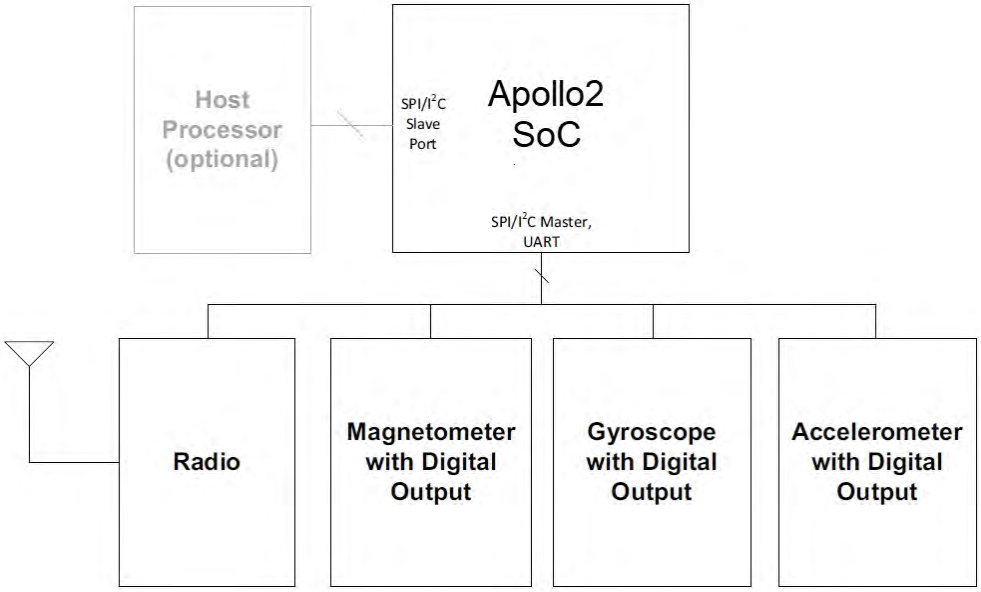
1.1 Overview

The Apollo SoC family is an ultra-low power, highly integrated microcontroller designed for battery-powered and portable, mobile devices. At the heart of the Apollo2 SoC is Ambiq's patented Subthreshold Power Optimized Technology (SPOT®) and a powerful Arm® Cortex®-M4 processor with Floating Point Unit. This combination dramatically reduces energy consumption while still enabling abundant application processing power to add greater capability and extended life to battery-operated devices.

With unprecedented levels of energy efficiency, the Apollo2 can deliver always-on keyword detection and voice assistant integration, local voice control, complex sensor processing, gesture recognition and activity monitoring applications within the smallest power budget.

The Apollo2 SoC is the 2nd generation controller building upon Ambiq's Apollo SoC product family. The Apollo2 SoC integrates up to 1 MB of flash memory and 256 KB of RAM to accommodate radio and sensor overhead while still leaving plenty of space for application code and algorithms. In addition to flexible serial channels for radio and sensor communication, this microcontroller also includes dual PDM inputs for applications that require digital microphones for near and mid-field always-on keyword detection, voice assistant integration, and voice control.

Figure 1-1: Typical Sensor Application Circuit for the Apollo2 SoC



1.2 Features

- Ultra-low supply current:
  - < 10 µA/MHz executing from FLASH or RAM at 3.3 V
  - < 3 µA deep sleep mode with RTC at 3.3 V
- High-performance Arm Cortex-M4 Processor
  - Up to 48 MHz clock frequency
  - Floating point unit
  - Memory protection unit
  - Wake-up interrupt controller with 32 interrupts
- Ultra-low power memory:
  - Up to 1 MB of flash memory for code/data
  - Up to 256 KB of low leakage RAM for code/data
  - 16 kB 2-way Associative Cache
- Ultra-low power interface for on- and off-chip sensors:
  - 14 bit ADC at up to 1.2 MS/s, 15 selectable input channels available
  - Voltage Comparator
  - Temperature sensor with ±3ºC accuracy after calibration
- Flexible serial peripherals:
  - 6x I^2C/SPI masters with 128-byte bidirectional FIFO for communication with sensors, radios, and other peripherals
  - 1x I^2C/SPI slave for host communications with 256-byte LRAM area for FIFO/host support
  - 2x UART modules with 32-location Tx and Rx FIFOs
  - PDM for mono and stereo audio microphone
  - 1x I^2S slave for PDM audio pass-through

- Rich set of clock sources:
  - 32.768 kHz XTAL oscillator
  - Low frequency RC oscillator – 1.024 kHz
  - High frequency RC oscillator – 48 MHz
  - RTC based on Ambiq's AM08X5/18X5 families
- Wide operating range: 1.755-3.63 V, –40 to 85°C
- Compact package options:
  - 4.5 x 4.5 mm (<0.59mm thk pkg) 64-pin BGA with 50 GPIO
  - 2.5 x 2.5 mm (<0.33mm thk pkg) 49-pin WLCSP with 34 GPIO
  - 2.5 x 2.5 mm (<0.30mm thk pkg) 49-pin WLCSP with 34 GPIO (Apollo2 Thin SoC)

1.3 Applications

- Headsets, Earphones, Wireless Earbuds
- Mobile Accessories
- Hearing Aids
- Always-Listening, Voice-Activated Appliances
- Portable Voice Assistants
- Remote Controls
- Smartcards
- Fingerprint Readers
- Wearable electronics including smart watches
- Smart meters
- Wireless sensors
- Activity and fitness monitors
- Consumer electronics

SECTION

2

Apollo2 SoC Package Pins

2.1 Pin Configuration

Figure 2-1: CSP Pin Configuration Diagram (Top View - Balls on Bottom)

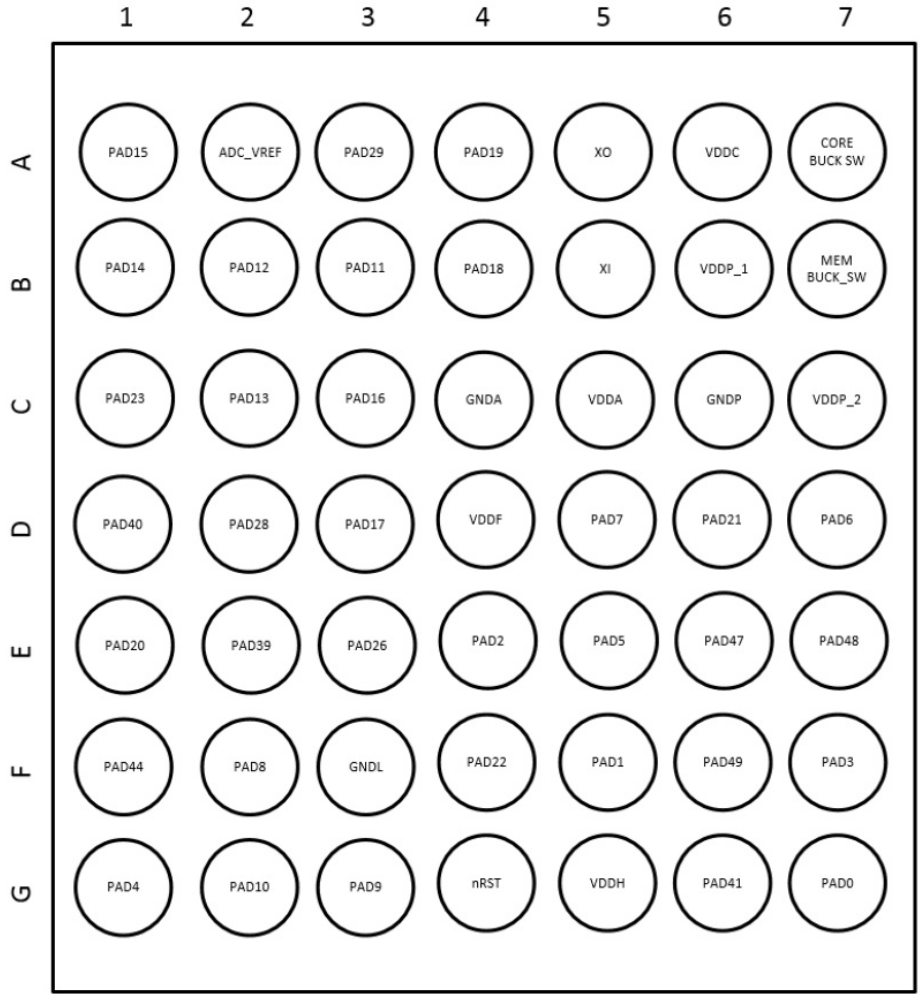
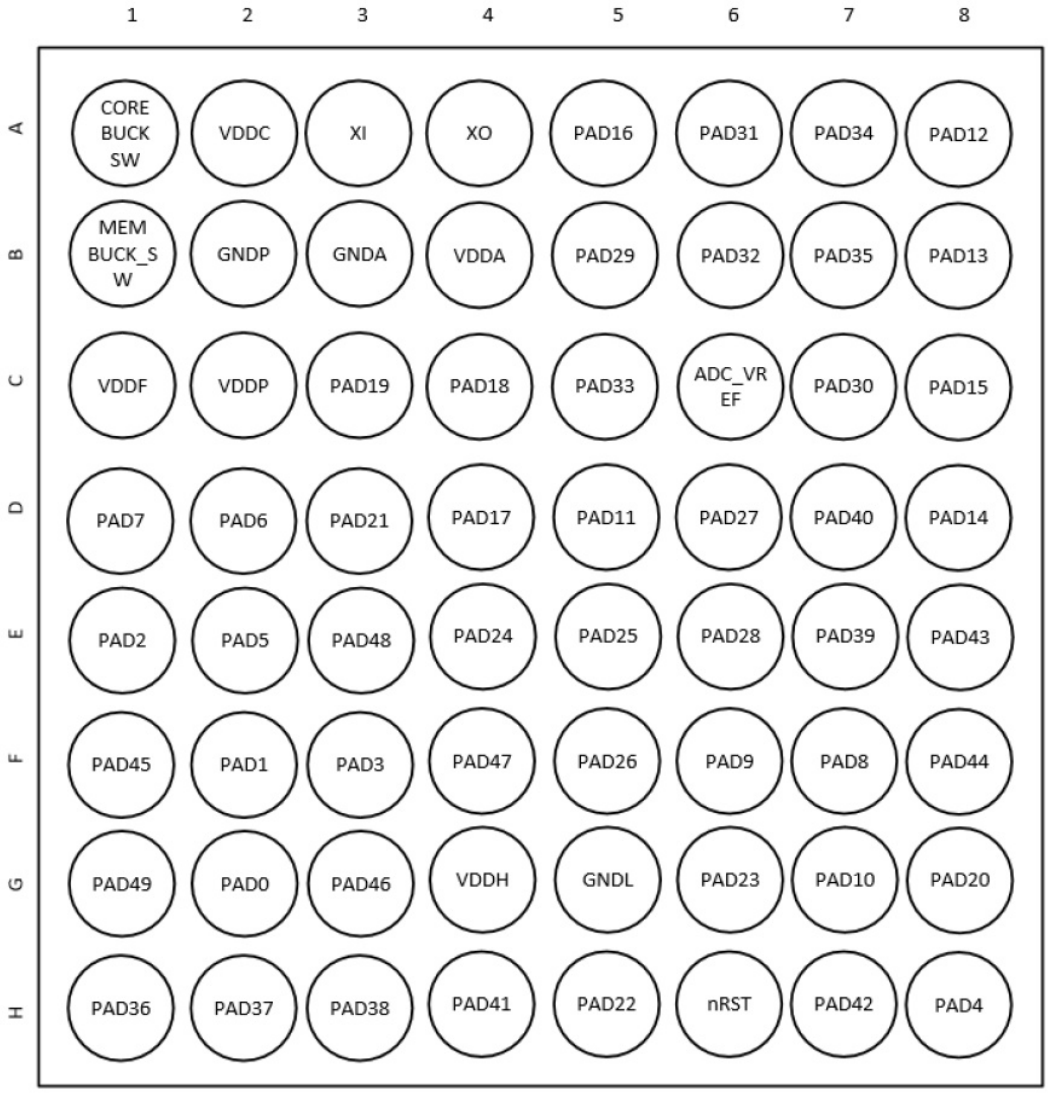


Figure 2-2: BGA Pin Configuration Diagram (Top View - Balls on Bottom)



2.2 Pin Connections

The following table lists the external pins of the Apollo2 SoC and their available functions. Highlighted in red background are function selection changes from silicon revision A to B, and changes to CSP package pinout.

Table 2-1: Pin List and Function Table - Power

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C2	B6,C7	-	-	VDDP	VDD Supply for Buck converters	Power
B2	C6	-	-	GNDP	Ground Connection for Buck converters	Ground
B4	C5	-	-	VDDA	Analog Voltage Supply	Power
B3	C4	-	-	GNDA	Ground for Analog Supply	Ground
G5	F3	-	-	GNDL	Ground for I/O Pads	Ground
G4	G5	-	-	VDDH	VDD Supply for I/O pads	Power
C6	A2	-	-	ADC_VREF	Dedicated ADC voltage reference	Power

Table 2-2: Pin List and Function Table - BUCK

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C1	D4	-	-	VDDF	Memory Buck Converter Voltage Output Supply	Power
A1	A7	-	-	COREBUCK_SW	Core Buck Converter Inductor Switch	Power
B1	B7	-	-	MEM_BUCK_SW	Memory Buck Converter Inductor Switch	Power
A2	A6	-	-	VDDC	Core Buck Converter Voltage Output Supply	Power

Table 2-3: Pin List and Function Table - OSCILLATOR

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A4	A5	-	-	XO	32.768 kHz Crystal Output	XT
A3	B5	-	-	XI	32.768 kHz Crystal Input	XT

Table 2-4: Pin List and Function Table - RESET

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
H6	G4	-	-	nRST	External Reset Input	Input/ Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G2	G7	0	0	SLSCL	I^2C Slave Clock	Input
			1	SLSCK	SPI Slave Clock	Input
			2	CLKOUT	Oscillator output clock	Output
			3	GPIO0	General Purpose I/O	Input/ Output
			4	MxSCKLB	SPI MX CLKIOUT-> CLKIN Loopback	Output
			5	M2SCK	SPI Master 2 Clock	Output
			6	MxSCLLB	I^2C MX CLKIOUT- >CLKIN Loopback	Input
			7	M2SCL	I^2C Master 2 Clock	Open Drain Output
F2	F5	1	0	SLSDA	I^2C Slave I/O Data	Bidirectional Open Drain
			1	SLMISO	SPI Slave Output Data	Output
			2	UART0TX	UART0 Transmit	Output
			3	GPIO1	General Purpose I/O	Input/ Output
			4	MxMISOLB	SPI M0 Data Loopback (int to IOM)	Input
			5	M2MISO	SPI Master 2 Input Data	Input
			6	MxSDALB	Loopback I^2C Master 0 I/O Data	Bidirectional Open Drain
			7	M2SDA	I^2C Master 2 I/O Data Bidirectional	Open Drain

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E1	E4	2	0	SLWIR3	SPI Slave I/O Pin for 3-Wire Mode	Bidirectional 3-state
			1	SLMOSI	SPI Slave Input Data	Input
			2	UART0RX	UART0 Receive	Input
			3	GPIO2	General Purpose I/O	Input/ Output
			4	MxMOSILB	SPI Master 0 Output Data (loopback)	Input
			5	M2MOSI	SPI Master 2 Output Data	Output
			6	MxWIR3LB	SPI Master 0 I/O Pin for 3-Wire Mode (loopback)	Bidirectional 3-state
			7	M2WIR3	SPI Master 1 I/O Pin for 3-Wire Mode (loopback)	Bidirectional 3-state
F3	F7	3	0	UA0RTS	UART0 Request to Send (RTS)	Output
			1	SLnCE	SPI Slave Chip Enable	Input
			2	M1nCE4	SPI Master 1 Chip Enable 4	Output
			3	GPIO3	General Purpose I/O	Input/ Output
			4	MxnCELB	SPI Master 0 Chip Enable (loopback)	Input
			5	M2nCE0	SPI Master 2 Chip Enable 0	Output
			6	TRIG1	ADC Trigger input	Input
			7	I2S_WCLK	I^2S L/R Clock	Input
H8	G1	4	0	UA0CTS	UART0 Clear to Send (CTS)	Input
			1	SLINT	Configurable Slave Interrupt	Output
			2	M0nCE5	SPI Master 0 Chip Enable 5	Output
			3	GPIO4	General Purpose I/O Input/	Output
			4	SLINTGP	Loopback Slave Interrupt	Input
			5	M2nCE5	SPI Master 2 Chip Enable 5	Output
			6	CLKOUT	Oscillator Output Clock	Output
			7	32KHz_XT	32 kHz Clock	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E2	E5	5	0	M0SCL	I^2C Master 0 Clock	Open Drain Output
			1	M0SCK	SPI Master 0 Clock	Output
			2	UA0RTS	UART0 Request To Send (RTS)	Output
			3	GPIO5	General Purpose I/O	Input/ Output
			4	M0SCKLB	SPI Master 0 Clock (loopback)	Output
			5	RSVD	Reserved for future use	
			6	M0SCLLB	I^2C Master 0 Clock (loopback)	Open Drain Output
			7	M1nCE2	SPI Master 1 Chip Enable 2	Output
D2	D7	6	0	M0SDA	I^2C Master 0 Data	Bidirectional Open Drain
			1	M0MISO	SPI Master 0 Input Data	Input
			2	UA0CTS	UART0 Clear To Send (CTS)	Input
			3	GPIO6	General Purpose I/O	Input/ Output
			4	SLMISOLB	Loopback SPI Slave Output Data (loopback)	Output
			5	M1nCE0	SPI Master 1 Chip Enable 0	Output
			6	SLSDALB	Loopback I^2C Slave I/O Data (loopback)	Bidirectional Open Drain
			7	I2S_DAT	I^2S Data output	Output
D1	D5	7	0	M0WIR3	SPI Master 0 I/O Pin for 3-Wire Mode	Bidirectional 3-state
			1	M0MOSI	SPI Master 0 Output Data	Output
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO7	General Purpose I/O	Input/ Output
			4	TRIG0	ADC Trigger Input	Input
			5	UART0TX	UART0 Transmit	Output
			6	SLWIR3LB	Loopback SPI Slave I/O Pin for 3-Wire Mode	Bidirectional 3-state
			7	M1nCE1	SPI Master 1 Chip Enable 1	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F7	F2	8	0	M1SCL	I^2C Master 1 Clock Open Drain	Output
			1	M1SCK	SPI Master 1 Clock	Output
			2	M0nCE4	SPI Master 0 Chip Enable 4	Output
			3	GPIO8	General Purpose I/O	Input/ Output
			4	M2nCE4	SPI Master 2 Chip Enable 4	Output
			5	M1SCKLB	SPI Master 1 Clock (loopback)	Output
			6	UART1TX	UART1 Transmit	Output
			7	M1SCLLB	I^2C Master 1 Clock (loopback)	Open Drain Output
F6	G3	9	0	M1SDA	I^2C Master 1 Data	Bidirectional Open Drain
			1	M1MISO	SPI Master 1 Input Data	Input
			2	M0nCE5	SPI Master 0 Chip Enable 5	Output
			3	GPIO9	General Purpose I/O	Input/ Output
			4	M4nCE5	SPI Master 4 Chip Enable 5	Output
			5	SLMISOLB	SPI Slave Output Data (loopback)	Output
			6	UART1RX	UART1 Receive	Input
			7	SLSDALB	I^2C Slave Data (loopback)	Bidirectional Open Drain
G7	G2	10	0	M1WIR3	SPI Master 1 I/O Pin for 3-Wire Mode	Bidirectional 3-state
			1	M1MOSI	SPI Master 1 Output Data	Output
			2	M0nCE6	SPI Master 0 Chip Enable 6	Output
			3	GPIO10	General Purpose I/O	Input/ Output
			4	M2nCE6	SPI Master 2 Chip Enable 6	Output
			5	UA1RTS	UART1 Request to Send (RTS)	Output
			6	M4nCE4	SPI Master 4 Chip Enable 4	Output
			7	SLWIR3LB	Loopback SPI Slave I/O Pin for 3-Wire Mode (loopback)	Bidirectional 3-state

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
D5	B3	11	0	ADCSE2	Analog to Digital Converter SE In2	Analog Input
			1	M0nCE0	SPI Master 0 Chip Enable 0	Output
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO11	General Purpose I/O	Input/ Output
			4	M2nCE7	SPI Master 2 Chip Enable 7	Output
			5	UA1CTS	UART1 Clear to Send (CTS)	Input
			6	UART0RX	UART0 receive	Input
			7	PDM_DATA	PDM audio data input	Input
A8	B2	12	0	ADCD0NSE9	Analog to Digital Converter Diff In0 Neg	Analog Input
			1	M1nCE0	SPI Master 1 Chip Enable 0	Output
			2	TCTA0	Timer/Counter A0	Output
			3	GPIO12	General Purpose I/O	Input/ Output
			4	CLKOUT	Oscillator output clock	Output
			5	PDM_CLK	PDM Clock output	Output
			6	UA0CTS	UART0 Clear to Send (CTS)	Input
			7	UART1TX	UART1 Transmit	Output
B8	C2	13	0	ADCD0PSE8	Analog to Digital Converter Diff In0 Pos	Analog Input
			1	M1nCE1	SPI Master 1 Chip Enable 1	Output
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO13	General Purpose I/O	Input/ Output
			4	M2nCE3	SPI Master 2 Chip Enable 3	Output
			5	RSVD	Reserved for future use	
			6	UA0RTS	UART0 RTS (Request to Send)	Output
			7	UART1RX	UART1 Receive	Input

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
D8	B1	14	0	ADCD1P	Analog to Digital Converter Diff In1 Pos	Analog Input
			1	M1nCE2	SPI Master 1 Chip Enable 2	Output
			2	UART1TX	UART1 Transmit	Output
			3	GPIO14	General Purpose I/O	Input/ Output
			4	M2nCE1	SPI Master 1 Chip Enable 1	Output
			5	RSVD	Reserved for future use	
			6	SWDCK	Software debug clock	Input
			7	32KHz_XT	32 kHz clock	Output
C8	A1	15	0	ADCD1N	Analog to Digital Converter Diff In1 Neg Analog	Input
			1	M1nCE3	SPI Master 1 Chip Enable 3	Output
			2	UART1RX	UART1 Receive	Input
			3	GPIO15	General Purpose I/O	Input/ Output
			4	M2nCE2	SPI Master 2 Chip Enable 2	Output
			5	RSVD	Reserved	
			6	SWDIO	Serial Wire Data I/O	Bidirectional 3-state
			7	SWO	Serial Wire Debug	Output
A5	C3	16	0	ADCSE0	Analog to Digital Converter SE In0	Input
			1	M0nCE4	SPI Master 0 Chip Enable 4	Output
			2	TRIG0	ADC Trigger Input 0	Input
			3	GPIO16	General Purpose I/O	Input/ Output
			4	M2nCE3	SPI Master 2 Chip Enable 3	Output
			5	CMPIN0	Voltage comparator input 0	Analog Input
			6	UART0TX	UART0 transmit	Output
			7	UA1RTS	UART1 Request to Send (RTS)	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
D4	D3	17	0	CMPRF1	Voltage Comparator Reference 1	Analog Input
			1	M0nCE1	SPI Master 0 Chip Enable 1	Output
			2	TRIG1	ADC Trigger Input 1	Input
			3	GPIO17	General Purpose I/O	Input/ Output
			4	M4nCE3	SPI Master 4 Chip Enable 3	Output
			5	RSVD	Reserved for future use	
			6	UART0RX	UART0 Receive	Input
			7	UA1CTS	UART1 Clear to Send (CTS)	Input
C4	B4	18	0	CMPIN1	Voltage Comparator Input 1	Analog Input
			1	M0nCE2	SPI Master 0 Chip Enable 2	Output
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO18	General Purpose I/O	Input/ Output
			4	M4nCE1	SPI Master 4 Chip Enable 1	Output
			5	RSVD	Reserved for future use	
			6	UART1TX	UART1 Transmit	Output
			7	32KHz_XT	32kHz clock	Output
C3	A4	19	0	CMPRF0	Comparator Reference 0	Analog Input
			1	M0nCE3	SPI Master 0 Chip Enable 3	Output
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO19	General Purpose I/O	Input/ Output
			4	TCTA1	Timer/counter A1	Output
			5	RSVD	Reserved for future use	
			6	UART1RX	UART1 receive	Input
			7	I2S_BCLK	I^2S Bit Clock	Input

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G8	E1	20	0	SWDCK	Software Debug Clock	Input
			1	M1nCE5	SPI Master 1 Chip Enable 5	Output
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO20	General Purpose I/O	Input/ Output
			4	UART0TX	UART0 Transmit	Output
			5	UART1TX	UART1 Transmit	Output
			6	RSVD	Reserved for future use	
			7	RSVD	Reserved for future use	
D3	D6	21	0	SWDIO	Software Data I/O	Bidirectional 3-state
			1	M1nCE6	SPI Master 1 Chip Enable 6	Output
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO21	General Purpose I/O	Input/ Output
			4	UART0RX	UART0 Receive	Input
			5	UART1RX	UART1 Receive	Input
			6	RSVD	Reserved for future use	
			7	RSVD	Reserved for future use	
H5	F4	22	0	UART0TX	UART0 Transmit	Output
			1	M1nCE7	SPI Master 1 Chip Enable 7	Output
			2	TCTA3	Timer/Counter A3	Output
			3	GPIO22	General Purpose I/O	Input/ Output
			4	PDM_CLK	PDM Clock Output	Output
			5	RSVD	Reserved for future use	
			6	TCTB1	Timer/counter B1	Output
			7	SWO	Serial Wire Debug	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G6	C1	23	0	UART0RX	UART0 Receive	Input
			1	M0nCE0	SPI Master 0 Chip Enable 0	Output
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO23	General Purpose I/O	Input/ Output
			4	PDM_DATA	PDM audio data input	Input
			5	CMPOUT	Voltage comparator output	Output
			6	TCTB1	Timer/counter B1	Output
			7	RSVD	Reserved for future use	
E4	-	24	0	M2nCE1	SPI Master 2 Chip Enable 1	Output
			1	M0nCE1	SPI Master 0 Chip Enable 1	Output
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO24	General Purpose I/O	Input/ Output
			4	M5nCE0	SPI Master 5 Chip Enable 0	Output
			5	TCTA1	Timer/counter A1	Output
			6	I2S_BCLK	I^2S Bit Clock	Input
			7	SWO	Serial Wire Debug	Output
E5	-	25	0	RSVD	Reserved for future use	
			1	M0nCE2	SPI Master 0 Chip Enable 2	Output
			2	TCTA0	Timer/Counter A0	Output
			3	GPIO25	General Purpose I/O	Input/ Output
			4	M2SDA	I^2C Master 2 Data	Bidirectional Open Drain
			5	M2MISO	SPI Master 2 Input Data	Input
			6	SLMISOLB	SPI Slave Output Data (loopback)	Output
			7	SLSDALB	I^2C Slave I/O Data	Bidirectional Open Drain

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F5	E3	26	0	RSVD	Reserved	
			1	M0nCE3	SPI Master 0 Chip Enable 3	Output
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO26	General Purpose I/O	Input/ Output
			4	M2nCE0	SPI Master 2 Chip Enable 0	Output
			5	TCTA1	Timer/counter A1	Output
			6	M5nCE1	SPI Master 5 Chip Enable 1	Output
			7	M3nCE0	SPI Master 3 Chip Enable 0	Output
D6	-	27	0	RSVD	Reserved for future use	
			1	M1nCE4	SPI Master 1 Chip Enable 4	Output
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO27	General Purpose I/O	Input/ Output
			4	M2SCL	I^2C Master 2 Clock Open Drain	Output
			5	M2SCK	SPI Master 2 Clock	Output
			6	M2SCKLB	SPI M2 CLKIOUT->CLKIN (loopback)	Input
			7	M2SCLLB	I^2C M2 CLKIOUT->CLKIN (loopback)	Input
E6	D2	28	0	I2S_WCLK	I^2S L/R Clock	Input
			1	M1nCE5	SPI Master 1 Chip Enable 5	Output
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO28	General Purpose I/O	Input/ Output
			4	M2WIR3	SPI Master 1 I/O pin for 3-wire mode	Bidirectional 3-state
			5	M2MOSI	SPI Master 2 Output Data	Output
			6	M5nCE3	SPI Master 5 Chip Enable 3	Output
			7	SLWIR3LB	SPI Slave I/O pin for 3-wire	Bidirectional 3-state

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B5	A3	29	0	ADCSE1	Analog to Digital Converter SE In1	Analog Input
			1	M1nCE6	SPI Master 1 Chip Enable 6	Output
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO29	General Purpose I/O	Input/ Output
			4	UA0CTS	UART0 Clear to Send (CTS)	Input
			5	UA1CTS	UART1 Clear to Send (CTS)	Input
			6	M4nCE0	SPI Master 4 Chip Enable 0	Output
			7	PDM_DATA	PDM audio data input to chip	Input
C7	-	30	0	RSVD	Reserved for future use	
			1	M1nCE7	SPI Master 1 Chip Enable 7	Output
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO30	General Purpose I/O	Input/ Output
			4	UART0TX	UART0 Transmit	Output
			5	UA1RTS	UART1 Request to Send (RTS)	Output
			6	RSVD	Reserved for future use	
			7	I2S_DAT	I^2S Data output	Output
A6	-	31	0	ADCSE3	Analog to Digital Converter SE In3	Analog Input
			1	M0nCE4	SPI Master 0 Chip Enable 4	Output
			2	TCTA3	Timer/Counter A3	Output
			3	GPIO31	General Purpose I/O	Input/ Output
			4	UART0RX	UART0 receive	Input
			5	TCTB1	Timer/counter B1	Output
			6	RSVD	Reserved for future use	
			7	RSVD	Reserved for future use	

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B6	-	32	0	ADCSE4	Analog to Digital Converter SE Input 4	Analog Input
			1	M0nCE5	SPI Master 0 Chip Enable 5	Output
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO32	General Purpose I/O	Input/ Output
			4	RSVD	Reserved for future use	
			5	TCTB1	Timer/counter B1 Output	
			6	RSVD	Reserved for future use	
			7	RSVD	Reserved for future use	
C5	-	33	0	ADCSE5	Analog to Digital Converter SE In5	Analog Input
			1	M0nCE6	SPI Master 0 Chip Enable 6	Output
			2	32KHz_XT	32kHz clock	Output
			3	GPIO33	General Purpose I/O	Input/ Output
			4	RSVD	Reserved for future use	
			5	M3nCE7	SPI Master 3 Chip Enable 7	Output
			6	TCTB1	Timer/counter B1	Output
			7	SWO	Serial Wire Debug	Output
A7	-	34	0	ADCSE6	Analog to Digital Converter SE In6	Analog Input
			1	M0nCE7	SPI Master 0 Chip Enable 7	Output
			2	M2nCE3	SPI Master 2 Chip Enable 3	Output
			3	GPIO34	General Purpose I/O	Input/ Output
			4	CMPRF2	Comparator reference 2	Analog Input
			5	M3nCE1	SPI Master 3 Chip Enable 1	Output
			6	M4nCE0	SPI Master 4 Chip Enable 0	Output
			7	M5nCE2	SPI Master 5 Chip Enable 2	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B7	-	35	0	ADCSE7	Analog to Digital Converter SE In7	Analog Input
			1	M1nCE0	SPI Master 1 Chip Enable 0	Output
			2	UART1TX	UART1 Transmit	Output
			3	GPIO35	General Purpose I/O	Input/ Output
			4	M4nCE6	SPI Master 4 Chip Enable 6	Output
			5	TCTA1	Timer/counter A1	Output
			6	UA0RTS	UART0 Request to Send (RTS)	Output
			7	M3nCE2	SPI Master 3 Chip Enable 2	Output
H1	-	36	0	TRIG1	ADC Trigger input	Input
			1	M1nCE1	SPI Master 1 Chip Enable 1	Output
			2	UART1RX	UART1 Receive	Input
			3	GPIO36	General Purpose I/O	Input/ Output
			4	32KHz_XT	32 kHz Clock	Output
			5	M2nCE0	SPI Master 2 Chip Enable 0	Output
			6	UA0CTS	UART0 Clear to Send (CTS)	Input
			7	M3nCE3	SPI Master 3 Chip Enable 3	Output
H2	-	37	0	TRIG2	ADC Trigger Input 2	Input
			1	M1nCE2	SPI Master 1 Chip Enable 2	Output
			2	UA0RTS	UART0 Request To Send (RTS)	Output
			3	GPIO37	General Purpose I/O	Input/ Output
			4	M3nCE4	SPI Master 3 Chip Enable 4	Output
			5	M4nCE1	SPI Master 4 Chip Enable 1	Output
			6	PDM_CLK	PDM Clock Output	Output
			7	TCTA1	Timer/counter A1	Output

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
H3	-	38	0	TRIG3	ADC Trigger Input 3	Input
			1	M1nCE3	SPI Master 1 Chip Enable 3	Output
			2	UA0CTS	UART0 Clear To Send (CTS) Input	Input
			3	GPIO38	General Purpose I/O	Input/ Output
			4	M3WIR3	SPI Master 3 I/O pin for 3-wire mode	Bidirectional 3-state
			5	M3MOSI	SPI Master 3 Output Data	Output
			6	M4nCE7	SPI Master 4 Chip Enable 7	Output
			7	SLWIR3LB	SPI Slave I/O pin for 3-wire	Bidirectional 3-state
E7	E2	39	0	UART0TX	UART0 transmit	Output
			1	UART1TX	UART1 Transmit	Output
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO39	General Purpose I/O	Input/ Output
			4	M4SCL	I^2C Master 4 Clock	Output
			5	M4SCK	SPI Master 4 Clock	Output
			6	M4SCKLB	SPI M4 CLKIOUT->CLKIN (loopback)	Input
			7	M4SCLLB	I^2C M4 CLKIOUT->CLKIN (loopback)	Input
D7	D1	40	0	UART0RX	UART0 receive	Input
			1	UART1RX	UART1 Receive	Input
			2	TRIG0	ADC trigger	Input
			3	GPIO40	General Purpose I/O	Input/ Output
			4	M4SDA	I^2C Master 4 Data	Bidirectional Open Drain
			5	M4MISO	SPI Master 4 data in	Input
			6	SLMISOLB	SPI slave output data (loopback)	Output
			7	SLSDALB	I^2C Slave I/O data	Bidirectional Open Drain

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
H4	G6	41	0	M2nCE1	SPI Master 2 chip enable 1	Output
			1	CLKOUT	Oscillator Output Clock	Output
			2	SWO	Serial Wire Debug Output	Output
			3	GPIO41	General Purpose I/O	Input/ Output
			4	M3nCE5	IO Master 3 Chip Enable	Output
			5	M5nCE7	IO Master 5 Chip Enable	Output
			6	M4nCE2	IO Master 4 Chip Enable	Output
			7	UA0RTS	UART0 Request to Send (RTS)	Output
H7	-	42	0	M2nCE2	SPI Master 2 Chip Enable 2	Output
			1	M0nCE0	SPI Master 0 Chip Enable 0	Output
			2	TCTA0	Timer/Counter A0	Output
			3	GPIO42	General Purpose I/O	Input/ Output
			4	M3SCL	I^2C Master 3 Clock Open Drain	Output
			5	M3SCK	SPI Master 3 Clock	Output
			6	M3SCKLB	SPI M3 CLKIOUT->CLKIN (loopback)	Input
			7	M3SCLLB	I^2C M3 CLKIOUT->CLKIN (loopback)	Input
E8	-	43	0	M2nCE4	SPI Master 1 Chip Enable 4	Output
			1	M0nCE1	SPI Master 0 Chip Enable 1	Output
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO43	General Purpose I/O	Input/ Output
			4	M3SDA	I^2C Master 3 Data	Bidirectional Open Drain
			5	M3MISO	SPI Master 3 Input data	Input
			6	SLMISOLB	SPI Slave output data (loopback)	Output
			7	SLSDALB	I^2C Slave I/O data (loopback)	Bidirectional Open Drain

Table 2-5: Pin List and Function Table - GPIO

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F8	F1	44	0	UA1RTS	UART1 Request to Send (RTS)	Output
			1	M0nCE2	SPI Master 0 Chip Enable 2	Output
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO44	General Purpose I/O	Input/ Output
			4	M4WIR3	SPI Master 4 I/O pin for 3-wire mode	Bidirectional 3-state
			5	M4MOSI	SPI Master 4 Data Out	Output
			6	M5nCE6	SPI Master 5 Chip Enable 6	Output
			7	SLWIR3LB	SPI Slave I/O pin for 3-wire	Bidirectional 3-state
F1	-	45	0	UA1CTS	UART1 Clear to Send (CTS)	Input
			1	M0nCE3	SPI Master 0 Chip Enable 3	Output
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO45	General Purpose I/O	Input/ Output
			4	M4nCE3	SPI Master 4 Chip Enable 3	Output
			5	M3nCE6	SPI Master 3 Chip Enable 6	Output
			6	M5nCE5	SPI Master 5 Chip Enable 5	Output
			7	SWO	Serial Wire Debug	Output
G3		46	0	32KHz_XT	32kHz Clock	Output
			1	M0nCE4	SPI Master 0 Chip Enable 4	Output
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO46	General Purpose I/O	Input/ Output
			4	TCTA1	Timer/counter A1	Output
			5	M5nCE4	SPI Master 5 Chip Enable 4	Output
			6	M4nCE4	SPI Master 4 Chip Enable 4	Output
			7	SWO	Serial Wire Debug	Output

Table 2-5: Pin List and Function Table - GPIO

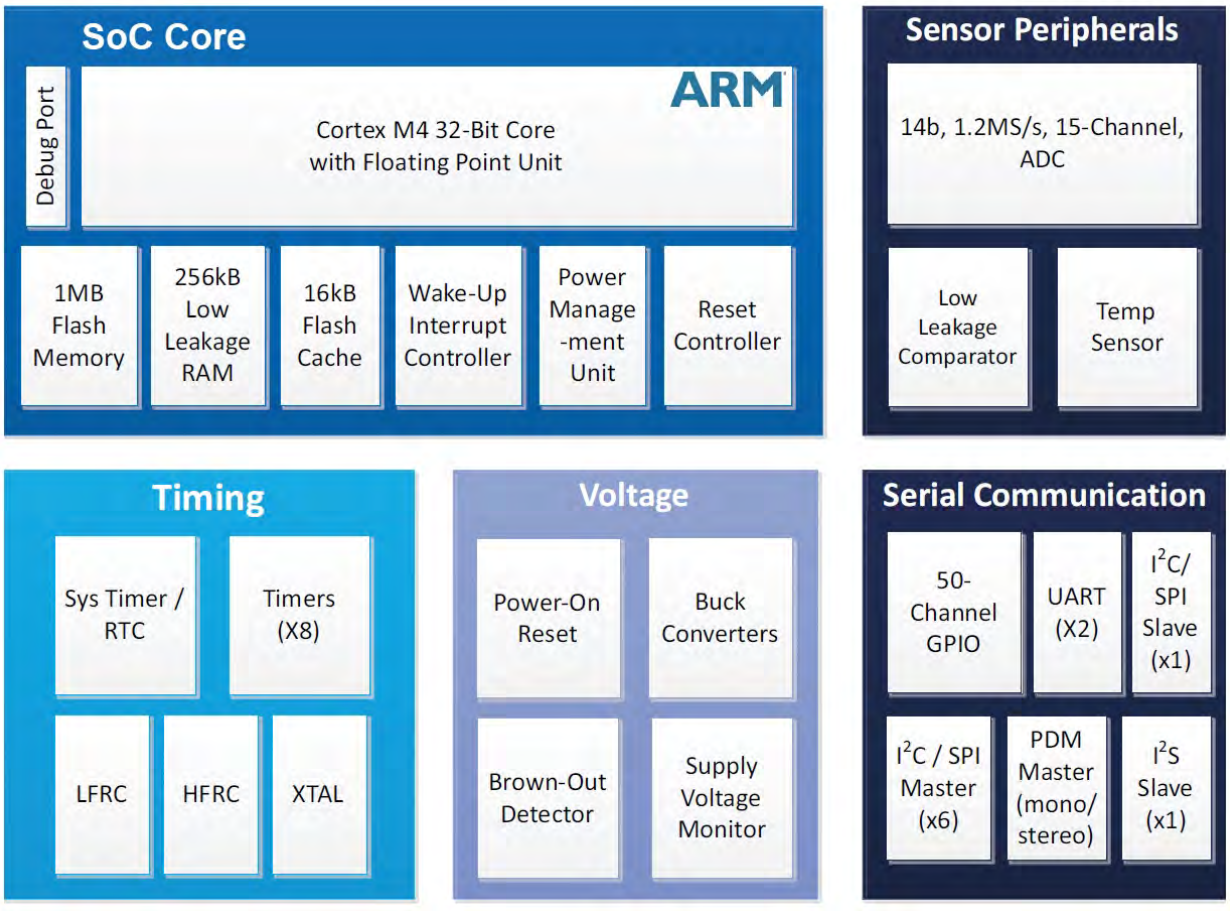
BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F4	E6	47	0	M2nCE5	SPI Master 2 Chip Enable 5	Output
			1	M0nCE5	SPI Master 0 Chip Enable 5	Output
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO47	General Purpose I/O	Input/ Output
			4	M5WIR3	SPI Master 5 I/O pin for 3-wire mode	Bidirectional 3-state
			5	M5MOSI	SPI Master 5 Data out	Output
			6	M4nCE5	SPI Master 4 Chip Enable 5	Output
			7	SLWIR3LB	SPI Slave I/O pin for 3-wire (loopback)	Bidirectional 3-state
E3	E7	48	0	M2nCE6	SPI Master 2 Chip Enable 6	Output
			1	M0nCE6	SPI Master 0 Chip Enable 6	Output
			2	TCTA3	Timer/Counter A3	Output
			3	GPIO48	General Purpose I/O	Input/ Output
			4	M5SCL	I^2C Master 5 Clock	Bidirectional Open Drain
			5	M5SCK	SPI Master 5 Clock	Output
			6	M5SCKLB	SPI M5 CLKIOUT->CLKIN (loopback)	Input
			7	M5SCLLB	I^2C M5 CLKIOUT->CLKIN (loopback)	Input
G1	F6	49	0	M2nCE7	SPI Master 2 Chip Enable 7	Output
			1	M0nCE7	SPI Master 0 Chip Enable 7	Output
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO49	General Purpose I/O	Input/ Output
			4	M5SDA	I^2C Master 5 Data	Bidirectional Open Drain
			5	M5MISO	SPI Master 5 Data in	Input
			6	SLMISOLB	SPI Slave Output Data	Output
			7	SLSDALB	I^2C Slave I/O Data (loopback)	Bidirectional Open Drain

SECTION

3

System Core

Figure 3-1: Block Diagram for the Ultra-Low Power Apollo2 SoC



The ultra-low power Apollo2 SoC, shown in Figure 3-1, is an ideal solution for battery-powered applications requiring sensor measurement and data analysis. In a typical system, the Apollo SoC serves as an applications processor for one or more sensors and/or radios. The SoC can

measure analog sensor outputs using an integrated 14 bit ADC and digital sensor outputs using the integrated serial master ports. The Cortex-M4 core with Floating Point Unit (referred to throughout this document as “M4”, “M4 Core” or “Cortex-M4”) integrated in the Apollo SoC is capable of running complex data analysis and sensor fusion algorithms to process the sensor data. The Cortex-M4 core with FPU also enables accelerated time-tomarket since application code may be efficiently executed in floating point form without the need to perform extensive fixed point optimizations. In other configurations, a host processor can communicate with the SoC over its serial slave port using the I^2C, SPI or I^2S protocol.

With unprecedented energy efficiency for sensor conversion and data analysis, the Apollo2 SoC enables months and years of battery life for products only achieving days or months of battery life today. For example, a fitness monitoring device with days or weeks of life on a rechargeable battery could be redesigned to achieve a year or more of life on a non-rechargeable battery. Similarly the Apollo SoCs enable the use of more complex sensor processing algorithms due to its extremely low active mode power of less than 10 µA/MHz. By using the Apollo SoCs, the aforementioned fitness monitoring device could achieve the current multi-day or multi-week battery life while adding new computation-intensive functions like context detection and gesture recognition.

At the center of the Apollo2 SoC is a 32-bit Arm Cortex-M4 processor with Floating Point Unit with several tightly coupled peripherals. The Ambiq implementation of the Cortex-M4 core delivers both greater performance and much lower power than 8-bit, 16-bit, and other comparable 32-bit cores. Code and data may be stored in the 1 MB Flash Memory and the 256 KB Low Leakage SRAM. The Wake-Up Interrupt Controller (WIC) coupled with the Cortex-M4 supports sophisticated and configurable sleep state transitions with a variety of interrupt sources.

A rich set of sensor peripherals enable the monitoring of several sensors. An integrated temperature sensor enables the measurement of ambient temperature. A scalable ultra-low power Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) monitors the temperature sensor, several internal voltages, and up to eight external sensor signals. The ADC is uniquely tuned for minimum power with a configurable measurement mode that does not require SoC intervention. In addition to integrated analog sensor peripherals, I^2C/SPI/PDM master ports and/or UART ports enables the SoC to communicate with external sensors and radios (such as Bluetooth transceivers) that have digital outputs.

The Apollo2 SoC also includes a set of timing peripherals and an RTC which is based on Ambiq’s AM08XX and AM18XX Real-Time Clock (RTC) families. The general purpose Timer/Counter Module (CTIMER), 32-bit System Timer (STIMER), and the RTC may be driven independently by one of three different clock sources: a low frequency RC oscillator, a high frequency RC oscillator, and a 32.768 kHz crystal (XTAL) oscillator. These clock sources use the proprietary advanced calibration techniques developed for the AM08XX and AM18XX products that achieve XTAL-like accuracy with RC-like power. Additionally, the Apollo SoC includes clock reliability func-

tions first offered in the AM08XX and AM18XX products. For example, the RTC can automatically switch from an XTAL source to an RC source in the event of an XTAL failure.

As with any Arm-based SoC, the Apollo2 SoC is supported by a complete suite of standard software development tools. Ambiq provides drivers for all peripherals along with basic application code to shorten development times. Software debug is facilitated by the addition of an instrumentation Trace Macrocell (ITM), a Trace Port Interface Unit (TPIU) and through the use of a Serial Wire Debugger interface (SWD).

SECTION

4

SoC Core Details

4.1 Functional Overview

At the center of the Apollo2 SoC is a 32-bit Arm Cortex-M4 core with the floating point option. This 3-stage pipeline implementation of the Arm v7-M architecture offers highly efficient processing in a very low power design. The Arm M DAP enables debugging access via a Serial Wire Interface from outside of the SoC which allows access to all of the memory and peripheral devices of the SoC.

The M4 core offers some other advantages including:

- Single 4 GB memory architecture with all Peripherals being memory-mapped
- Low-Power Consumption Modes:
  - Active
  - Sleep
  - Deep-Sleep
  - Power-Off
- Interrupts and Events
  - NVIC – interrupt controller
  - WIC – Wake-Up Interrupt Controller
  - Sleep-on-Exit (reduces interrupt overhead, used in an ISR SW structure)
  - WFI (enter sleep modes, wait for interrupts)

The following sections provide behavioral and performance details about each of the peripherals controlled by the SoC core. Where multiple instances of a peripheral exist on Apollo2 SoC (e.g., thesix I^2C/SPI master modules), base memory addresses for the registers are provided for each and noted as INSTANCE 0, INSTANCE 1, etc.

4.2 Interrupts

Within the SoC, multiple peripherals can generate interrupts. In some cases, a single peripheral may be able to generate multiple different interrupts. Each interrupt signal generated by a peripheral is connected back to the M4 core in two places. First, the interrupts are connected to the Nested Vectored Interrupt Controller, NVIC, in the core. This connection provides the standard changes to program flow associated with interrupt processing. Additionally, they are connected to the WIC outside of the core, allowing the interrupt sources to wake the M4 core when it is in a deep sleep (SRPG) mode.

The SoC supports the M4 NMI as well as the normal interrupt types. For details on the Interrupt model of the M4, see the *Cortex-M4 Devices Generic User Guide*, document number DUI0553A.

Table 4-1 is the M4 Vector Table for Apollo2 SoC.

Table 4-1: Arm Cortex-M4 Vector Table for Apollo2 SoC

Exception Number	IRQ Number	Offset	Vector	Peripheral/Description
256	239	0x03FC	IRQ239	
.		.	.	
.		.	.	
.		0x00C4	IRQ28-31	SW INT[0-3]
.		0x00C0	IRQ27	Reserved
.		0x00AC	IRQ19-26	Stimer Compare [0:7]
.		0x008C	IRQ18	Stimer Capture/Overflow
.		0x0088	IRQ17	PDM
.		0x0084	IRQ16	ADC
.		0x0080	IRQ15	UART1
.		0x007C	IRQ14	UART0
.		0x0078	IRQ13	Counter/Timers
.		0x0074	IRQ12	GPIO
.		0x0070	IRQ11	I^2C/SPI Master 5
.		0x006C	IRQ10	I^2C/SPI Master 4
.		0x0068	IRQ9	I^2C/SPI Master 3
.		0x0064	IRQ8	I^2C/SPI Master 2
.		0x0060	IRQ7	I^2C/SPI Master 1
		0x005C	IRQ6	I^2C/SPI Master 0
		0x0058	IRQ5	I^2C/SPI Slave Register Access
		0x0054	IRQ4	I^2C/SPI Slave
		0x0050	IRQ3	Voltage Comparator
18	2	0x004C	IRQ2	Clock Control and RTC
17	1	0x0048	IRQ0	Brownout Detection

Table 4-1: Arm Cortex-M4 Vector Table for Apollo2 SoC *(Continued)*

Exception Number	IRQ Number	Offset	Vector	Peripheral/Description
16	0	0x0044	Systick	
15	-1	0x0040	PendSV	
14	-2	0x003C	Reserved	
13		0x0038	Reserved for Debug	
12				
11	-5	0x002C	SVCall	
10			Reserved	
9				
8				
7				
6	-10		Usage Fault	
5	-11	0x0014	Bus Fault	
4	-12	0x0010	Memory Management Fault	
3	-13	0x000C	Hard Fault	
2	-14	0x0008	NMI	Unused
1		0x0004	Reset	
		0x0000	Initial SP value	

The Cortex-M4 allows the user to assign various interrupts to different priority levels based on the requirements of the application. In this SoC implementation, 8 different priority levels are available.

One additional feature of the M4 interrupt architecture is the ability to relocate the Vector Table to a different address. This could be useful if the application requires a different set of interrupt service routines for a particular mode of an application. The software could move the Vector Table into SRAM and reassign the interrupt service routine entry addresses as needed.

Hardware interrupts are assigned in the SoC to the M4 NVIC as shown below.

Table 4-2: SoC Interrupt Assignments

IRQ	Peripheral/Description
NMI	Unused
IRQ0	Brownout Detection
IRQ1	Watchdog Timer
IRQ2	Clock Control and RTC
IRQ3	Voltage Comparator

Table 4-2: SoC Interrupt Assignments *(Continued)*

IRQ	Peripheral/Description
IRQ4	I^2C / SPI Slave
IRQ5	I^2C / SPI Slave Register Access
IRQ6	I^2C / SPI Master0
IRQ7	I^2C / SPI Master1
IRQ8	I^2C / SPI Master2
IRQ9	I^2C / SPI Master3
IRQ10	I^2C / SPI Master4
IRQ11	I^2C / SPI Master5
IRQ12	GPIO
IRQ13	Counter/Timers
IRQ14	UART0
IRQ15	UART1
IRQ16	ADC
IRQ17	PDM
IRQ18	STimer Capture/Overflow
IRQ19-26	STimer Compare[0:7]
IRQ27	Reserved
IRQ28-31	SW INT[0-3]

4.3 Memory Map

Arm has a well-defined memory map for devices based on the Arm v7-M Architecture. The M4 further refines this map in the area of the Peripheral and System address ranges. Below is the system memory map as defined by Arm:

Table 4-3: Arm Cortex-M4 Memory Map

Address	Name	Executable	Description
0x00000000 – 0x1FFFFFFF	Code	Y	Flash Memory
0x20000000 – 0x3FFFFFFF	Reserved	N	Reserved
0x40000000 – 0x5FFFFFFF	Peripheral	N	On-chip peripheral address space
0x60000000 – 0x9FFFFFFF	External RAM	Y	External / Off-chip Memory

Table 4-3: Arm Cortex-M4 Memory Map *(Continued)*

Address	Name	Executable	Description
0xA0000000 – 0xDFFFFFFF	External Device	N	External device memory
0xE0000000 – 0xE00FFFFF	Private Peripheral Bus	N	NVIC, System timers, System Control Block
0xE0100000 – 0xFFFFFFFF	Vendor_SYS	N	Vendor Defined

The SoC-specific implementation of this memory map is as follows:

Table 4-4: SoC System Memory Map

Address	Name	Executable	Description
0x00000000 – 0x000FFFFF	Flash	Y	Flash Memory
0x00100000 – 0x07FFFFFF	Reserved	X	No device at this address range
0x08000000 – 0x08000FFF	Boot Loader ROM	Y	Execute Only Boot Loader and Flash Helper Functions.
0x08001000 – 0x0FFFFFFF	Reserved	X	No device at this address range
0x10000000 – 0x1003FFFF	SRAM	Y	Low-power SRAM
0x10040000 – 0x3FFFFFFF	Reserved	X	No device at this address range
0x40000000 – 0x4FFFFFFF	Peripheral – APB	N	APB Peripheral devices
0x50000000 – 0x5FFFFFFF	Peripheral – AHB	N	AHB Peripheral devices
0x60000000 – 0xDFFFFFFF	Reserved	X	No device at this address range
0xE0000000 – 0xE00FFFFF	PPB	N	NVIC, System timers, System Control Block
0xE0100000 – 0xEFFFFFFF	Reserved	X	No device at this address range
0xF0000000 – 0xF0000FFF	Reserved	X	No device at this address range
0xF0001000 – 0xFFFFFFFF	Reserved	X	No device at this address range

Peripheral devices within the memory map are allocated on 4 KB boundaries, allowing each device up to 1024 32-bit control and status registers. Peripherals will return undefined read data when an attempt to access a register which does not exist occurs. Peripherals, whether accessed via the APB or the AHB, will always accept any write data sent to their registers without attempting to return an ERROR response. Specifically, a write to a read-only register would just become a don't-care write.

Table 4-5 on page 69 shows the address mapping for the peripheral devices of the Base Platform.

Table 4-5: SoC Peripheral Device Memory Map

Address	Device
0x40000000 – 0x400003FF	Reset / BoD Control
0x40000400 – 0x40003FFF	Reserved
0x40004000 – 0x400041FF	Clock Generator / RTC
0x40004200 – 0x40007FFF	Reserved
0x40008000 – 0x400083FF	Timers
0x40008400 – 0x4000BFFF	Reserved
0x4000C000 – 0x4000C3FF	Voltage Comparator
0x4000C400 – 0x4000FFFF	Reserved
0x40010000 – 0x400103FF	GPIO Control
0x40010400 – 0x40017FFF	Reserved
0x40018000 – 0x40018FFF	Flash Cache Control
0x40019000 – 0x4001BFFF	Reserved
0x4001C000 – 0x4001C3FF	UART0
0x4001C400 – 0x4001CFFF	Reserved
0x4001D000 – 0x4001D3FF	UART1
0x4001D400 – 0x4001FFFF	Reserved
0x40020000 – 0x400203FF	Miscellaneous Control
0x40020400 – 0x40020FFF	Reserved
0x40021000 – 0x400213FF	Power Control
0x40021400 – 0x40023FFF	Reserved
0x40024000 – 0x400243FF	Watchdog Timer
0x40024400 – 0x4FFFFFFF	Reserved
0x50000000 – 0x500003FF	I^2C / SPI Slave
0x50000400 – 0x50003FFF	Reserved
0x50004000 – 0x500043FF	I^2C / SPI Master0
0x50004400 – 0x50004FFF	Reserved
0x50005000 – 0x500053FF	I^2C / SPI Master1
0x50005400 – 0x50005FFF	Reserved
0x50006000 – 0x500063FF	I^2C / SPI Master2

Table 4-5: SoC Peripheral Device Memory Map *(Continued)*

Address	Device
0x50006400 – 0x50006FFF	Reserved
0x50007000 – 0x500073FF	I^2C / SPI Master3
0x50007400 – 0x50007FFF	Reserved
0x50008000 – 0x500083FF	I^2C / SPI Master4
0x50008400 – 0x50008FFF	Reserved
0x50009000 – 0x500093FF	I^2C / SPI Master5
0x50009400 – 0x5000FFFF	Reserved
0x50010000 – 0x500103FF	ADC
0x50010400 – 0x50010FFF	Reserved
0x50011000 – 0x500113FF	PDM
0x50011400 – 0x5001FFFF	Reserved
0x50020000 – 0x5002FFFF	Flash OTP
0x50030000 – 0x5FFFFFFF	Reserved

4.4 Memory Protection Unit (MPU)

The Apollo2 SoC includes an MPU which is a core component for memory protection. The M4 processor supports the standard Armv7 Protected Memory System Architecture model. The MPU provides full support for:

- Protection regions
- Overlapping protection regions, with ascending region priority:
  - 7 = highest priority
  - 0 = lowest priority
- Access permissions
- Exporting memory attributes to the system.

MPU mismatches and permission violations invoke the programmable-priority MemManage fault handler. See the *Arm®v7-M Architecture Reference Manual* for more information.

Use the MPU to:

- Enforce privilege rules
- Separate processes
- Enforce access rules

4.5 System Buses

The Arm Cortex-M4 utilizes 3 instances of the AMBA AHB bus for communication with memory and peripherals. The ICode bus is designed for instruction fetches from the 'Code' memory space while the DCode bus is designed for data and debug accesses in that same region. The System bus is designed for fetches to the SRAM and other peripheral devices of the SoC.

The Apollo2 SoC maps the available SRAM memory onto an address space within the 'Code' memory space. This gives the user the opportunity to perform instruction and data fetches from the lower-power SRAM to effectively lower the power consumption of the SoC.

The peripherals of the Apollo2 SoC which are infrequently accessed are located on an AMBA APB bus. A bridge exists which translates the accesses from the System AHB to the APB. Accesses to these peripherals will inject a single wait-state on the AHB during any access cycle.

4.6 Power Management

The Power Management Unit (PMU) is a finite-state machine that controls the transitions of the SoC between power modes. When moving from Active Mode to Deep Sleep Mode, the PMU manages the state-retention capability of the registers within the Cortex-M4 core and also controls the shutdown of the voltage regulators of the SoC. Once in the Deep Sleep Mode, the PMU, in conjunction with the Wake-Up

Interrupt Controller, waits for a wakeup event. When the event is observed, the PMU begins the power restoration process by re-enabling the on-chip voltage regulators and restoring the CPU register state. The M4 is then returned to active mode once all state is ready.

The Apollo2 SoC power modes are described in the subsequent discussion along with the operation of the PMU.

4.6.1 Cortex-M4 Power Modes

The Arm Cortex-M4 defines the following 3 power modes:

- Active
- Sleep
- Deep Sleep

In addition to the above Arm-defined modes, the Apollo2 SoC will support a Shutdown mode in which the entire device is powered down except for the logic required to support a Power-On Reset.

Each mode is described below.

4.6.1.1 Active Mode

In the Active Mode, the M4 is powered up, clocks are active, and instructions are being executed. In this mode, the M4 expects all (enabled) devices attached to the AHB and APB to be powered and clocked for normal access. All of the non-debug Arm clocks (FCLK, HCLK) are active in this state.

To transition from the Active Mode to any of the lower-power modes, a specific sequence of instructions is executed on the M4 core. First, specific bits in the Arm v7-M System Control Register must be set to determine the mode to enter. See page B3-269 of the *Arm v7-M Architecture Reference Manual* for more details.

After the SCR is setup, code can enter the low-power states using one of the 2 following methods:

- Execute a Wait-For-Interrupt (WFI) instruction.
- Set the SLEEPONEXIT bit of the SCR such that the exit from an ISR will automatically return to a sleep state.

The M4 will enter a low-power mode after one of these are performed (assuming all conditions are met) and remain there until some event causes the core to return to Active Mode. The possible reasons to return to Active Mode are:

- A reset
- An enabled Interrupt is received by the NVIC
- A Debug Event is received from the DAP

4.6.1.2 Sleep Mode

In the Sleep Mode, the M4 is powered up, but the clocks (HCLK, FCLK) are not active. The power supply is still applied to the M4 logic such that it can immediately become active on a wakeup event and begin executing instructions.

4.6.1.3 Deep Sleep Mode

In the Deep Sleep Mode, the M4 enters SRPG mode where the main power is removed, but the flops retain their state. The clocks are not active, and the SoC clock sources for HCLK and FCLK can be deactivated. To facilitate the removal of the source supply and entry into SRPG mode, the M4 will handshake with the Wake-up Interrupt Controller and Power Management Unit and set up the possible wakeup conditions.

4.6.2 System Power Modes

In addition to the CPU power states, there are system power states defined as follows.

4.6.2.1 SYS Active Burst (S_{ACTB})

CPU is in Active Mode and executing instructions. All peripheral devices are on and available.

4.6.2.2 SYS Active (S_{ACT})

CPU is in Active Mode and executing instructions. All peripheral devices are on and available.

4.6.2.3 SYS Sleep Mode 0 (S_{S0})

In SYS Sleep Mode 0, this is a low power state for the SoC. In this mode, all SRAM memory is retained (up to 384KB), Flash memory is in standby, HFRC is on, main core clock domain is gated but peripheral clock domains can be on. CPU is in Sleep Mode.

This state can be entered if a peripheral device (such as SPI/UART/I^2C) is actively transferring data and the time window is sufficient for CPU to enter Sleep Mode but is not long enough to go into a Deep Sleep Mode.

4.6.2.4 SYS Sleep Mode 1 (S_{S1})

In SYS Sleep Mode 1, this is a low power state for the SoC. In this mode, all SRAM memory is retained, Flash memory is in standby, HFRC is on, all functional clocks are gated. CPU is in Sleep Mode.

This state can be entered if a no peripheral device (SPI/UART/I^2C) is actively transferring data, however, communication may occur within a short time window which will prevent the CPU from entering Deep Sleep Mode (and subsequently the system from entering a lower power state).

This state is also referred to as “Active Idle”. In other words, all power domains are powered on, but all clocks are gated. This state is a good power baseline for the system as it represents the active mode DC power level. Typically the power in this state is dominated by leakage, and always-on functional blocks.

4.6.2.5 SYS Deep Sleep Mode 0 (S_{DS0})

In SYS Deep Sleep Mode 0, this is a deep low power state for the SoC. In this mode, SRAM is in retention (capacity controlled by software), cache memory is in retention (16 KB), Flash memory is in power down, HFRC is on, main core power domain is off but peripheral power domains can be on. CPU is in Deep Sleep. Core logic state is retained.

This state can be entered if a peripheral device (SPI/UART/I^2C) is actively (or intermittently) transferring data but the window of acquisition is long enough to allow the CPU to go into a deeper low power state.

4.6.2.6 SYS Deep Sleep Mode 1 (S_{DS1})

In SYS Deep Sleep Mode 1, this is a deep low power state for the SoC. In this mode, SRAM is in retention (capacity controlled by software), cache memory is powered down, Flash memory is in power down, HFRC is on, main core power domain is off but peripheral power domains can be on. CPU is in Deep Sleep. Core logic state is retained.

This state can be entered if the latency to warm up the cache can be tolerated. This could be an extended wait for peripheral communication event.

4.6.2.7 SYS Deep Sleep Mode 2 (S_{DS2})

In SYS Deep Sleep Mode 2, this is the minimum power state that the SoC can resume normal operation.

In this mode, minimal SRAM memory is retained as needed for software to resume (note that SRAM can have 0-256 KB in retention depending on the software/system functional and latency requirements), Cache is powered off (no retention), Flash memory is in power down, HFRC is off, XTAL is ON, all internal switched power domains are off/gated. CPU is in Deep Sleep. Core logic state is retained.

This state can be entered when all activity has suspended for a duration of time sufficient to sustain the longer exit latencies to resume. This could be a state where periodic data samples are taken and the data is locally processed/accumulated/transferred at long time intervals. This state can only be entered (vs S_{DS1}) if the peripheral devices are either not enabled/active or if the application can afford to save/restore the state of the controller(s) on entry/exit of this mode.

4.6.2.8 SYS Deep Sleep Mode 3 (S_{DS3})

In SYS Deep Sleep Mode 3, this is a deep sleep power state for the SoC. In this mode, no memory is in retention, all memory is powered down, LFRC is on (HFRC

and XTAL are off), all internal switched power domains are off/gated. CPU is in Deep Sleep. Core logic state is retained. Single timer is running.

This state can be entered on long inactivity periods. Also can be used for very low power ADC sampling without CPU interaction.

4.6.2.9 SYS OFF Mode (SOFF)

In SYS OFF Mode, SoC is completely powered down with no power supplied. CPU is in shutdown mode with no state retention. Only Flash memory is retained.

This mode is controlled external to the SoC by removing power to the device.

4.6.3 Power Control

The Power Control block provides control and status for the power state of all the power domains and voltage regulators in the SoC. Software can control these blocks via power control registers within this block.

The power control block controls the power sequence to power up or power down a particular peripheral device and memory power domain. Status of each of these can be monitored in the respective power control status register. The power controller also supports event notification to indicate peripheral power transition completion. Event notification is the preferred power-optimized method in lieu of status polling.

The power controller is also the primary control block for the Buck converters as well as the LDO regulators when Bucks are disabled. Similarly, event notification is supported to provide the appropriate handshake to software as needed as well as status register indicators.

This block handles all power sequencing during initial power on and all power mode transitions.

4.6.4 PWRCTRL Registers

PWR Controller Register Bank
INSTANCE 0 BASE ADDRESS:0x40021000
Power Controller register Bank - this is the place SW writes to.

4.6.4.1 Register Memory Map

Table 4-6: PWRCTRL Register Map

Address(es)	Register Name	Description
0x40021000	SUPPLYSRC	Memory and Core Voltage Supply Source Select Register
0x40021004	POWERSTATUS	Power Status Register for SoC supplies and peripherals
0x40021008	DEVICEEN	Device Enables for Apollo2 modules
0x4002100C	SRAMPWDINSLEEP	Power down an SRAM Banks in Deep Sleep mode
0x40021010	MEMEN	Disables individual banks of the MEMORY array
0x40021014	PWRONSTATUS	POWER ON Status
0x40021018	SRAMCTRL	SRAM Control register
0x4002101C	ADCSTATUS	Power Status Register for ADC Block
0x40021020	MISCOPT	Power Optimization Control Bits

4.6.4.2 PWRCTRL Registers

4.6.4.2.1 SUPPLYSRC Register

Memory and Core Voltage Supply Source Select Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40021000
Memory and Core Voltage Supply Source Select Register

Table 4-7: SUPPLYSRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													SWITCH_LDO_IN_SLEE	COREBUCKEN	MEMBUCKEN

Table 4-8: SUPPLYSRC Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	SWITCH_LDO_IN_SLEEP	0x1	RW	Switches the CORE DOMAIN from BUCK mode (if enabled) to LDO when CPU is in DEEP SLEEP. If all the devices are off then this does not matter and LDO (low power mode) is used EN = 0x1 - Automatically switch from CORE BUCK to CORE LDO when CPU is in DEEP SLEEP
1	COREBUCKEN	0x0	RW	Enables and Selects the Core Buck as the supply for the low-voltage power domain. EN = 0x1 - Enable the Core Buck for the low-voltage power domain.
0	MEMBUCKEN	0x0	RW	Enables and select the Memory Buck as the supply for the Flash and SRAM power domain. EN = 0x1 - Enable the Memory Buck as the supply for flash and SRAM.

4.6.4.2.2 POWERSTATUS Register

Power Status Register for MCU supplies and peripherals
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40021004
Power Status Register for MCU supplies and peripherals

Table 4-9: POWERSTATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														COREBUCKON	MEMBUCKON

Table 4-10: POWERSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	COREBUCKON	0x0	RO	Indicates whether the Core low-voltage domain is supplied from the LDO or the Buck. LDO = 0x0 - Indicates the the LDO is supplying the Core low-voltage. BUCK = 0x1 - Indicates the the Buck is supplying the Core low-voltage.
0	MEMBUCKON	0x0	RO	Indicate whether the Memory power domain is supplied from the LDO or the Buck. LDO = 0x0 - Indicates the LDO is supplying the memory power domain. BUCK = 0x1 - Indicates the Buck is supplying the memory power domain.

4.6.4.2.3 DEVICEEN Register

DEVICE ENABLES
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40021008
DEVICE ENABLES for Apollo2 Modules

Table 4-11: DEVICEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					PDM	ADC	UART1	UART0	IO_MASTER5	IO_MASTER4	IO_MASTER3	IO_MASTER2	IO_MASTER1	IO_MASTER0	IO_SLAVE

Table 4-12: DEVICEEN Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10	PDM	0x0	RW	Enable PDM Digital Block EN = 0x1 - Enable PDM DIS = 0x0 - Disables PDM
9	ADC	0x0	RW	Enable ADC Digital Block EN = 0x1 - Enable ADC DIS = 0x0 - Disables ADC
8	UART1	0x0	RW	Enable UART 1 EN = 0x1 - Enable UART 1 DIS = 0x0 - Disables UART 1
7	UART0	0x0	RW	Enable UART 0 EN = 0x1 - Enable UART 0 DIS = 0x0 - Disables UART 0
6	IO_MASTER5	0x0	RW	Enable IO MASTER 5 EN = 0x1 - Enable IO MASTER 5 DIS = 0x0 - Disables IO MASTER 5
5	IO_MASTER4	0x0	RW	Enable IO MASTER 4 EN = 0x1 - Enable IO MASTER 4 DIS = 0x0 - Disables IO MASTER 4
4	IO_MASTER3	0x0	RW	Enable IO MASTER 3 EN = 0x1 - Enable IO MASTER 3 DIS = 0x0 - Disables IO MASTER 3
3	IO_MASTER2	0x0	RW	Enable IO MASTER 2 EN = 0x1 - Enable IO MASTER 2 DIS = 0x0 - Disables IO MASTER 2
2	IO_MASTER1	0x0	RW	Enable IO MASTER 1 EN = 0x1 - Enable IO MASTER 1 DIS = 0x0 - Disables IO MASTER 1

Table 4-12: DEVICEEN Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	IO_MASTER0	0x0	RW	Enable IO MASTER 0 EN = 0x1 - Enable IO MASTER 0 DIS = 0x0 - Disables IO MASTER 0
0	IO_SLAVE	0x0	RW	Enable IO SLAVE EN = 0x1 - Enable IO SLAVE DIS = 0x0 - Disables IO SLAVE

4.6.4.2.4 SRAMPWDINSLEEP Register

Powerdown an SRAM Banks in Deep Sleep mode
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4002100C
Powerdown an SRAM Banks in Deep Sleep mode

Table 4-13: SRAMPWDINSLEEP Register

31	30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
CACHE_PWD_SLP	RSVD / SRAMSLEEPPOWERDOWN

Table 4-14: SRAMPWDINSLEEP Register Bits

Bit	Name	Reset	RW	Description
31	CACHE_P-WD_SLP	0x0	RW	Enable CACHE BANKS to power down in deep sleep EN = 0x1 - CACHE BANKS POWER DOWN in CORE SLEEP DIS = 0x0 - CACHE BANKS STAYS in Retention in CORE SLEEP

Table 4-14: SRAMPWDINSLEEP Register Bits

Bit	Name	Reset	RW	Description
30:11	RSVD	0x0	RO	RESERVED
10:0	SRAMSLEEP-POWERDOWN	0x0	RW	Selects which SRAM banks are powered down in deep sleep mode, causing the contents of the bank to be lost. NONE = 0x0 - All banks retained GROUP0_SRAM0 = 0x1 - 0KB-8KB SRAM GROUP0_SRAM1 = 0x2 - 8KB-16KB SRAM GROUP0_SRAM2 = 0x4 - 16KB-24KB SRAM GROUP0_SRAM3 = 0x8 - 24KB-32KB SRAM GROUP1 = 0x10 - 32KB-64KB SRAMs GROUP2 = 0x20 - 64KB-96KB SRAMs GROUP3 = 0x40 - 96KB-128KB SRAMs GROUP4 = 0x80 - 128KB-160KB SRAMs GROUP5 = 0x100 - 160KB-192KB SRAMs GROUP6 = 0x200 - 192KB-224KB SRAMs GROUP7 = 0x400 - 224KB-256KB SRAMs SRAM8K = 0x1 - Do not Retain lower 8KB SRAM16K = 0x3 - Do not Retain lower 16KB SRAM32K = 0xF - Do not Retain lower 32KB SRAM64K = 0x1F - Do not Retain lower 64KB SRAM128K = 0x7F - Do not Retain lower 128KB ALLBUTLOWER8K = 0x7FE - All banks but lower 8k powered down. ALLBUTLOWER16K = 0x7FC - All banks but lower 16k powered down. ALLBUTLOWER24K = 0x7F8 - All banks but lower 24k powered down. ALLBUTLOWER32K = 0x7F0 - All banks but lower 32k powered down. ALLBUTLOWER64K = 0x7E0 - All banks but lower 64k powered down. ALLBUTLOWER128K = 0x780 - All banks but lower 128k powered down. ALL = 0x7FF - All banks powered down.

4.6.4.2.5 MEMEN Register

Disables individual banks of the MEMORY array
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40021010
Disables individual banks of the MEMORY array

Table 4-15: MEMEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CACHEB2	RSVD	CACHEB0	RSVD																FLASH1	FLASH0	SRAMEN										

Table 4-16: MEMEN Register Bits

Bit	Name	Reset	RW	Description
31	CACHEB2	0x1	RW	Enable CACHE BANK 2 EN = 0x1 - Enable CACHE BANK 2 DIS = 0x0 - Disable CACHE BANK 2
30	RSVD	0x0	RO	Reserved
29	CACHEB0	0x1	RW	Enable CACHE BANK 0 EN = 0x1 - Enable CACHE BANK 0 DIS = 0x0 - Disable CACHE BANK 0

Table 4-16: MEMEN Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
28:13	RSVD	0x0	RO	RESERVED
12	FLASH1	0x1	RW	Enable FLASH1 EN = 0x1 - Enable FLASH1 DIS = 0x0 - Disables FLASH1
11	FLASH0	0x1	RW	Enable FLASH 0 EN = 0x1 - Enable FLASH 0 DIS = 0x0 - Disables FLASH 0
10:0	SRAMEN	0x7ff	RW	Enables power for selected SRAM banks (else an access to its address space to generate a Hard Fault). NONE = 0x0 - All banks disabled GROUP0_SRAM0 = 0x1 - 0KB-8KB SRAM GROUP0_SRAM1 = 0x2 - 8KB-16KB SRAM GROUP0_SRAM2 = 0x4 - 16KB-24KB SRAM GROUP0_SRAM3 = 0x8 - 24KB-32KB SRAM GROUP1 = 0x10 - 32KB-64KB SRAMs GROUP2 = 0x20 - 64KB-96KB SRAMs GROUP3 = 0x40 - 96KB-128KB SRAMs GROUP4 = 0x80 - 128KB-160KB SRAMs GROUP5 = 0x100 - 160KB-192KB SRAMs GROUP6 = 0x200 - 192KB-224KB SRAMs GROUP7 = 0x400 - 224KB-256KB SRAMs SRAM8K = 0x1 - ENABLE lower 8KB SRAM16K = 0x3 - ENABLE lower 16KB SRAM32K = 0xF - ENABLE lower 32KB SRAM64K = 0x1F - ENABLE lower 64KB SRAM128K = 0x7F - ENABLE lower 128KB SRAM256K = 0x7FF - ENABLE lower 256KB ALL = 0x7FF - All banks ENABLED

4.6.4.2.6 PWRONSTATUS Register

POWER ON Status
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x40021014
POWER ON Status

Table 4-17: PWRONSTATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD										PD_CACHEB2	RSVD	PD_CACHEB0	PD_GRP7_SRAM	PD_GRP6_SRAM	PD_GRP5_SRAM	PD_GRP4_SRAM	PD_GRP3_SRAM	PD_GRP2_SRAM	PD_GRP1_SRAM	PD_GRP0_SRAM3	PD_GRP0_SRAM2	PD_GRP0_SRAM1	PD_GRP0_SRAM0	PDADC	PD_FLAM1	PD_FLAM0	PD_PDM	PDC	PDB	PDA	PD_MCUL

Table 4-18: PWRONSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:22	RSVD	0x0	RO	This bit field is reserved for future use.
21	PD_CACHEB2	0x0	RO	This bit is 1 if power is supplied to CACHE BANK 2
20	RSVD	0x0	RO	Reserved
19	PD_CACHEB0	0x0	RO	This bit is 1 if power is supplied to CACHE BANK 0
18	PD_GRP7_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP7
17	PD_GRP6_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP6
16	PD_GRP5_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP5
15	PD_GRP4_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP4
14	PD_GRP3_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP3
13	PD_GRP2_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP2
12	PD_GRP1_SRAM	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_GRP1
11	PD_GRP0_SRAM3	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_SRAM0_3
10	PD_GRP0_SRAM2	0x0	RO	This bit is 1 if power is supplied to SRAM domain PD_SRAM0_2
9	PD_GRP0_SRAM1	0x0	RO	This bit is 1 if power is supplied to SRAM domain SRAM0_1
8	PD_GRP0_SRAM0	0x0	RO	This bit is 1 if power is supplied to SRAM domain SRAM0_0
7	PDADC	0x0	RO	This bit is 1 if power is supplied to domain PD_ADC
6	PD_FLAM1	0x0	RO	This bit is 1 if power is supplied to domain PD_FLAM1
5	PD_FLAM0	0x0	RO	This bit is 1 if power is supplied to domain PD_FLAM0
4	PD_PDM	0x0	RO	This bit is 1 if power is supplied to domain PD_PDM
3	PDC	0x0	RO	This bit is 1 if power is supplied to power domain C, which supplies IOM3-5.
2	PDB	0x0	RO	This bit is 1 if power is supplied to power domain B, which supplies IOM0-2.
1	PDA	0x0	RO	This bit is 1 if power is supplied to power domain A, which supplies IOS and UART0,1.
0	PD_MCUL	0x0	RO	This bit is 1 if power is supplied to power domain PD_MCUL.

4.6.4.2.7 SRAMCTRL Register

SRAM Control register
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x40021018
SRAM Control register

Table 4-19: SRAMCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													SRAM_MASTER_CLKGATE	SRAM_CLKGATE	SRAM_LIGHT_SLEEP

Table 4-20: SRAMCTRL Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	This bit field is reserved for future use.
2	SRAM_MASTER_CLKGATE	0x0	RW	Enables top-level clock gating in the SRAM block. This bit should be enabled for lowest power operation. EN = 0x1 - Enable Master SRAM Clock Gate DIS = 0x0 - Disables Master SRAM Clock Gating
1	SRAM_CLKGATE	0x0	RW	Enables individual per-RAM clock gating in the SRAM block. This bit should be enabled for lowest power operation. EN = 0x1 - Enable Individual SRAM Clock Gating DIS = 0x0 - Disables Individual SRAM Clock Gating
0	SRAM_LIGHT_SLEEP	0x1	RW	Enable LS (light sleep) of cache RAMs. When this bit is set, the RAMS will be put into light sleep mode while inactive. Note: if the SRAM is actively used, this may have an adverse affect on power since entering/exiting LS mode may consume more power than would be saved. EN = 0x1 - Enable LIGHT SLEEP for SRAMs DIS = 0x0 - Disables LIGHT SLEEP for SRAMs

4.6.4.2.8 ADCSTATUS Register

Power Status Register for ADC Block
OFFSET: 0x0000001C
INSTANCE 0 ADDRESS: 0x4002101C
Power Status Register for ADC Block

Table 4-21: ADCSTATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										ADC_REFBUF_PWD	ADC_REFKEEP_PWD	ADC_VBAT_PWD	ADC_VPTAT_PWD	ADC_BGT_PWD	ADC_PWD

Table 4-22: ADCSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5	ADC_REFBUF_PWD	0x0	RO	This bit indicates that the ADC REFBUF is powered down
4	ADC_REFKEEP_PWD	0x0	RO	This bit indicates that the ADC REFKEEP is powered down
3	ADC_VBAT_PWD	0x0	RO	This bit indicates that the ADC VBAT resistor divider is powered down
2	ADC_VPTAT_PWD	0x0	RO	This bit indicates that the ADC temperature sensor input buffer0x0 is powered down
1	ADC_BGT_PWD	0x0	RO	This bit indicates that the ADC Band Gap is powered down
0	ADC_PWD	0x0	RO	This bit indicates that the ADC is powered down

4.6.4.2.9 MISCOPT Register

Power Optimization Control Bits
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x40021020
Power Optimization Control Bits

Table 4-23: MISCOPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													DIS_LDOLPMODE_TIMERS	DIS_LDOLPMODE_HFRC	ADC_EN_MASK

Table 4-24: MISCOPT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	DIS_LDOLPMODE_TIMERS	0x0	RW	Setting this bit will enable the MEM LDO to be in LPMODE during deep sleep even when the ctimers or stimers are running
1	DIS_LDOLPMODE_HFRC	0x0	RW	Setting this bit will enable the Core LDO to be in LPMODE during deep sleep even when HFRC is enabled.
0	ADC_EN_MASK	0x0	RW	Control Bit to mask the ADC_EN in the adc_p-wr_down equation.

4.7 Debug Interfaces

A number of useful debug facilities are provided in the Apollo2 SoC.

4.7.1 Debugger Attachment

An external debugger can be connected to the SoC using Arm's Serial Wire Debug (SWD) interface. The SWD interface is a 2-wire interface that is supported by a variety of off-the-shelf commercial debuggers, enabling customers to utilize their development environment of choice.

4.7.2 Instrumentation Trace Macrocell (ITM)

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

4.7.3 Trace Port Interface Unit (TPIU)

The Apollo2 SoC includes a Cortex-M4 Trace Port Interface Unit (TPIU). The Cortex-M4 TPIU is an Arm IP component that acts as a bridge between the on-chip trace data from the ITM and the single pin supporting the Serial Wire Viewer Protocol.

The TPIU includes a Trace Output Serializer that can format and send the SWV protocol in either a Manchester encoded form or as a standard UART stream.

4.7.4 Faulting Address Trapping Hardware

The Apollo2 SoC offers an optional facility for trapping the address associated with bus faults occurring on any of the three AMBA AHB buses on the chip. This facility must be specifically enabled so that energy is not wasted when one is not actively debugging.

4.8 ITM Registers

Arm ITM Registers.
INSTANCE 0 BASE ADDRESS:0x00000000

4.8.1 Register Memory Map

Table 4-25: ITM Register Map

Address(es)	Registered Name	Description
0xE0000000	STIM0	Stimulus Port Register 0
0xE0000004	STIM1	Stimulus Port Register 1
0xE0000008	STIM2	Stimulus Port Register 2
0xE000000C	STIM3	Stimulus Port Register 3
0xE0000010	STIM4	Stimulus Port Register 4
0xE0000014	STIM5	Stimulus Port Register 5
0xE0000018	STIM6	Stimulus Port Register 6
0xE000001C	STIM7	Stimulus Port Register 7
0xE0000020	STIM8	Stimulus Port Register 8
0xE0000024	STIM9	Stimulus Port Register 9
0xE0000028	STIM10	Stimulus Port Register 10
0xE000002C	STIM11	Stimulus Port Register 11
0xE0000030	STIM12	Stimulus Port Register 12
0xE0000034	STIM13	Stimulus Port Register 13
0xE0000038	STIM14	Stimulus Port Register 14
0xE000003C	STIM15	Stimulus Port Register 15
0xE0000040	STIM16	Stimulus Port Register 16
0xE0000044	STIM17	Stimulus Port Register 17
0xE0000048	STIM18	Stimulus Port Register 18
0xE000004C	STIM19	Stimulus Port Register 19
0xE0000050	STIM20	Stimulus Port Register 20
0xE0000054	STIM21	Stimulus Port Register 21
0xE0000058	STIM22	Stimulus Port Register 22
0xE000005C	STIM23	Stimulus Port Register 23
0xE0000060	STIM24	Stimulus Port Register 24
0xE0000064	STIM25	Stimulus Port Register 25
0xE0000068	STIM26	Stimulus Port Register 26
0xE000006C	STIM27	Stimulus Port Register 27
0xE0000070	STIM28	Stimulus Port Register 28
0xE0000074	STIM29	Stimulus Port Register 29
0xE0000078	STIM30	Stimulus Port Register 30
0xE000007C	STIM31	Stimulus Port Register 31

Table 4-25: ITM Register Map *(Continued)*

Address(es)	Registered Name	Description
0xE0000E00	TER	Trace Enable Register.
0xE0000E40	TPR	Trace Privilege Register.
0xE0000E80	TCR	Trace Control Register.
0xE0000FB0	LOCKAREG	Lock Access Register
0xE0000FB4	LOCKSREG	Lock Status Register
0xE0000FD0	PID4	Peripheral Identification Register 4
0xE0000FD4	PID5	Peripheral Identification Register 5
0xE0000FD8	PID6	Peripheral Identification Register 6
0xE0000FDC	PID7	Peripheral Identification Register 7
0xE0000FE0	PID0	Peripheral Identification Register 0
0xE0000FE4	PID1	Peripheral Identification Register 1
0xE0000FE8	PID2	Peripheral Identification Register 2
0xE0000FEC	PID3	Peripheral Identification Register 3
0xE0000FF0	CID0	Component Identification Register 1
0xE0000FF4	CID1	Component Identification Register 1
0xE0000FF8	CID2	Component Identification Register 2
0xE0000FFC	CID3	Component Identification Register 3

4.8.2 ITM Registers

4.8.2.1 STIM0 Register

Stimulus Port Register 0
OFFSET: 0xE0000000
INSTANCE 0 ADDRESS: 0xE0000000
Stimulus Port Register 0

Table 4-26: STIM0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM0																															

Table 4-27: STIM0 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM0	0x0	RW	Stimulus Port Register 0.

4.8.2.2 STIM1 Register

Stimulus Port Register 1
OFFSET: 0xE0000004
INSTANCE 0 ADDRESS: 0xE0000004
Stimulus Port Register 1

Table 4-28: STIM1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM1																															

Table 4-29: STIM1 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM1	0x0	RW	Stimulus Port Register 1.

4.8.2.3 STIM2 Register

Stimulus Port Register 2
OFFSET: 0xE0000008
INSTANCE 0 ADDRESS: 0xE0000008
Stimulus Port Register 2

Table 4-30: STIM2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM2																															

Table 4-31: STIM2 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM2	0x0	RW	Stimulus Port Register 2.

4.8.2.4 STIM3 Register

Stimulus Port Register 3
OFFSET: 0xE000000C
INSTANCE 0 ADDRESS: 0xE000000C
Stimulus Port Register 3

Table 4-32: STIM3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM3																															

Table 4-33: STIM3 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM3	0x0	RW	Stimulus Port Register 3.

4.8.2.5 STIM4 Register

Stimulus Port Register 4
OFFSET: 0xE0000010
INSTANCE 0 ADDRESS: 0xE0000010
Stimulus Port Register 4

Table 4-34: STIM4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM4																															

Table 4-35: STIM4 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM4	0x0	RW	Stimulus Port Register 4.

4.8.2.6 STIM5 Register

Stimulus Port Register 5
OFFSET: 0xE0000014
INSTANCE 0 ADDRESS: 0xE0000014
Stimulus Port Register 5

Table 4-36: STIM5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM5																															

Table 4-37: STIM5 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM5	0x0	RW	Stimulus Port Register 5.

4.8.2.7 STIM6 Register

Stimulus Port Register 6
OFFSET: 0xE0000018
INSTANCE 0 ADDRESS: 0xE0000018
Stimulus Port Register 6

Table 4-38: STIM6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM6																															

Table 4-39: STIM6 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM6	0x0	RW	Stimulus Port Register 6.

4.8.2.8 STIM7 Register

Stimulus Port Register 7
OFFSET: 0xE000001C
INSTANCE 0 ADDRESS: 0xE000001C
Stimulus Port Register 7

Table 4-40: STIM7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM7																															

Table 4-41: STIM7 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM7	0x0	RW	Stimulus Port Register 7.

4.8.2.9 STIM8 Register

Stimulus Port Register 8
OFFSET: 0xE0000020
INSTANCE 0 ADDRESS: 0xE0000020
Stimulus Port Register 8

Table 4-42: STIM8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM8																															

Table 4-43: STIM8 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM8	0x0	RW	Stimulus Port Register 8.

4.8.2.10 STIM9 Register

Stimulus Port Register 9
OFFSET: 0xE0000024
INSTANCE 0 ADDRESS: 0xE0000024
Stimulus Port Register 9

Table 4-44: STIM9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM9																															

Table 4-45: STIM9 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM9	0x0	RW	Stimulus Port Register 9.

4.8.2.11 STIM10 Register

Stimulus Port Register 10
OFFSET: 0xE0000028
INSTANCE 0 ADDRESS: 0xE0000028
Stimulus Port Register 10

Table 4-46: STIM10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM10																															

Table 4-47: STIM10 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM10	0x0	RW	Stimulus Port Register 10.

4.8.2.12 STIM11 Register

Stimulus Port Register 11
OFFSET: 0xE000002C
INSTANCE 0 ADDRESS: 0xE000002C
Stimulus Port Register 11

Table 4-48: STIM11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM11																															

Table 4-49: STIM11 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM11	0x0	RW	Stimulus Port Register 11.

4.8.2.13 STIM12 Register

Stimulus Port Register 12
OFFSET: 0xE0000030
INSTANCE 0 ADDRESS: 0xE0000030
Stimulus Port Register 12

Table 4-50: STIM12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM12																															

Table 4-51: STIM12 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM12	0x0	RW	Stimulus Port Register 12.

4.8.2.14 STIM13 Register

Stimulus Port Register 13
OFFSET: 0xE0000034
INSTANCE 0 ADDRESS: 0xE0000034
Stimulus Port Register 13

Table 4-52: STIM13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM13																															

Table 4-53: STIM13 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM13	0x0	RW	Stimulus Port Register 13.

4.8.2.15 STIM14 Register

Stimulus Port Register 14
OFFSET: 0xE0000038
INSTANCE 0 ADDRESS: 0xE0000038
Stimulus Port Register 14

Table 4-54: STIM14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM14																															

Table 4-55: STIM14 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM14	0x0	RW	Stimulus Port Register 14.

4.8.2.16 STIM15 Register

Stimulus Port Register 15
OFFSET: 0xE000003C
INSTANCE 0 ADDRESS: 0xE000003C
Stimulus Port Register 15

Table 4-56: STIM15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM15																															

Table 4-57: STIM15 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM15	0x0	RW	Stimulus Port Register 15.

4.8.2.17 STIM16 Register

Stimulus Port Register 16
OFFSET: 0xE0000040
INSTANCE 0 ADDRESS: 0xE0000040
Stimulus Port Register 16

Table 4-58: STIM16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM16																															

Table 4-59: STIM16 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM16	0x0	RW	Stimulus Port Register 16.

4.8.2.18 STIM17 Register

Stimulus Port Register 17
OFFSET: 0xE0000044
INSTANCE 0 ADDRESS: 0xE0000044
Stimulus Port Register 17

Table 4-60: STIM17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM17																															

Table 4-61: STIM17 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM17	0x0	RW	Stimulus Port Register 17.

4.8.2.19 STIM18 Register

Stimulus Port Register 18
OFFSET: 0xE0000048
INSTANCE 0 ADDRESS: 0xE0000048
Stimulus Port Register 18

Table 4-62: STIM18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM18																															

Table 4-63: STIM18 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM18	0x0	RW	Stimulus Port Register 18.

4.8.2.20 STIM19 Register

Stimulus Port Register 19
OFFSET: 0xE000004C
INSTANCE 0 ADDRESS: 0xE000004C
Stimulus Port Register 19

Table 4-64: STIM19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM19																															

Table 4-65: STIM19 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM19	0x0	RW	Stimulus Port Register 19.

4.8.2.21 STIM20 Register

Stimulus Port Register 20
OFFSET: 0xE0000050
INSTANCE 0 ADDRESS: 0xE0000050
Stimulus Port Register 20

Table 4-66: STIM20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM20																															

Table 4-67: STIM20 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM20	0x0	RW	Stimulus Port Register 20.

4.8.2.22 STIM21 Register

Stimulus Port Register 21
OFFSET: 0xE0000054
INSTANCE 0 ADDRESS: 0xE0000054
Stimulus Port Register 21

Table 4-68: STIM21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM21																															

Table 4-69: STIM21 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM21	0x0	RW	Stimulus Port Register 21.

4.8.2.23 STIM22 Register

Stimulus Port Register 22
OFFSET: 0xE0000058
INSTANCE 0 ADDRESS: 0xE0000058
Stimulus Port Register 22

Table 4-70: STIM22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM22																															

Table 4-71: STIM22 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM22	0x0	RW	Stimulus Port Register 22.

4.8.2.24 STIM23 Register

Stimulus Port Register 23
OFFSET: 0xE000005C
INSTANCE 0 ADDRESS: 0xE000005C
Stimulus Port Register 23

Table 4-72: STIM23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM23																															

Table 4-73: STIM23 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM23	0x0	RW	Stimulus Port Register 23.

4.8.2.25 STIM24 Register

Stimulus Port Register 24
OFFSET: 0xE0000060
INSTANCE 0 ADDRESS: 0xE0000060
Stimulus Port Register 24

Table 4-74: STIM24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM24																															

Table 4-75: STIM24 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM24	0x0	RW	Stimulus Port Register 24.

4.8.2.26 STIM25 Register

Stimulus Port Register 25
OFFSET: 0xE0000064
INSTANCE 0 ADDRESS: 0xE0000064
Stimulus Port Register 25

Table 4-76: STIM25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM25																															

Table 4-77: STIM25 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM25	0x0	RW	Stimulus Port Register 25.

4.8.2.27 STIM26 Register

Stimulus Port Register 26
OFFSET: 0xE0000068
INSTANCE 0 ADDRESS: 0xE0000068
Stimulus Port Register 26

Table 4-78: STIM26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM26																															

Table 4-79: STIM26 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM26	0x0	RW	Stimulus Port Register 26.

4.8.2.28 STIM27 Register

Stimulus Port Register 27
OFFSET: 0xE000006C
INSTANCE 0 ADDRESS: 0xE000006C
Stimulus Port Register 27

Table 4-80: STIM27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM27																															

Table 4-81: STIM27 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM27	0x0	RW	Stimulus Port Register 27.

4.8.2.29 STIM28 Register

Stimulus Port Register 28
OFFSET: 0xE0000070
INSTANCE 0 ADDRESS: 0xE0000070
Stimulus Port Register 28

Table 4-82: STIM28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM28																															

Table 4-83: STIM28 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM28	0x0	RW	Stimulus Port Register 28.

4.8.2.30 STIM29 Register

Stimulus Port Register 29
OFFSET: 0xE0000074
INSTANCE 0 ADDRESS: 0xE0000074
Stimulus Port Register 29

Table 4-84: STIM29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM29																															

Table 4-85: STIM29 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM29	0x0	RW	Stimulus Port Register 29.

4.8.2.31 STIM30 Register

Stimulus Port Register 30
OFFSET: 0xE0000078
INSTANCE 0 ADDRESS: 0xE0000078
Stimulus Port Register 30

Table 4-86: STIM30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM30																															

Table 4-87: STIM30 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM30	0x0	RW	Stimulus Port Register 30.

4.8.2.32 STIM31 Register

Stimulus Port Register 31
OFFSET: 0xE000007C
INSTANCE 0 ADDRESS: 0xE000007C
Stimulus Port Register 31

Table 4-88: STIM31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIM31																															

Table 4-89: STIM31 Register Bits

Bit	Name	Reset	RW	Description
31:0	STIM31	0x0	RW	Stimulus Port Register 31.

4.8.2.33 TER Register

Trace Enable Register
OFFSET: 0xE0000E00
INSTANCE 0 ADDRESS: 0xE0000E00
Trace Enable Register

Table 4-90: TER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STIMENA																															

Table 4-91: TER Register Bits

Bit	Name	Reset	RW	Description
31:0	STIMENA	0x0	RW	Bit mask to enable tracing on ITM stimulus ports. One bit per stimulus port.

4.8.2.34 TPR Register

Trace Privilege Register
OFFSET: 0xE0000E40
INSTANCE 0 ADDRESS: 0xE0000E40
Trace Privilege Register

Table 4-92: TPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												PRIVMASK			

Table 4-93: TPR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RW	RESERVED
3:0	PRIVMASK	0x0	RW	Bit mask to enable tracing on ITM stimulus ports. bit[0] = stimulus ports[7:0], bit[1] = stimulus ports[15:8], bit[2] = stimulus ports[23:16], bit[3] = stimulus ports[31:24].

4.8.2.35 TCR Register

Trace Control Register
OFFSET: 0xE0000E80
INSTANCE 0 ADDRESS: 0xE0000E80
Trace Control Register

Table 4-94: TCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD								BUSY	ATB_ID							RSVD				TS_FREQ		TS_PRESCALE		RSVD			SWV_ENABLE	DWT_ENABLE	SYNC_ENABLE	TS_ENABLE	ITM_ENABLE

Table 4-95: TCR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23	BUSY	0x0	RW	Set when ITM events present and being drained.
22:16	ATB_ID	0x0	RW	ATB ID for CoreSight system.
15:12	RSVD	0x0	RO	RESERVED
11:10	TS_FREQ	0x0	RW	Global Timestamp Frequency.
9:8	TS_PRESCALE	0x0	RW	Timestamp prescaler: 0b00 = no prescaling 0b01 = divide by 4 0b10 = divide by 16 0b11 = divide by 64.
7:5	RSVD	0x0	RO	RESERVED

Table 4-95: TCR Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
4	SWV_ENABLE	0x0	RW	Enable SWV behavior - count on TPIUEMIT and TPIU-BAUD.
3	DWT_ENABLE	0x0	RW	Enables the DWT stimulus.
2	SYNC_ENABLE	0x0	RW	Enables sync packets for TPIU.
1	TS_ENABLE	0x0	RW	Enables differential timestamps. Differential time-stamps are emitted when a packet is written to the FIFO with a non-zero timestamp counter, and when the timestamp counter overflows. Timestamps are emitted during idle times after a fixed number of cycles. This provides a time reference for packets and inter-packet gaps.
0	ITM_ENABLE	0x0	RW	Enable ITM. This is the master enable, and must be set before ITM Stimulus and Trace Enable registers can be written.

4.8.2.36 LOCKAREG Register

Lock Access Register
OFFSET: 0xE0000FB0
INSTANCE 0 ADDRESS: 0xE0000FB0
Lock Access Register

Table 4-96: LOCKAREG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LOCKAREG																															

Table 4-97: LOCKAREG Register Bits

Bit	Name	Reset	RW	Description
31:0	LOCKAREG	0x0	RW	Key register value. Key = 0xC5ACCE55 - Key

4.8.2.37 LOCKSREG Register

Lock Status Register
OFFSET: 0xE0000FB4
INSTANCE 0 ADDRESS: 0xE0000FB4
Lock Status Register

Table 4-98: LOCKSREG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													BYTEACC	ACCESS	PRESENT

Table 4-99: LOCKSREG Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	BYTEACC	0x0	RO	Cannot implement 8-bit lock accesses.
1	ACCESS	0x0	RO	Write access to component is blocked. All writes are ignored, reads are permitted.
0	PRESENT	0x1	RO	Indicates that a lock mechanism exists for this component.

4.8.2.38 PID4 Register

Peripheral Identification Register 4
OFFSET: 0xE0000FD0
INSTANCE 0 ADDRESS: 0xE0000FD0
Peripheral Identification Register 4

Table 4-100: PID4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID4																															

Table 4-101: PID4 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID4	0x4	R0	Peripheral Identification 4.

4.8.2.39 PID5 Register

Peripheral Identification Register 5
OFFSET: 0xE0000FD4
INSTANCE 0 ADDRESS: 0xE0000FD4
Peripheral Identification Register 5

Table 4-102: PID5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID5																															

Table 4-103: PID5 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID5	0x0	R0	Peripheral Identification 5.

4.8.2.40 PID6 Register

Peripheral Identification Register 6
OFFSET: 0xE0000FD8
INSTANCE 0 ADDRESS: 0xE0000FD8
Peripheral Identification Register 6

Table 4-104: PID6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID6																															

Table 4-105: PID6 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID6	0x0	R0	Peripheral Identification 6.

4.8.2.41 PID7 Register

Peripheral Identification Register 7
OFFSET: 0xE0000FDC
INSTANCE 0 ADDRESS: 0xE0000FDC
Peripheral Identification Register 7

Table 4-106: PID7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID7																															

Table 4-107: PID7 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID7	0x0	R0	Peripheral Identification 7.

4.8.2.42 PID0 Register

Peripheral Identification Register 0
OFFSET: 0xE0000FE0
INSTANCE 0 ADDRESS: 0xE0000FE0
Peripheral Identification Register 0

Table 4-108: PID7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID0																															

Table 4-109: PID0 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID0	0x1	R0	Peripheral Identification 0.

4.8.2.43 PID1 Register

Peripheral Identification Register 1
OFFSET: 0xE0000FE4
INSTANCE 0 ADDRESS: 0xE0000FE4
Peripheral Identification Register 1

Table 4-110: PID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID1																															

Table 4-111: PID1 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID1	0xb0	R0	Peripheral Identification 1.

4.8.2.44 PID2 Register

Peripheral Identification Register 2
OFFSET: 0xE0000FE8
INSTANCE 0 ADDRESS: 0xE0000FE8
Peripheral Identification Register 2

Table 4-112: PID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID2																															

Table 4-113: PID2 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID2	0x3b	R0	Peripheral Identification 2.

4.8.2.45 PID3 Register

Peripheral Identification Register 3
OFFSET: 0xE0000FEC
INSTANCE 0 ADDRESS: 0xE0000FEC
Peripheral Identification Register 3

Table 4-114: PID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PID3																															

Table 4-115: PID3 Register Bits

Bit	Name	Reset	RW	Description
31:0	PID3	0x0	R0	Peripheral Identification 3.

4.8.2.46 CID0 Register

Component Identification Register 1
OFFSET: 0xE0000FF0
INSTANCE 0 ADDRESS: 0xE0000FF0
Component Identification Register 1

Table 4-116: CID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CID0																															

Table 4-117: CID0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CID0	0xd	R0	Component Identification 1.

4.8.2.47 CID1 Register

Component Identification Register 1
OFFSET: 0xE0000FF4
INSTANCE 0 ADDRESS: 0xE0000FF4
Component Identification Register 1

Table 4-118: CID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CID1																															

Table 4-119: CID1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CID1	0xe0	R0	Component Identification 1.

4.8.2.48 CID2 Register

Component Identification Register 2
OFFSET: 0xE0000FF8
INSTANCE 0 ADDRESS: 0xE0000FF8
Component Identification Register 2

Table 4-120: CID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CID2																															

Table 4-121: CID2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CID2	0x5	R0	Component Identification 2.

4.8.2.49 CID3 Register

Component Identification Register 3
OFFSET: 0xE0000FFC
INSTANCE 0 ADDRESS: 0xE0000FFC
Component Identification Register 3

Table 4-122: CID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CID3																															

Table 4-123: CID3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CID3	0xb1	R0	Component Identification 3.

4.9 MCUCTRL Registers

MCU Miscellaneous Control Logic
INSTANCE 0 BASE ADDRESS:0x40020000

4.9.1 Register Memory Map

Table 4-124: MCUCTRL Register Map

Address(es)	Register Name	Description
0x40020000	CHIP_INFO	Chip Information Register
0x40020004	CHIPID0	Unique Chip ID 0
0x40020008	CHIPID1	Unique Chip ID 1
0x4002000C	CHIPREV	Chip Revision
0x40020010	VENDORID	Unique Vendor ID
0x40020060	BUCK	Analog Buck Control
0x40020064	BUCK2	Buck Control Reg2
0x40020068	BUCK3	Buck control reg 3
0x40020084	LDOREG2	LDO Control Register 2
0x40020100	BODPORCTRL	BOD and PDR control Register
0x4002010C	ADCCAL	ADC Calibration Control
0x40020110	ADCBATTLOAD	ADC Battery Load Enable
0x4002011C	ADCREFCOMP	ADC Reference Keeper and Comparator Control
0x40020120	XTALCTRL	XTAL Oscillator Control
0x40020124	XTALGENCTRL	XTAL Oscillator General Control
0x400201A0	BOOTLOADERLOW	Determines whether the bootloader code is visible at address 0x00000000
0x400201A4	SHADOWVALID	Register to indicate whether the shadow registers have been successfully loaded from the Flash Information Space.
0x400201C0	ICODEFAULTADDR	ICODE bus address which was present when a bus fault occurred.
0x400201C4	DCODEFAULTADDR	DCODE bus address which was present when a bus fault occurred.
0x400201C8	SYSFAULTADDR	System bus address which was present when a bus fault occurred.
0x400201CC	FAULTSTATUS	Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register.
0x400201D0	FAULTCAPTUREEN	Enable the fault capture registers
0x40020220	PMUENABLE	Control bit to enable/disable the PMU
0x40020250	TPIUCTRL	TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface.

4.9.2 MCUCTRL Registers

4.9.2.1 CHIP_INFO Register

Chip Information Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40020000
Chip Information Register

Table 4-125: CHIP_INFO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PARTNUM																															

Table 4-126: CHIP_INFO Register Bits

Bit	Name	Reset	RW	Description
31:0	PARTNUM	0x3000000	R0	BCD part number. PARTNUM= 0x3000000 - Apollo2 part number is 0x03XXXXXX.

4.9.2.2 CHIPID0 Register

Unique Chip ID 0
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40020004
Unique Chip ID 0

Table 4-127: CHIPID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 4-128: CHIPID0 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Unique chip ID 0. VALUE = Upper 32-bits of a 64-bit encoded unique chip ID

4.9.2.3 CHIPID1 Register

Unique Chip ID 1
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40020008
Unique Chip ID 1

Table 4-129: CHIPID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 4-130: CHIPID1 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Unique chip ID 1. VALUE = Lower 32-bits of a 64-bit encoded unique chip ID

4.9.2.4 CHIPREV Register

Chip Revision
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4002000C
Chip Revision

Table 4-131: CHIPREV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								REVMAJ				REVMIN			

Table 4-132: CHIPREV Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:4	REVMAJ	0x2	RO	Major Revision ID. B = 0x2 - Apollo2 revision B A = 0x1 - Apollo2 revision A
3:0	REVMIN	0x2	RO	Minor Revision ID. REV2 = 0x2 - Apollo2 minor revision 2 REV0 = 0x0 - Apollo2 minor revision 0

4.9.2.5 VENDORID Register

Unique Vendor ID
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40020010
Unique Vendor ID

Table 4-133: VENDORID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 4-134: VENDORID Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Unique Vendor ID AMBIQ = 0x414D4251 - Ambiq Vendor ID

4.9.2.6 BUCK Register

Analog Buck Control
OFFSET: 0x00000060
INSTANCE 0 ADDRESS: 0x40020060
Analog Buck Control

Table 4-135: BUCK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								MEMBUCKRST	COREBUCKRST	BYPBUCKMEM	MEMBUCKPWD	SLEEPBUCK	COREBUCKPWD	BYPBUCKCORE	BUCKSWE

Table 4-136: BUCK Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	MEMBUCKRST	0x0	RW	Reset control override for Mem Buck; 0=enabled, 1=reset; Value is propagated only when the BUCKSWE bit is active, otherwise control is from the power control module.
6	COREBUCKRST	0x0	RW	Reset control override for Core Buck; 0=enabled, 1=reset; Value is propagated only when the BUCKSWE bit is active, otherwise control is from the power control module.
5	BYPBUCKMEM	0x0	RW	Not used. Additional control of buck is available in the power control module.

Table 4-136: BUCK Register Bits

Bit	Name	Reset	RW	Description
4	MEMBUCKPWD	0x0	RW	Memory buck power down override. 1=Powered Down; 0=Enabled; Value is propagated only when the BUCKSWE bit is active, otherwise control is from the power control module. EN = 0x0 - Memory Buck Enable.
3	SLEEPBUCKANA	0x0	RW	HFRC clkgen bit 0 override. When set, this will override to 0 bit 0 of the hfrc_freq_clkgen internal bus
2	COREBUCKPWD	0x0	RW	Core buck power down override. 1=Powered Down; 0=Enabled; Value is propagated only when the BUCKSWE bit is active, otherwise control is from the power control module. EN = 0x0 - Core Buck enable.
1	BYPBUCKCORE	0x0	RW	Not used. Additional control of buck is available in the power control module.
0	BUCKSWE	0x0	RW	Buck Register Software Override Enable. This will enable the override values for MEMBUCKPWD, COREBUCKPWD, COREBUCKRST, MEMBUCKRST, all to be propagated to the control logic, instead of the normal power control module signal. Note - Must take care to have correct value for ALL the register bits when this SWE is enabled. OVERRIDE_DIS = 0x0 - BUCK Software Override Disable. OVERRIDE_EN = 0x1 - BUCK Software Override Enable.

4.9.2.7 BUCK2 Register

Buck Control Reg2
OFFSET: 0x00000064
INSTANCE 0 ADDRESS: 0x40020064
Buck Control Reg2

Table 4-137: BUCK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																				RSVD		HYSTBUCKCORE	HYSTBUCKMEM	RSVD				RSVD			

Table 4-138: BUCK2 Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	RESERVED
11:10	RSVD	0x0	RW	RESERVED
9	HYSTBUCKCORE	0x0	RW	Enable/disable hysteresis on core buck converters internal comparators. DIS = 0x0 - Disable hysteresis on core buck converters internal comparators. EN = 0x1 - Enable hysteresis on core buck converters internal comparators.
8	HYSTBUCKMEM	0x0	RW	Enable/disable hysteresis on memory buck converters internal comparators. DIS = 0x0 - Disable hysteresis on memory buck converters internal comparators. EN = 0x1 - Enable hysteresis on memory buck converters internal comparators.
7:4	RSVD	0x0	RW	RESERVED
3:0	RSVD	0x0	RW	RESERVED

4.9.2.8 BUCK3 Register

Buck control reg 3
OFFSET: 0x00000068
INSTANCE 0 ADDRESS: 0x40020068
Buck control reg 3

Table 4-139: BUCK3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD										RSVD				MEMBUCKBURSTEN	RSVD				RSVD						COREBUCKBURSTEN	COREBUCKZXTRIM					RSVD	

Table 4-140: BUCK3 Register Bits

Bit	Name	Reset	RW	Description
31:22	RSVD	0x0	RO	RESERVED
21:18	RSVD	0x0	RW	RESERVED
17	MEMBUCKBURSTEN	0x0	RW	MEM Buck burst enable 0=disable, 0=disabled, 1=enable.
16:13	RSVD	0x0	RW	RESERVED
12:11	RSVD	0x0	RW	RESERVED
10:7	RSVD	0x0	RW	RESERVED
6	COREBUCKBURSTEN	0x0	RW	Core Buck burst enable. 0=disabled, 1=enabled
5:2	COREBUCKZXTRIM	0x0	RW	Core buck zero crossing trim value
1:0	RSVD	0x0	RW	RESERVED

4.9.2.9 LDOREG2 Register

LDO Control Register 2
OFFSET: 0x00000084
INSTANCE 0 ADDRESS: 0x40020084
LDO Control Register 2

Table 4-141: LDOREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD									CORELDOVDDLEN	RAMLDOLPMODE	PWDRAMLDO	PWDANALDO	PWDMEMLDO	PWDCORELDO	SLEEPANALDO	SLEEPMEMLDO	SLEEPCORELDO	VREFSELANALDO	VREFSELSRAMLDO	VREFSELFLASHLDO	VREFSELCORELDO	RSVD					RSVD					LDO2SWE

Table 4-142: LDOREG2 Register Bits

Bit	Name	Reset	RW	Description
31:23	RSVD	0x0	RW	Reserved
22	CORELDOVDDLEN	0x0	RW	Core LDO output enable. 0=Hi-Z, 1=enable. This value is propagated only when LDO2SWE bit is active(1).

Table 4-142: LDOREG2 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
21	RAMLDOLPMODE	0x0	RW	RAM LDO LP Mode. 0=normal mode, 1=low power mode; This value is propagated only when LDO2SWE bit is active(1). LPMODE = 0x1 - RAM IN LP mode
20	PWDRAMLDO	0x0	RW	RAM LDO Power Down. 0=powered up, 1=powered down ; This value is propagated only when LDO2SWE bit is active(1). PWR_DN = 0x1 - Power down RAM LDO.
19	PWDANALDO	0x0	RW	Analog LDO Power Down. This value is propagated only when LDO2SWE bit is active(1). PWR_DN = 0x1 - Power down Analog LDO.
18	PWDMEMLDO	0x0	RW	MEM LDO Power Down. This value is propagated only when LDO2SWE bit is active(1). PWR_DN = 0x1 - Power down Flash LDO.
17	PWDCORELDO	0x0	RW	CORE LDO Power Down. This value is propagated only when LDO2SWE bit is active(1). PWR_DN = 0x1 - Power down Core LDO.
16	SLEEPANALDO	0x0	RW	Analog LDO Sleep. This value is propagated only when LDO2SWE bit is active(1). SLEEP = 0x1 - Analog LDO sleep.
15	SLEEPMEMLDO	0x0	RW	FLASH LDO Sleep. This value is propagated only when LDO2SWE bit is active(1). SLEEP = 0x1 - SRAM LDO sleep.
14	SLEEPCORELDO	0x0	RW	CORE LDO Sleep. This value is propagated only when LDO2SWE bit is active(1). SLEEP = 0x1 - Core LDO sleep.
13	VREFSELANALDO	0x0	RW	CONTROL BIT IS NOT USED. PLEASE TREAT AS RESERVED
12	VREFSELSRAMLDO	0x0	RW	CONTROL BIT IS NOT USED. PLEASE TREAT AS RESERVED
11	VREFSELFLASHLDO	0x0	RW	CONTROL BIT IS NOT USED. PLEASE TREAT AS RESERVED
10	VREFSELCORELDO	0x0	RW	CONTROL BIT IS NOT USED. PLEASE TREAT AS RESERVED
9:6	RSVD	0x0	RW	RESERVED
5:1	RSVD	0x0	RW	RESERVED
0	LDO2SWE	0x0	RW	LDO2 Software Override Enable. If enabled (=1), this will enable the override values from this register to be used instead of the normal control signals for the following fields: CORELDOVDDLEN, RAMLDOLPMODE, PWDRAM-LDO, PWDANALDO, PWDMEMLDO, PWDCORELDO, SLEEPANALDO, SLEEPMEMLDO, SLEEPCORELDO. OVERRIDE_DIS = 0x0 - LDO2 Software Override Disable. OVERRIDE_EN = 0x1 - LDO2 Software Override Enable.

4.9.2.10 BODPORCTRL Register

BOD and PDR control Register
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x40020100
BOD and PDR control Register

Table 4-143: BODPORCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												BODEXTREFSEL	PDREXTREFSEL	PWDBOD	PWDPDR

Table 4-144: BODPORCTRL Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	BODEXTREFSEL	0x0	RW	BOD External Reference Select. SELECT = 0x1 - BOD external reference select.
2	PDREXTREFSEL	0x0	RW	PDR External Reference Select. SELECT = 0x1 - PDR external reference select.
1	PWDBOD	0x0	RW	BOD Power Down. PWR_DN = 0x1 - BOD power down.
0	PWDPDR	0x0	RW	PDR Power Down. PWR_DN = 0x1 - PDR power down.

4.9.2.11 ADCCAL Register

ADC Calibration Control
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x4002010C
ADC Calibration Control

Table 4-145: ADCCAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														ADCCALIBRATED	CALONPWRUP

Table 4-146: ADCCAL Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	ADCCALIBRATED	0x0	RO	Status for ADC Calibration FALSE = 0x0 - ADC is not calibrated TRUE = 0x1 - ADC is calibrated
0	CALONPWRUP	0x1	RW	Run ADC Calibration on initial power up sequence DIS = 0x0 - Disable automatic calibration on initial power up EN = 0x1 - Enable automatic calibration on initial power up

4.9.2.12 ADCBATTLOAD Register

ADC Battery Load Enable
OFFSET: 0x00000110
INSTANCE 0 ADDRESS: 0x40020110
ADC Battery Load Enable

Table 4-147: ADCBATTLOAD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															BATTLOAD

Table 4-148: ADCBATTLOAD Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	BATTLOAD	0x0	RW	Enable the ADC battery load resistor DIS = 0x0 - Battery load is disconnected EN = 0x1 - Battery load is enabled

4.9.2.13 ADCREFCOMP Register

ADC Reference Keeper and Comparator Control
OFFSET: 0x0000011C
INSTANCE 0 ADDRESS: 0x4002011C
ADC Reference Keeper and Comparator Control

Table 4-149: ADCREFCOMP Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17	16	15 14 13	12 11 10 09 08	07 06 05 04 03 02 01	00
RSVD	ADCRFCMPEN	RSVD	RSVD	RSVD	ADC_REFCOMP_OUT

Table 4-150: ADCREFCOMP Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	RESERVED
16	ADCRFCMPEN	0x0	RW	ADC Reference comparator power down
15:13	RSVD	0x0	RO	RESERVED
12:8	RSVD	0x0	RW	RESERVED
7:1	RSVD	0x0	RO	RESERVED
0	ADC_REFCOMP_OUT	0x0	RO	Output of the ADC reference comparator

4.9.2.14 XTALCTRL Register

XTAL Oscillator Control
OFFSET: 0x00000120
INSTANCE 0 ADDRESS: 0x40020120
XTAL Oscillator Control

Table 4-151: XTALCTRL Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10	09	08	07 06	05	04	03	02	01	00
RSVD	RSVD	RSVD		PWDBODXTAL	PWDCMPRXTAL	PWDCOREXTAL	BYPCMPRXTAL	FDBKDSBLXTAL	XTALSWE

Table 4-152: XTALCTRL Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9:8	RSVD	0x0	RW	RESERVED
7:6	RSVD	0x0	RW	RESERVED
5	PWDBODXTAL	0x0	RW	XTAL Power down on brown out. PWRUPBOD = 0x0 - Power up xtal on BOD PWRDNBOD = 0x1 - Power down XTAL on BOD

Table 4-152: XTALCTRL Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
4	PWDCMPRXTAL	0x0	RW	XTAL Oscillator Power Down Comparator override. Value is only propagated when XTALSWE = 1, otherwise, the power down control is sourced from the normal clk_gen and init logic. PWRUPCOMP = 0x0 - Power up XTAL oscillator comparator. PWRDNCOMP = 0x1 - Power down XTAL oscillat:or comparator.
3	PWDCOREXTAL	0x0	RW	XTAL Oscillator Power Down Core override. Value is only propagated when XTALSWE = 1, otherwise, the power down control is sourced from the normal clk_gen and init logic. PWRUPCORE = 0x0 - Power up XTAL oscillator core. PWRDNCORE = 0x1 - Power down XTAL oscillator core.
2	BYPCMPRXTAL	0x0	RW	XTAL Oscillator Bypass Comparator. USECOMP = 0x0 - Use the XTAL oscillator comparator. BYPCOMP = 0x1 - Bypass the XTAL oscillator comparator.
1	FDBKDSBLXTAL	0x0	RW	XTAL Oscillator Disable Feedback. EN = 0x0 - Enable XTAL oscillator comparator. DIS = 0x1 - Disable XTAL oscillator comparator.
0	XTALSWE	0x0	RW	XTAL Software Override Enable. Enabling this will allow the fields of PWDCMPRXTAL and PWDCOREXTAL to be actively sourced to the analog module, instead of the normal logic. OVERRIDE_DIS = 0x0 - XTAL Software Override Disable. OVERRIDE_EN = 0x1 - XTAL Software Override Enable.

4.9.2.15 XTALGENCTRL Register

XTAL Oscillator General Control
OFFSET: 0x00000124
INSTANCE 0 ADDRESS: 0x40020124
XTAL Oscillator General Control

Table 4-153: XTALGENCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																		RSVD						RSVD							ACWARMUP

Table 4-154: XTALGENCTRL Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	RESERVED
13:8	RSVD	0x0	RW	RESERVED
7:2	RSVD	0x0	RW	RESERVED
1:0	ACWARMUP	0x0	RW	Auto-calibration delay control 1SEC = 0x0 - Warmup period of 1-2 seconds 2SEC = 0x1 - Warmup period of 2-4 seconds 4SEC = 0x2 - Warmup period of 4-8 seconds 8SEC = 0x3 - Warmup period of 8-16 seconds

4.9.2.16 BOOTLOADERLOW Register

Determines whether the bootloader code is visible at address 0x00000000
OFFSET: 0x000001A0
INSTANCE 0 ADDRESS: 0x400201A0
Determines whether the bootloader code is visible at address 0x00000000

Table 4-155: BOOTLOADERLOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															VALUE

Table 4-156: BOOTLOADERLOW Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	VALUE	0x1	RW	Determines whether the bootloader code is visible at address 0x00000000 or not. ADDR0 = 0x1 - Bootloader code at 0x00000000.

4.9.2.17 SHADOWVALID Register

Register to indicate whether the shadow registers have been successfully loaded from the Flash Information Space.
OFFSET: 0x000001A4
INSTANCE 0 ADDRESS: 0x400201A4
Register to indicate whether the shadow registers have been successfully loaded from the Flash Information Space.

Table 4-157: SHADOWVALID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														BL_DSLEEP	VALID

Table 4-158: SHADOWVALID Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	BL_DSLEEP	0x1	RO	Indicates whether the bootloader should sleep or deep sleep if no image loaded. DEEPSLEEP = 0x1 - Bootloader will go to deep sleep if no flash image loaded
0	VALID	0x1	RO	Indicates whether the shadow registers contain valid data from the Flash Information Space. VALID = 0x1 - Flash information space contains valid data.

4.9.2.18 ICODEFAULTADDR Register

ICODE bus address which was present when a bus fault occurred.
OFFSET: 0x000001C0
INSTANCE 0 ADDRESS: 0x400201C0
ICODE bus address which was present when a bus fault occurred.

Table 4-159: ICODEFAULTADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ADDR																															

Table 4-160: ICODEFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	ADDR	0x0	RO	The ICODE bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register.

4.9.2.19 DCODEFAULTADDR Register

DCODE bus address which was present when a bus fault occurred.
OFFSET: 0x000001C4
INSTANCE 0 ADDRESS: 0x400201C4
DCODE bus address which was present when a bus fault occurred.

Table 4-161: DCODEFAULTADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ADDR																															

Table 4-162: DCODEFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	ADDR	0x0	RO	The DCODE bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register.

4.9.2.20 SYSFAULTADDR Register

System bus address which was present when a bus fault occurred.
OFFSET: 0x000001C8
INSTANCE 0 ADDRESS: 0x400201C8
System bus address which was present when a bus fault occurred.

Table 4-163: SYSFAULTADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ADDR																															

Table 4-164: SYSFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	ADDR	0x0	RO	SYS bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register.

4.9.2.21 FAULTSTATUS Register

Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register.
OFFSET: 0x000001CC
INSTANCE 0 ADDRESS: 0x400201CC
Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register.

Table 4-165: FAULTSTATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													SYS	DCODE	ICODE

Table 4-166: FAULTSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	SYS	0x0	RW	SYS Bus Decoder Fault Detected bit. When set, a fault has been detected, and the SYSFAULTADDR register will contain the bus address which generated the fault. NOFAULT = 0x0 - No bus fault has been detected. FAULT = 0x1 - Bus fault detected.
1	DCODE	0x0	RW	DCODE Bus Decoder Fault Detected bit. When set, a fault has been detected, and the DCODEFAULTADDR register will contain the bus address which generated the fault. NOFAULT = 0x0 - No DCODE fault has been detected. FAULT = 0x1 - DCODE fault detected.
0	ICODE	0x0	RW	The ICODE Bus Decoder Fault Detected bit. When set, a fault has been detected, and the ICODEFAULTADDR register will contain the bus address which generated the fault. NOFAULT = 0x0 - No ICODE fault has been detected. FAULT = 0x1 - ICODE fault detected.

4.9.2.22 FAULTCAPTUREEN Register

Enable the fault capture registers
OFFSET: 0x000001D0
INSTANCE 0 ADDRESS: 0x400201D0
Enable the fault capture registers

Table 4-167: FAULTCAPTUREEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															ENABLE

Table 4-168: FAULTCAPTUREEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	ENABLE	0x0	RW	Fault Capture Enable field. When set, the Fault Capture monitors are enabled and addresses which generate a hard fault are captured into the FAULTADDR registers. DIS = 0x0 - Disable fault capture. EN = 0x1 - Enable fault capture.

4.9.2.23 PMUENABLE Register

Control bit to enable/disable the PMU
OFFSET: 0x00000220
INSTANCE 0 ADDRESS: 0x40020220
Control bit to enable/disable the PMU

Table 4-169: PMUENABLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															ENABLE

Table 4-170: PMUENABLE Register Bits

Bit	Name	Reset	RW	Description
31:0	RSVD	0x0	RO	RESERVED
0	ENABLE	0x1	RW	PMU Enable Control bit. When set, the SoC's PMU will place the SoC into the lowest power consuming Deep Sleep mode upon execution of a WFI instruction (dependent on the setting of the SLEEPDEEP bit in the Arm SCR register). When cleared, regardless of the requested sleep mode, the PMU will not enter the lowest power Deep Sleep mode, instead entering the Sleep mode. DIS = 0x0 - Disable SoC power management. EN = 0x1 - Enable SoC power management.

4.9.2.24 TPIUCTRL Register

TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface.
OFFSET: 0x00000250
INSTANCE 0 ADDRESS: 0x40020250
TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface.

Table 4-171: TPIUCTRL Register

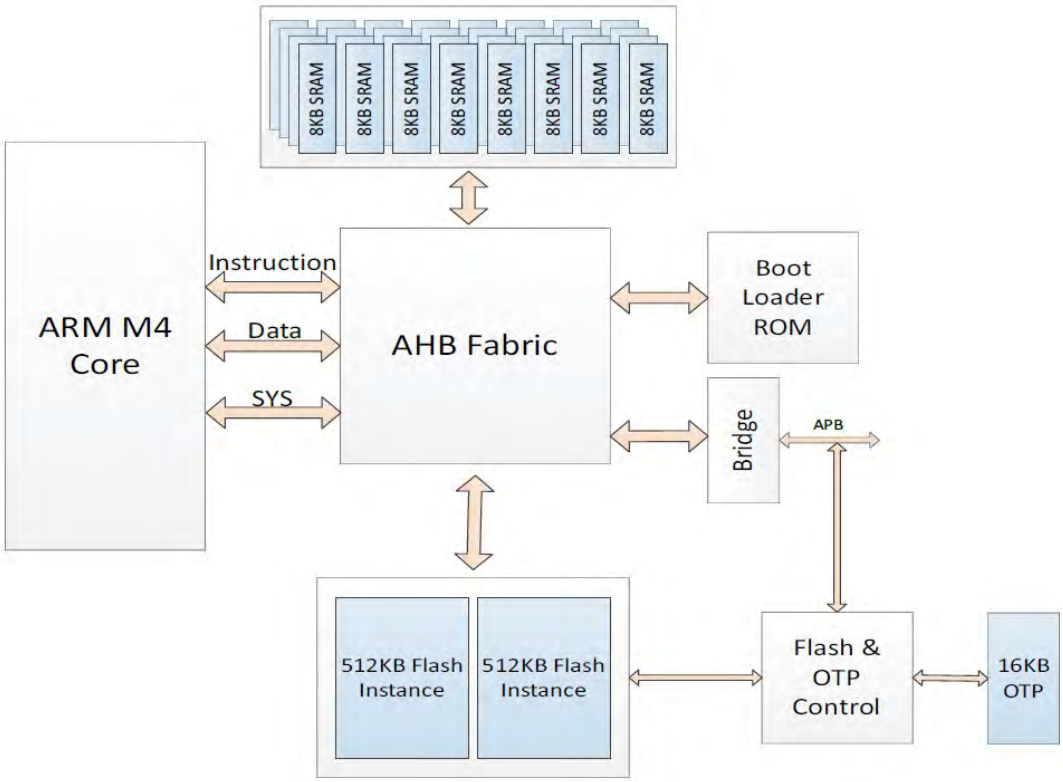
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					CLKSEL			RSVD							ENABLE

Table 4-172: TPIUCTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:8	CLKSEL	0x0	RW	This field selects the frequency of the ARM M4 TPIU port. LOW_PWR = 0x0 - Low power state. 0MHz = 0x0 - Low power state. HFRC_DIV_2 = 0x1 - Selects HFRC divided by 2 as the source TPIU clk HFRC_DIV_8 = 0x2 - Selects HFRC divided by 8 as the source TPIU clk HFRC_DIV_16 = 0x3 - Selects HFRC divided by 16 as the source TPIU clk HFRC_DIV_32 = 0x4 - Selects HFRC divided by 32 as the source TPIU clk
7:1	RSVD	0x0	RO	RESERVED
0	ENABLE	0x0	RW	TPIU Enable field. When set, the ARM M4 TPIU is enabled and data can be streamed out of the MCU's SWO port using the ARM ITM and TPIU modules. DIS = 0x0 - Disable the TPIU. EN = 0x1 - Enable the TPIU.

4.10 Memory Subsystem

Figure 4-1: Memory Subsystem



4.10.1 Features

The Apollo2 SoC integrates four kinds of memory as shown in Figure 4-1:

- SRAM (with Flash cache)
- Boot Loader ROM
- One Time Programmable (OTP) memory

Key features include:

- 256 KB SRAM
- 2 instances of 512 KB flash memory (up to 1 MB total)
- 16 KB Flash cache (2-way set-associative, 512 entry, 128-bit line size)
- 16 KB OTP
  - 8 KB contain factory preset per chip trim values.
  - 8 KB for customer use, including flash protection fields
- Flash Protection specified in 16 KB Chunks
  - 64 OTP bits specify Write Protected Chunks
  - 64 OTP bits specify Read Protected Chunks
  - A Chunk is Execute Only if Both Corresponding Protection Bits Specified
  - OTP bits Specify Debugger Lock Out State
  - OTP bits Can Protect SRAM Contents From Debugger Inspection

4.10.2 Functional Overview

The Apollo2 SoC Integrates up to 1024 KB of on-board Flash non-volatile memory and 16 KB of one time programmable memory. These memories are managed by the APB flash controller for write operations.

During normal SoC code execution, the Flash Cache Controller translates requests from the CPU core to the Flash Memory instance for instruction and data fetches. The Controller is designed to return data in zero wait-states when accesses hit into the cache and can operate up to the maximum operating frequency of the CPU core. On cache misses, the controller issues miss requests to the Flash memory controller.

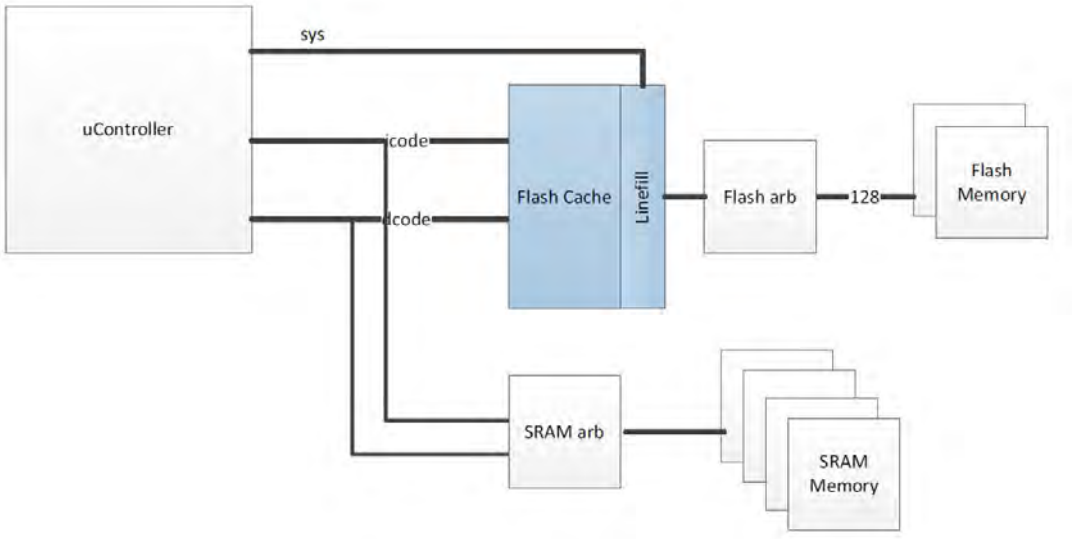
The Flash Memory Controller facilitates flash erase and programming operations. When erase or programming operations are active, instructions cannot be fetched for execution from the Flash memory, so the on-chip SRAM would have to be used for code execution. The cache controller ensures these operations are synchronized. To facilitate the management of Flash updates and OTP programming, a number of Flash helper functions are provided in the boot loader ROM.

The boot loader ROM contains2 instructions that are executed upon power up of the processor. Once a valid reset vector is establish at offset zero in the flash memory, the boot loader transfers control to users application by issuing a POR type reset which causes the core to enter the reset vector in flash. This process occupies less than 100 instructions in the boot loader. The remainder of the boot loader is occupied by a set of flash helper functions.

4.10.3 Flash Cache

4.10.3.1 Functional Overview

Figure 4-2: Block Diagram for Apollo2 SoC with Flash Cache



Apollo2 SoC incorporates a Flash cache to the ICode and DCode path from the microcontroller. This controller is intended to provide single cycle read access to Flash and reduce overall accesses to the Flash to reduce power. The controller is a unified ICode and DCode cache controller. The cache fill path is arbitrated between cache misses as well as the other Flash read agents (Info, Reg, BIST). Flash The cache is configurable 2-way set associative, 128b line size.

4.10.3.2 Cache Operation

To enable the cache, software should write the CACHECFG register with the desired setting. The ENABLE field in this register will power up the cache SRAMs and initiate the cache startup sequence which will flush the cache RAMs. Once the sequence is complete (indicated by the CACHE_READY bit in the CACHECTRL register), the cache will automatically begin servicing instruction and/or data fetches from the cache depending on the state of the ICACHE_ENABLE and DCACHE_ENABLE values. Software can choose to enable/disable these independently and they can be dynamically changed during operation.

Additionally, the non-cachable region registers can be used to mark regions as non-cached, which supersedes the I/D enable bits and causes all fetches from within this range to be non-cached.

The cache will automatically flush data contents if flash is erased/programmed or if the primary cache enable bit is disabled. Additionally, software can invalidate the cache by writing the INVALIDATE bit of the CACHECTRL register. Since this register

contains only status information (on reads) and activates controls based on bits set, there is no need to perform a read-modify-write.

For any mode changes, the cache should first be disabled by writing the ENABLE bit to 0, changing the configuration, then re-writing the enable bit to a 1.

4.10.3.3 Cache Performance Monitors

The cache also includes logic to monitor cache performance, which should be used in conjunction with the STIMER or CTIMER to determine elapsed time. The instruction and data buses have independent monitoring logic that keep counts of the following conditions:

- ACCESS_COUNT - total number of reads performed on the bus
- LOOKUP_COUNT - number of tag lookups performed
- HIT_COUNT - number of tag lookups that result in a hit
- LINE_COUNT - number of reads that were serviced from the line buffers (on a miss or non-cached access) or directly from the RAM because they fell within the same line as the previous lookup.

The LOOKUP and LINE counts should sum to the ACCESS COUNT and the number of cache misses can be calculated as LOOKUP_COUNT - HIT_COUNT.

NOTE: The DMONn and IMONn registers should be read with the cache monitor disabled (CACHECFG[ENABLE_MONITOR] = 0x0).

Cache monitor counters will automatically freeze the counts when either of the access counters reaches a value of 0xFFFF0000 to prevent the counters from rolling over. The monitor counts can be reset at any time by writing the RESET_STAT bit in the CACHECTRL register.

The monitors do not provide an indication of wait-states added to accesses, so the elapsed time should be used to infer this value (wait states are added as a result of cache misses or contention for the tag lookup if both buses require a simultaneous lookup).

4.11 CACHECTRL Registers

Flash Cache Controller
INSTANCE 0 BASE ADDRESS:0x40018000

4.11.1 Register Memory Map

Table 4-173: CACHECTRL Register Map

Address(es)	Register Name	Description
0x40018000	CACHECFG	Flash Cache Control Register
0x40018004	FLASHCFG	Flash Control Register
0x40018008	CACHECTRL	Cache Control
0x40018010	NCR0START	Flash Cache Noncachable Region 0 Start Address
0x40018014	NCR0END	Flash Cache Noncachable Region 0 End
0x40018018	NCR1START	Flash Cache Noncachable Region 1 Start
0x4001801C	NCR1END	Flash Cache Noncachable Region 1 End
0x40018030	CACHEMODE	Flash Cache Mode Register. Used to trim performance/power.
0x40018040	DMON0	Data Cache Total Accesses
0x40018044	DMON1	Data Cache Tag Lookups
0x40018048	DMON2	Data Cache Hits
0x4001804C	DMON3	Data Cache Line Hits
0x40018050	IMON0	Instruction Cache Total Accesses
0x40018054	IMON1	Instruction Cache Tag Lookups
0x40018058	IMON2	Instruction Cache Hits
0x4001805C	IMON3	Instruction Cache Line Hits

4.11.2 CACHECTRL Registers

4.11.2.1 CACHECFG Register

Flash Cache Control Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40018000
Flash Cache Control Register

Table 4-174: CACHECFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD							ENABLE_MONITOR	RSVD			DATA_CLKGATE	SMDLY				DLY				CACHE_LS	CACHE_CLKGATE	DCACHE_ENABLE	ICACHE_ENABLE	SERIAL	CONFIG			ENABLE_NC1	ENABLE_NC0	LRU	ENABLE

Table 4-175: CACHECFG Register Bits

Bit	Name	Reset	RW	Description
31:25	RSVD	0x0	RO	This bit field is reserved for future use.
24	ENABLE_MONITOR	0x0	RW	Enable Cache Monitoring Stats.Only enable this for debug/ performance analysis since it will consume additional power.See IMON/DMON registers for data.
23:21	RSVD	0x0	RO	This bit field is reserved for future use.
20	DATA_CLKGATE	0x1	RW	Enable clock gating of entire cache data array subsystem. This should be enabled for normal operation.
19:16	SMDLY	0x6	RW	Unused. Should be left at default value.
15:12	DLY	0x6	RW	Unused. Should be left at default value.
11	CACHE_LS	0x1	RW	Enable LS (light sleep) of cache RAMs. This should not be enabled for normal operation. When this bit is set, the cache's RAMS will be put into light sleep mode while inactive. NOTE: If the cache is actively used, this may have an adverse affect on power since entering/exiting LS mode may consume more power than would be saved.
10	CACHE_CLKGATE	0x1	RW	Enable clock gating of individual cache RAMs. This bit should be enabled for normal operation for lowest power consumption.
9	DCACHE_ENABLE	0x0	RW	Enable Flash Data Caching. When set to 1, all instruction accesses to flash will be cached.
8	ICACHE_ENABLE	0x0	RW	Enable Flash Instruction Caching. When set to 1, all instruction accesses to flash will be cached.
7	SERIAL	0x0	RW	Bit field should always be programmed to 0.
6:4	CONFIG	0x5	RW	Sets the cache configuration. Only a single configuration of 0x5 is valid. W2_128B_512E = 0x5 - Two-way set associative, 128-bit linesize, 512 entries (8 SRAMs active)

Table 4-175: CACHECFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
3	ENABLE_NC1	0x0	RW	Enable Non-cacheable region 1. See the NCR1 registers to set the region boundaries and size.
2	ENABLE_NC0	0x0	RW	Enable Non-cacheable region 0. See the NCR0 registers to set the region boundaries and size.
1	LRU	0x0	RW	Sets the cache replacement policy. 0=LRR (least recently replaced), 1=LRU (least recently used). LRR minimizes writes to the TAG SRAM and is recommended.
0	ENABLE	0x0	RW	Enables the main flash cache controller logic and enables power to the cache RAMs. Instruction and Data caching need to be enabled independently using the ICACHE_ENABLE and DCACHE_ENABLE bits.

4.11.2.2 FLASHCFG Register

Flash Control Register
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40018004
Flash Control Register

Table 4-176: FLASHCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													RD_WAIT		

Table 4-177: FLASHCFG Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	This bit field is reserved for future use.
2:0	RD_WAIT	0x1	RW	Sets read wait states for flash accesses (in clock cycles). This should be left at the default value for normal flash operation.

4.11.2.3 CACHECTRL Register

Cache Control
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40018008
Cache Control

Table 4-178: CACHECTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					FLASH1_SLM_ENABLE	FLASH1_SLM_DISABLE	FLASH1_SLM_STATUS	RSVD	FLASH0_SLM_ENABLE	FLASH0_SLM_DISABLE	FLASH0_SLM_STATUS	RSVD	CACHE_READY	RESET_STAT	INVALIDATE

Table 4-179: CACHECTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield is reserved for future use.
10	FLASH1_SLM_ENABLE	0x0	WO	Enable Flash Sleep Mode. After writing this bit, the flash instance 1 will enter a low-power mode until the CPU writes the SLM_DISABLE bit or a flash access occurs. Wake from SLM requires ~5us, so this should only be set if the flash will not be accessed for reasonably long time.
9	FLASH1_SLM_DISABLE	0x0	WO	Disable Flash Sleep Mode. Allows CPU to manually disable SLM mode. Performing a flash read will also wake the array.
8	FLASH1_SLM_STATUS	0x0	RO	Flash Sleep Mode Status. When 1, flash instance 1 is asleep.
10:7	RSVD	0x0	RO	This bit field is reserved for future use.
6	FLASH0_SLM_ENABLE	0x0	WO	Enable Flash Sleep Mode. After writing this bit, the flash instance 0 will enter a low-power mode until the CPU writes the SLM_DISABLE bit or a flash access occurs. Wake from SLM requires ~5us, so this should only be set if the flash will not be accessed for reasonably long time.
5	FLASH0_SLM_DISABLE	0x0	WO	Disable Flash Sleep Mode. Allows CPU to manually disable SLM mode. Performing a flash read will also wake the array.
4	FLASH0_SLM_STATUS	0x0	RO	Flash Sleep Mode Status. When 1, flash instance 0 is asleep.
6:3	RSVD	0x0	RO	This bit field is reserved for future use.
2	CACHE_READY	0x0	RO	Cache Ready Status. A value of 1 indicates the cache is enabled and not processing an invalidate operation.
1	RESET_STAT	0x0	WO	Writing a 1 to this bit field will reset the cache monitor statistics (DMON0-3, IMON0-3). Statistic gathering can be paused/stopped by disabling the MONITOR_ENABLE bit in CACHECFG, which will maintain the count values until the stats are reset by writing this bit field. CLEAR = 0x1 - Clear Cache Stats
0	INVALIDATE	0x0	WO	Writing a 1 to this bit field invalidates the flash cache contents. GO = 0x1 - Initiate a programming operation to flash info.

4.11.2.4 NCR0START Register

Flash Cache Noncachable Region 0 Start Address.
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40018010
Flash Cache Noncachable Region 0 Start Address.

Table 4-180: NCR0START Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												ADDR																RSVD			

Table 4-181: NCR0START Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bit field is reserved for future use.
19:4	ADDR	0x0	RW	Start address for non-cacheable region 0. The physical address of the start of this region should be programmed to this register and must be aligned to a 16-byte boundary (thus the lower 4 address bits are unused).
3:0	RSVD	0x0	RO	This bit field is reserved for future use.

4.11.2.5 NCR0END Register

Enable the fault capture registers
OFFSET: 0x000001D0
INSTANCE 0 ADDRESS: 0x400201D0
Enable the fault capture registers

Table 4-182: NCR0END Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												ADDR																RSVD			

Table 4-183: NCR0END Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bit field is reserved for future use.
19:4	ADDR	0x0	RW	End address for non-cacheable region 0. The physical address of the end of this region should be programmed to this register and must be aligned to a 16-byte boundary (thus the lower 4 address bits are unused).
3:0	RSVD	0x0	RO	This bit field is reserved for future use.

4.11.2.6 NCR1START Register

Flash Cache Noncachable Region 1 Start
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x40018018
Flash Cache Noncachable Region 1 Start

Table 4-184: NCR1START Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												ADDR																RSVD			

Table 4-185: NCR1START Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bit field is reserved for future use.
19:4	ADDR	0x0	RW	Start address for non-cacheable region 1. The physical address of the start of this region should be programmed to this register and must be aligned to a 16-byte boundary (thus the lower 4 address bits are unused).
3:0	RSVD	0x0	RO	This bit field is reserved for future use.

4.11.2.7 NCR1END Register

Flash Cache Noncachable Region 1 End
OFFSET: 0x0000001C
INSTANCE 0 ADDRESS: 0x4001801C
Flash Cache Noncachable Region 1 End

Table 4-186: NCR1END Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												ADDR																RSVD			

Table 4-187: NCR1END Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bit field is reserved for future use.
19:4	ADDR	0x0	RW	End address for non-cacheable region 1. The physical address of the end of this region should be programmed to this register and must be aligned to a 16-byte boundary (thus the lower 4 address bits are unused).
3:0	RSVD	0x0	RO	This bit field is reserved for future use.

4.11.2.8 DMON0 Register

Data Cache Total Accesses
OFFSET: 0x00000040
INSTANCE 0 ADDRESS: 0x40018040
Data Cache Total Accesses

Table 4-188: DMON0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DACCESS_COUNT																															

Table 4-189: DMON0 Register Bits

Bit	Name	Reset	RW	Description
31:0	DACCESS_COUNT	0x0	RO	Total accesses to data cache.

4.11.2.9 DMON1 Register

Data Cache Tag Lookups
OFFSET: 0x00000044
INSTANCE 0 ADDRESS: 0x40018044
Data Cache Tag Lookups

Table 4-190: DMON1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DLOOKUP_COUNT																															

Table 4-191: DMON1 Register Bits

Bit	Name	Reset	RW	Description
31:0	DLOOKUP_COUNT	0x0	RO	Total tag lookups from data cache

4.11.2.10 DMON2 Register

Data Cache Hits
OFFSET: 0x00000048
INSTANCE 0 ADDRESS: 0x40018048
Data Cache Hits

Table 4-192: DMON2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DHIT_COUNT																															

Table 4-193: DMON2 Register Bits

Bit	Name	Reset	RW	Description
31:0	DHIT_COUNT	0x0	RO	Cache hits from lookup operations

4.11.2.11 DMON3 Register

Data Cache Line Hits
OFFSET: 0x0000004C
INSTANCE 0 ADDRESS: 0x4001804C
Data Cache Line Hits

Table 4-194: DMON3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DLINE_COUNT																															

Table 4-195: DMON3 Register Bits

Bit	Name	Reset	RW	Description
31:0	DLINE_COUNT	0x0	RO	Cache hits from line cache.

4.11.2.12 IMON0 Register

Instruction Cache Total Accesses
OFFSET: 0x00000050
INSTANCE 0 ADDRESS: 0x40018050
Instruction Cache Total Accesses

Table 4-196: IMON0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IACCESS_COUNT																															

Table 4-197: IMON0 Register Bits

Bit	Name	Reset	RW	Description
31:0	IACCESS_COUNT	0x0	RO	Total accesses to Instruction cache.

4.11.2.13 IMON1 Register

Instruction Cache Tag Lookups
OFFSET: 0x00000054
INSTANCE 0 ADDRESS: 0x40018054
Instruction Cache Tag Lookups

Table 4-198: IMON1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ILOOKUP_COUNT																															

Table 4-199: IMON1 Register Bits

Bit	Name	Reset	RW	Description
31:0	ILOOKUP_COUNT	0x0	RO	Total tag lookups from Instruction cache.

4.11.2.14 IMON2 Register

Instruction Cache Hits
OFFSET: 0x00000058
INSTANCE 0 ADDRESS: 0x40018058
Instruction Cache Hits

Table 4-200: IMON2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IHIT_COUNT																															

Table 4-201: IMON2 Register Bits

Bit	Name	Reset	RW	Description
31:0	IHIT_COUNT	0x0	RO	Cache hits from lookup operations.

4.11.2.15 IMON3 Register

Instruction Cache Line Hits
OFFSET: 0x0000005C
INSTANCE 0 ADDRESS: 0x4001805C
Instruction Cache Line Hits

Table 4-202: IMON3 Register

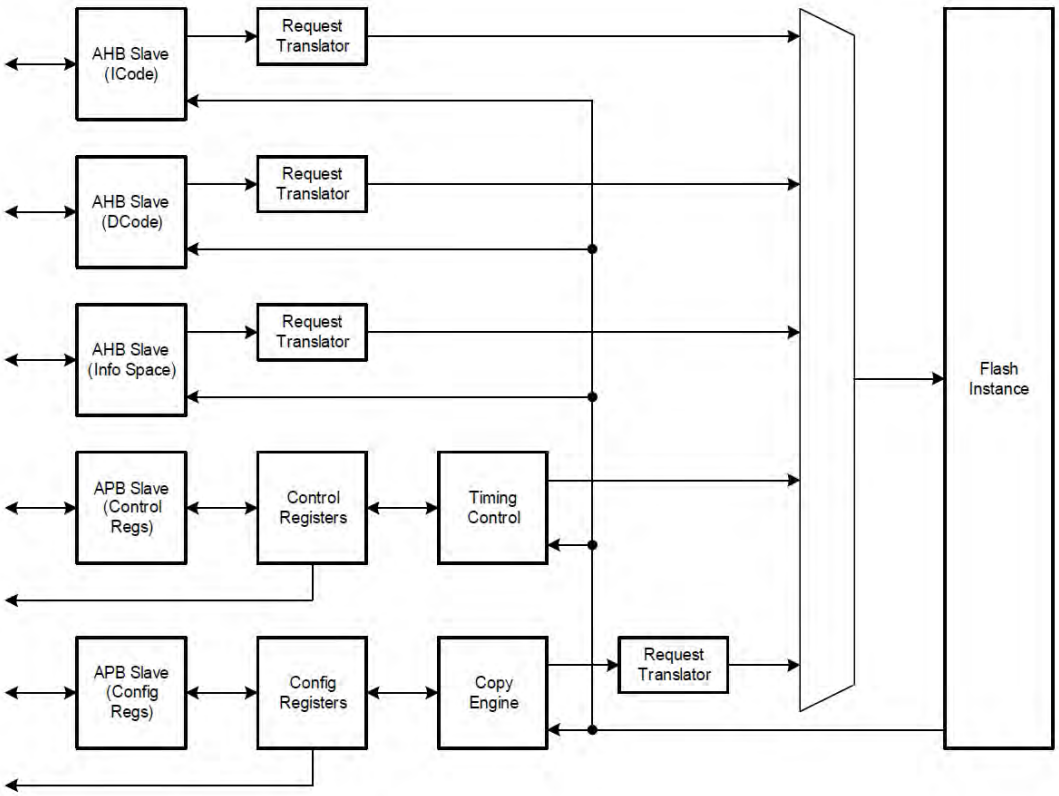
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ILINE_COUNT																															

Table 4-203: IMON3 Register Bits

Bit	Name	Reset	RW	Description
31:0	ILINE_COUNT	0x0	RO	Cache hits from line cache.

4.12 Flash Memory Controller

Figure 4-3: Detailed Block Diagram



4.12.1 Functional Overview

During normal SoC code execution, the Flash Memory Controller translates requests from the CPU core (via the Flash cache) to the Flash Memory instance for instruction and data fetches. The Controller is designed to return data to the cache in single wait-state and can operate up to the maximum operating frequency of half the CPU core frequency.

The Controller facilitates flash erase and programming operations through the control registers. When erase or programming operations are active, data cannot be fetched from the Flash memory. This will be naturally handled by the cache controller fill logic to stall until the program operation is complete and the Flash device is available. With the cache enabled, this collision should happen very infrequently.

Another function of the Controller is to capture the configuration values which are distributed to the various on-chip peripherals of the SoC at chip power-up. These are read from the Information Space of the Flash Memory and captured in registers to be used by the other peripherals. The configuration values are reloaded each time a full-chip POI cycle occurs.

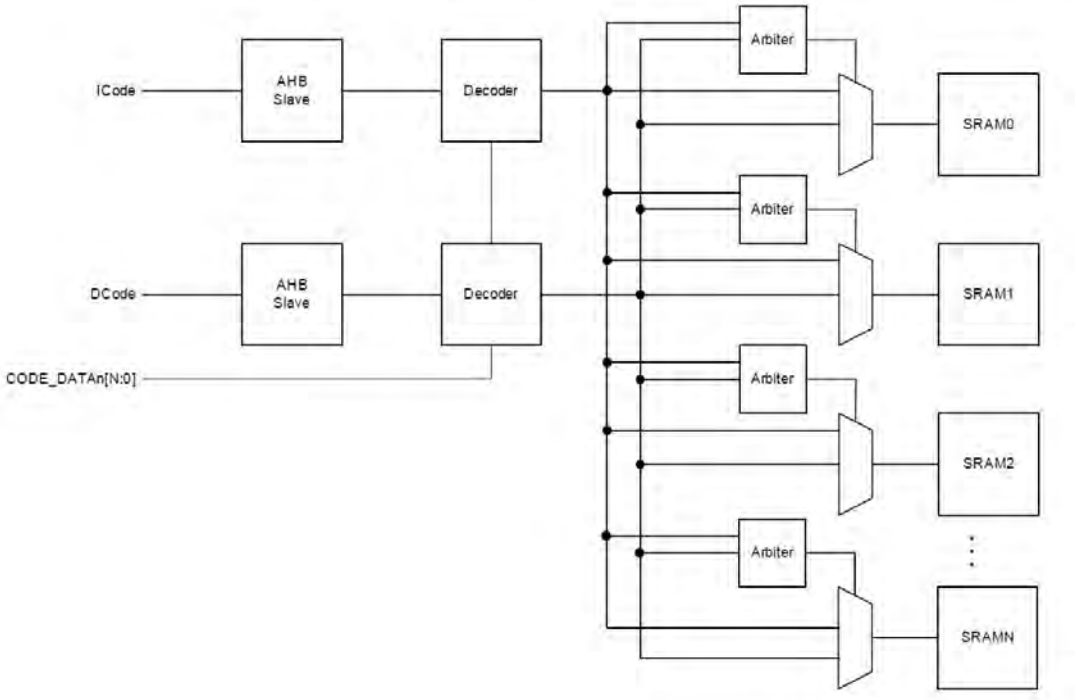
NOTES:

- A Flash page size is 8 KB, which is the minimum size that can be erased. An erase cycle will set all the bits in the Flash page to 1's and bits can be programmed only to a zero state. The same 32-bit word can be programmed a maximum of 4 times, otherwise data corruption or retention issues may appear within the word line. The AmbiqSuite SDK provides a "modify" function to help with this.
- The number of individual word-size programming cycles performed per word line (512 bytes) should be limited to no more than 160 before an erase. Full-line programming of the entire 512-byte word line is faster and less stressful such that programming all 512 bytes in a single transaction counts as only 60 individual writes. Therefore, following a a full-line programming of all 512 bytes, no more than an additional 100 word-programming operations can be performed before an erase is required. Doing more than the specified number of program cycles to the same line before an erase operation may cause data corruption or retention issues within the word line.

4.13 SRAM Interface

4.13.1 Functional Overview

Figure 4-4: Block Diagram for the SRAM Interface



The SRAM Interface translates requests from the CPU core to the SRAM Memory Instances for instruction and data fetches. The SRAM interface is designed to return data in zero wait-states and can operate up to the maximum operating frequency of the CPU core. The Interface contains arbitration logic for each SRAM instance which allows one of 2 bus slaves access to the SRAM on any given cycle.

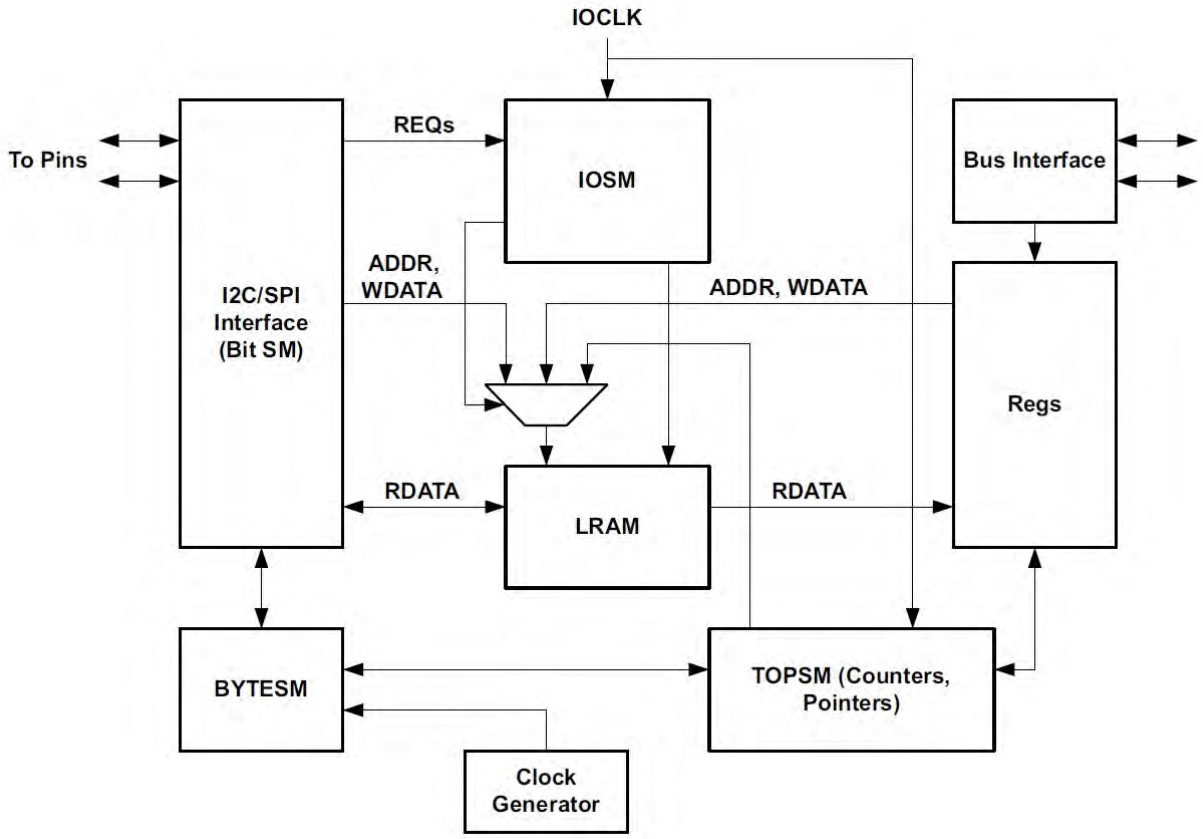
Figure 4-4 shows a logical block diagram of the SRAM Interface, where n = 7 for this SoC.

SECTION

5

I²C/SPI Master Module

Figure 5-1: Block Diagram for the I²C/SPI Master Module



5.1 Functional Overview

The Apollo2 SoC includes six I^{2}C/SPI Master Modules (IOM), shown in Figure 5-1 on page 141, each of which functions as the master of an I^{2}C or SPI interface as selected by the IOMSTRn_IOMCFG_IFCSEL bit. A 128-byte bidirectional FIFO and a sophisticated Command mechanism allow simple initiation of I/O operations without requiring software interaction.

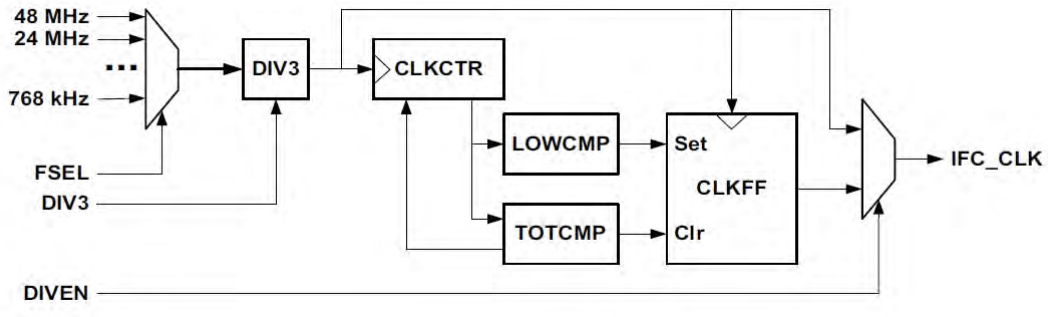
In I^{2}C mode the I^{2}C/SPI Master supports 7- and 10-bit addressing, multi-master arbitration, interface frequencies from 1.2 kHz to 1.0 MHz and up to 255-byte burst operations. In SPI mode the I^{2}C/SPI Master supports up to 8 slaves with automatic nCE selection, 3 and 4-wire implementation, all SPI polarity/phase combinations and up to 4095-byte burst operations, with both standard embedded address operations and raw read/write transfers. Interface timing limits are as specified in the Serial Peripheral Interface (SPI) Master Interface table of the Electrical Characteristics chapter.

The Apollo2 SoC supports four Master SPI ports (1-3 and 5) and two High Speed Master SPI ports (0 and 4).

5.2 Interface Clock Generation

The I^{2}C/SPI Master can generate a wide range of I/O interface clocks, as shown in Figure 5-2. The source clock is a scaled version of the HFRC 48 MHz clock, selected by IOMSTRn_CLKCFG_FSEL. A divide-by-3 circuit may be selected by IOMSTRn_CLKCFG_DIV3, which is particularly important in creating a useful SPI frequency of 16 MHz. The output of the divide-by-3 circuit may then be divided by an 8-bit value, IOMSTRn_CLKCFG_TOTPER + 1, to produce the interface clock. This structure allows very precise specification of the interface frequency, and produces a minimum available interface frequency of 1.2 kHz. If TOTPER division is enabled by IOMSTRn_CLKCFG_DIVEN, the length of the low period of the clock is specified by IOMSTRn_CLKCFG_LOWPER + 1. Otherwise, the clock will have a 50% duty cycle.

Figure 5-2: I^{2}C/SPI Master Clock Generation



5.3 Command Operation

In order to minimize the amount of time the CPU must be awake during I^{2}C/SPI Master operations, the architecture of the I^{2}C/SPI Master is organized around processing commands which transfer data to and from an internal 128-byte FIFO.

Table 5-1 shows the format of the CMD Register for an I^{2}C operation, when the IOMSTRn_IOMCFG_IFCSEL bit is 0. Table 5-2 shows the format of the operation, when the IOMSTRn_IOMCFG_IFCSEL bit is a 1.

Table 5-1: CMD Register for I^{2}C Operations

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OPER			CONT	LSB	10-BIT	ADDRESS										OFFSET								LENGTH							

Table 5-2: CMD Register for SPI Operations

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OPER			CONT	LSB	UPLNGTH				RSVD				CHN			OFFSET								LENGTH							

Table 5-3: CMD Register Field Description

Bit	Name	Reset	RW	Description
31:29	OPER	0x0	RW	This field describes the operation. The valid operations are: 000: Normal Write 010: Raw Write 100: Normal Read 110: Raw Read
28	CONT	0x0	RW	If set to 1, the operation will be continued (see *Section 5.5.12 Holding the Interface with CONT on page 150* and *Section 5.6.9 SPI Polarity and Phase on page 154*)
27	LSB	0x0	RW	If set to 0, data bytes are transferred most significant bit first. If set to 1, data bytes are transferred least significant bit first.
26	10-BIT (I^{2}C ONLY)	0x0	RW	If set to 1, 10-bit I^{2}C addressing is used. If set to 0, 7-bit I^{2}C addressing is used.
25:16	ADDRESS (I^{2}C ONLY)	0x0	RW	Address of the slave device on the I^{2}C bus. If 7-bit addressing is specified, only the lower 7 bits of this fields are used.
26:23	UPLNGTH (SPI ONLY)	0x0	RW	This field contains the upper four bits of the 12-bit length field, allowing transfers up to 4095 bytes in SPI mode.
22:19	RSVD (SPI ONLY)	0x0	RW	RESERVED
18:16	CHN (SPI ONLY)	0x0	RW	This field selects which of the eight nCE outputs will be asserted for this transfer. This allows interfacing to up to eight separate SPI devices by simply using different CHN values.

Table 5-3: CMD Register Field Description *(Continued)*

Bit	Name	Reset	RW	Description
15:8	OFFSET	0x0	RW	In I^2C mode, this field specifies the data transferred in the OFFSET byte for Normal Read and Write transfers. In SPI mode, this field specifies the first byte written in Normal Read and Write transfers. Raw transfers in either I^2C or SPI mode do not use the OFFSET field.
7:0	LENGTH	0x0	RW	This field defines the length of the transfer in bytes. For Normal Read and Write operations, the LENGTH field may be set to 0, so that only the OFFSET byte is written and no other data is transferred. In SPI mode, the UPLNGTH field contains the upper 4 bits of the LENGTH so that up to 4095 bytes may be transferred in a single operation.

For writes to the interface, software writes data to the FIFO (IOMSTRn_FIFO) and then sends a single command to the IOMSTRn_CMD Register. Unless the LENGTH field of the CMD is zero, at least one word (4 bytes) of data must be written into the FIFO prior to writing the CMD Register or an ICMD interrupt will be generated and the operation will be terminated.The Command includes either the I^2C slave address or the SPI channel select, the desired address offset and the length of the transfer. At that point the I^2C/SPI Master executes the entire transfer, so the CPU can go to sleep. If more than 128 bytes are to be transferred, the Master will generate a THR interrupt when the FIFOSIZ value, IOMSTRn_FIFOPTR_FIFOSIZ, drops below the write threshold IOMSTRn_FIFOTHR_FIFOWTHR so the CPU can wake up and refill the FIFO. The I^2C/SPI Master will generate the CMDCMP interrupt when the command is complete. In each case, the total number of bytes transferred in each operation is specified in the LENGTH field of the CMD Register. If software executes a write to the FIFO when it is full (FIFOSIZ is greater than 124) the FOVFL interrupt will be generated and the transfer will be terminated.

For reads, the CMD Register is first written with the command and the CPU can go to sleep. The Master initiates the read and transfers read data to the FIFO. If the FIFOSZ value exceeds the read threshold IOMSTRn_FIFOTHR_FIFORTHR, a THR interrupt is generated so the CPU can wake up and empty the FIFO. A CMDCMP interrupt is also generated when the Command completes. If software executes a read from the FIFO when it has less than a word of data the FUNDFL interrupt will be generated and the transfer will be terminated. FUNDFL will not be generated if the read transfer has already completed, so that software can read the last FIFO word even if it is incomplete.

If the FIFO empties on a write or fills on a read, the I^2C/SPI Master will simply pause the interface clock until the CPU has read or written a byte from the FIFO. This avoids the requirement that the thresholds be set conservatively so that the processor can wake up fewer times on long transfers without a risk of an underflow or overflow aborting a transfer in progress.

If software initiates an incorrect operation, such as attempting to read the FIFO on a write operation or when it is empty, or write the FIFO on a read operation or

when it is full, the Master will generate an IACC error interrupt. If software attempts to write the Command Register when another Command is underway or write the CMD register with a write command when the FIFO is empty (unless the LENGTH field in the CMD is zero), the Master will generate an ICMD error interrupt.

5.4 FIFO

The I^{2}C/SPI Master includes a 128-byte local RAM (LRAM) for data transfers. The LRAM functions as a FIFO. Only 32-bit word accesses are supported to the FIFO from the CPU. When a write operation is underway, a word written to the FIFO will increment the IOMSTRn_FIFOPTR_FIFOSIZ register by 4 and decrement the IOMSTRn_FIFOPTR_FIFOREM register by 4. Reading a byte from the FIFO via the I/O interface decrements FIFOSIZ by 1 and increments FIFOREM by 1. When a read operation is underway, a word read from the FIFO decrements FIFOSIZ by 4 and increments FIFOREM by 4. A byte read from the I/O interface into the FIFO increments FIFOSIZ by 1 and decrements FIFOREM by 1. If FIFOSIZ becomes one during a write operation or 0x40 on a read operation and there is more data to be transferred, the clock of the I/O interface is paused until software accesses the FIFO.

Two threshold registers, FIFORTHR and FIFOWTHR indicate when a THR interrupt should be generated to signal the processor that data should be transferred.

5.5 I^{2}C Interface

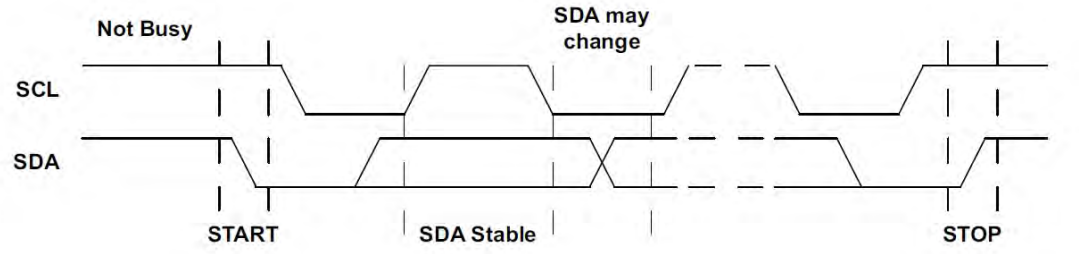
The I^{2}C/SPI Master supports a flexible set of Commands to implement a variety of standard I^{2}C operations. The I^{2}C interface consists of two lines: one bi-directional data line (SDA) and one clock line (SCL). Both the SDA and the SCL lines must be connected to a positive supply voltage via a pull-up resistor. By definition, a device that sends a message is called the “transmitter”, and the device that accepts the message is called the “receiver”. The device that controls the message transfer by driving SCL is called “master”. The devices that are controlled by the master are called “slaves”. The Apollo2 SoC I^{2}C Master is always a master device.

The following protocol has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high.
- Changes in the data line while the clock line is high will be interpreted as control signals.

A number of bus conditions have been defined (see Figure 5-3 on page 146) and are described in the following sections.

Figure 5-3: Basic I^{2}C Conditions



5.5.1 Bus Not Busy

Both SDA and SCL remain high.

5.5.2 Start Data Transfer

A change in the state of SDA from high to low, while SCL is high, defines the START condition. A START condition which occurs after a previous START, but before a STOP, is called a RESTART condition, and functions exactly like a normal STOP followed by a normal START.

5.5.3 Stop Data Transfer

A change in the state of SDA from low to high, while SCL is high, defines the STOP condition.

5.5.4 Data Valid

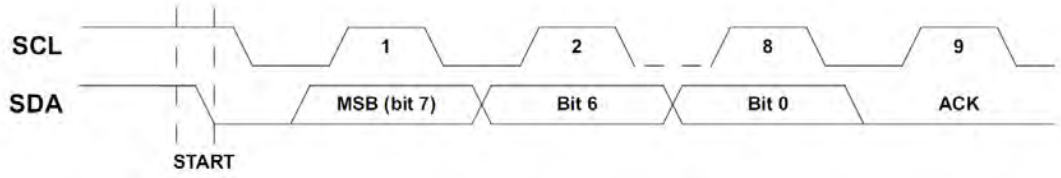
After a START condition, SDA is stable for the duration of the high period of SCL. The data on SDA may be changed during the low period of SCL. There is one clock pulse per bit of data. Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and STOP conditions is not limited. The information is transmitted byte wide and each receiver acknowledges with a ninth bit.

5.5.5 Acknowledge

Each byte of eight bits is followed by one acknowledge (ACK) bit as shown in Figure 10. This acknowledge bit is a low level driven onto SDA by the receiver, whereas the master generates an extra acknowledge related SCL pulse. A slave receiver which is addressed is obliged to generate an acknowledge after the reception of each byte. Also, on a read transfer, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges must pull down the SDA line during the

acknowledge clock pulse in such a way that the SDA line is a stable low during the high period of the acknowledge related SCL pulse. A master receiver must signal an end-of-data to the slave transmitter by not generating an acknowledge (a NAK) on the last byte that has been clocked out of the slave. In this case, the transmitter must leave the data line high to enable the master to generate the STOP condition. If I/O Host attempts an I^2C operation but no slave device generates an ACK, or if a slave fails to generate an ACK on a data byte before the transfer is complete, a NAK interrupt will be generated.

Figure 5-4: I^2C Acknowledge



5.5.6 I^2C Slave Addressing

For normal I^2C reads and writes, the Command specifies the address to be sent on the interface. Both 7-bit and 10-bit addressing are supported, as selected by 10BIT in the Command. The address is specified in the ADDRESS field.

Figure 5-5 shows the operation in 7-bit mode in which the master addresses the slave with a 7-bit address configured as 0xD0 in the lower 7 bits of the ADDRESS field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of an attached slave device, the eighth bit indicates a write (RW = 0) or a read (RW = 1) operation and the slave supplies the ACK. If no slave acknowledges the address, the transfer is terminated and a NAK error interrupt is generated.

Figure 5-5: I^2C 7-bit Address Operation

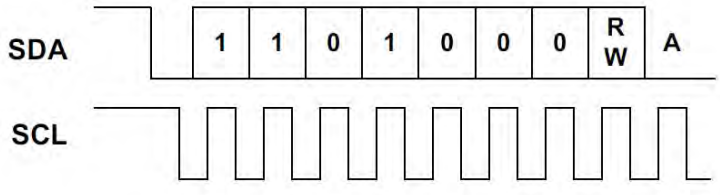
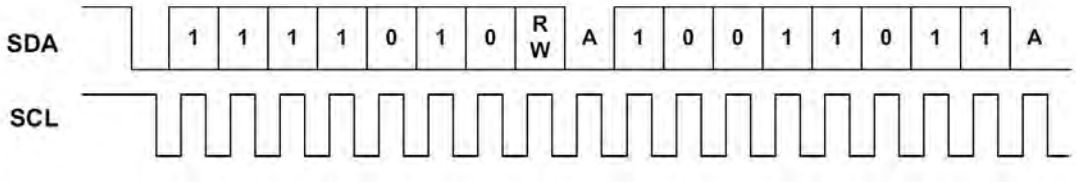


Figure 5-6 on page 148 shows the operation with which the master addresses the Apollo2 SoC with a 10-bit address configured at 0x536. After the START condition, the 10-bit preamble 0b11110 is transmitted first, followed by the upper two bits of the ADDRESS field and the eighth bit indicating a write (RW = 0) or a read (RW = 1) operation. If the upper two bits match the address of an attached slave device, it supplies the ACK. The next transfer includes the lower 8 bits of the ADDRESS field, and if these bits also match I2CADDR the slave again supplies the ACK. If no slave

acknowledges either address byte, the transfer is terminated and a NAK error interrupt is generated.

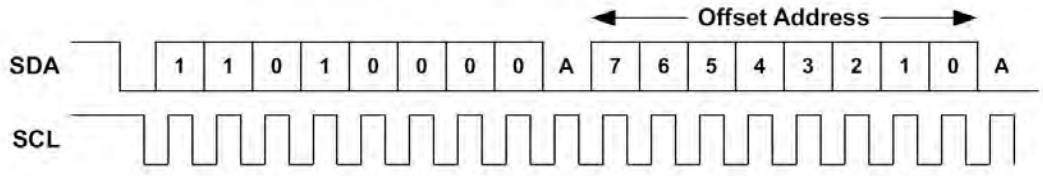
Figure 5-6: I2C 10-bit Address Operation



5.5.7 I2C Offset Address Transmission

If the OPER field of the CMD register selects a Normal Read or Write, the I2C/SPI Master will first send an 8-bit Offset Address byte, where the offset is specified in the OFFSET field of CMD. This transfer is shown in Figure 5-7. The Offset Address is loaded into the Address Pointer of the slave.

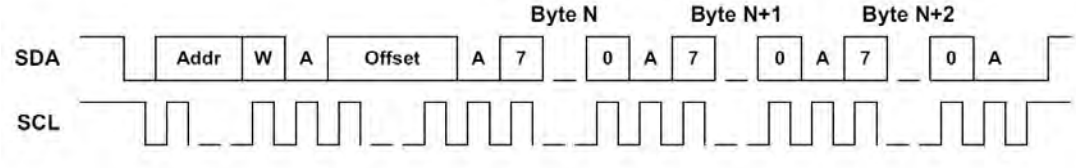
Figure 5-7: I2C Offset Address Transmission



5.5.8 I2C Normal Write Operation

In a Normal write operation the I2C/SPI Master transmits to a slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address, as in Figure 5-7. The next byte is written to the slave register selected by the Address Pointer (which was loaded with the Offset Address) and the Address Pointer is incremented. Subsequent transfers write bytes into successive registers until a STOP condition is received, as shown in Figure 5-8.

Figure 5-8: I2C Normal Write Operation

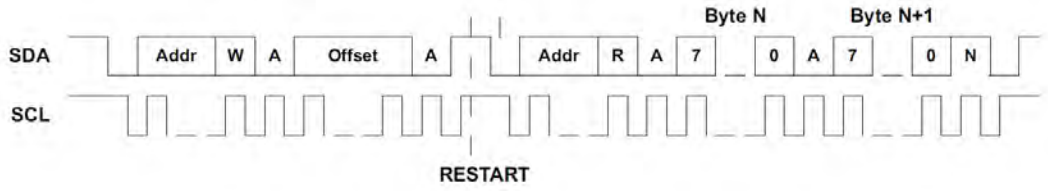


5.5.9 I²C Normal Read Operation

If a Normal Read operation is selected, the I²C/SPI Master first executes an Offset Address Transmission to load the Address Pointer of the slave with the desired Offset Address.

A subsequent operation will again issue the address of the slave but with the RW bit as a 1 indicating a read operation. As shown in Figure 5-9, this transaction begins with a RESTART condition so that the interface will be held in a multi-master environment. After the address operation, the slave becomes the transmitter and sends the register value from the location pointed to by the Address Pointer, and the Address Pointer is incremented. Subsequent transactions produce successive register values, until the I²C/SPI Master receiver responds with a NAK and a STOP to complete the operation.

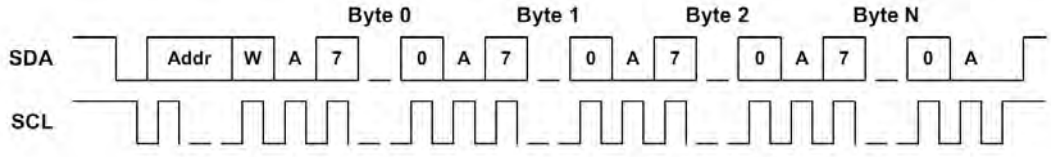
Figure 5-9: I²C Normal Read Operation



5.5.10 I²C Raw Write Operation

If a Raw Write is selected in the OPER field of the CMD register, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes as shown in Figure 5-10. This provides support for slave devices which do not implement the standard offset address architecture. The OFFSET field is not used in this case.

Figure 5-10: I²C Raw Write Operation

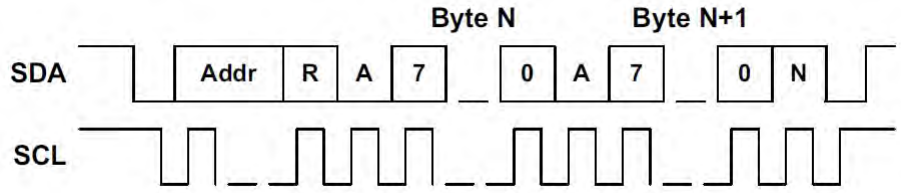


5.5.11 I²C Raw Read Operation

If a Raw Read is selected in the OPERfield of the CMD register, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes with a read as shown in Figure 5-11 on page 150. This is important for slave devices which do not support an Address Pointer architecture. For devices which do include an Address Pointer, multiple Raw Reads may be executed after a Normal

Read to access subsequent registers as the Address Pointer increments, without having to execute the Offset Address Transmission for each access.

Figure 5-11: I²C Raw Read Operation



5.5.12 Holding the Interface with CONT

In all of the previously described transactions, the I²C/SPI Master terminates the I²C operation with a STOP sequence. In environments where there are other masters connected to the I²C interface, it may be necessary for the Apollo2 SoC to hold the interface between Commands to insure that another master does not inadvertently access the same slave that the Apollo2 SoC is accessing. In order to implement this functionality, the CONT bit should be set in the CMD Register. This will cause the I²C/SPI Master to keep SDA high at the end of the transfer so that a STOP does not occur, and the next transaction begins with a RESTART instead of a START. Note that for a Normal Read the interface is held between the Offset Address Transmission and the actual read independent of the state of CONT, but it CONT is set the read transaction will not terminate with a STOP.

5.5.13 I²C Multi-Master Arbitration

The Apollo2 SoC I²C/SPI Master supports multi-master arbitration in I²C mode. There are two cases which must be handled.

The first is the case where another master initiates an I²C operation when the Apollo2 SoC Master is inactive. In this case the I²C/SPI Master will detect an I²C START operation on the interface and the START interrupt will be asserted, which tells the software not to generate any IO operations (which will not be executed in any case). Software then waits for the STOP interrupt, which reenables operation.

The second case is where another master initiates an operation at the same time as the Apollo2 SoC. In this case there will be a point where one master detects that it is not driving SDA low but the bus signal is low, and that master loses the arbitration to the other master. If the Apollo2 SoC I²C/SPI Master detects that it has lost arbitration, it will assert the ARB interrupt and immediately terminate its operation. Software must then wait for the STOP interrupt and re-execute the current Command.

5.6 SPI Operations

5.6.1 SPI Configuration

The I^2C/SPI Master supports all combinations of the polarity (CPOL) and phase (CPHA) modes of SPI using the IOMSTRn_IOMCFG_SPOL and IOMSTRn_IOMCFG_SPHA bits. It also may be configured in either 3-wire or 4-wire mode. In 4-wire mode, the MOSI and MISO interface signals use separate IO pins. In 3-wire mode, MOSI and MISO are multiplexed on a single IO pin for more efficient pin utilization. The 3/4 wire configuration is selected in the mapping function of the PINCFG module.

SPI operations may transfer up to 4095 bytes in a single transfer, as the UPLNGTH field of the CMD Register is appended to the standard LENGTH field to provided 12-bit length specification.

5.6.2 SPI Slave Addressing

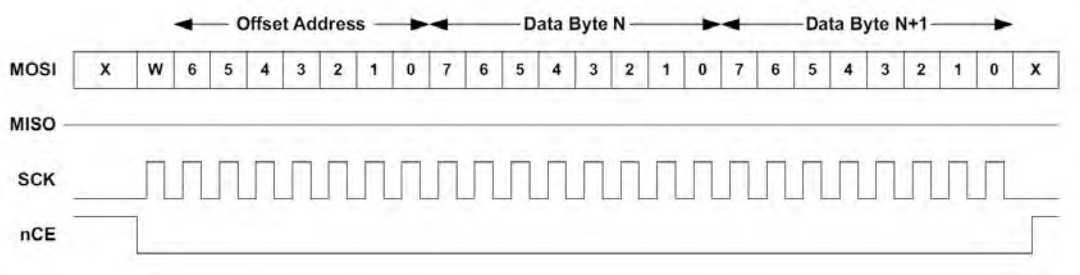
In SPI mode, the Command specifies the slave channel to be used in the CHN field. The I^2C/SPI Master supports up to eight slaves, each of which has its own nCE signal which can be configured on an IO pin. Additional slaves may be supported using GPIO pins and external decoding.

5.6.3 SPI Normal Write

Figure 5-12 on page 152 shows the case of a SPI Normal Write with a one-byte address offset operation, whereby a write operation is selected in the OPER field. The operation is initiated when the I^2C/SPI Master pulls one of the eight nCE signals low. At that point the I^2C/SPI Master begins generating the clock on SCK and the offset address is transmitted from the master on the MOSI line, with the upper R/W bit of the offset field indicating read (if 0) or write (if 1). In this example the R/W bit is a one selecting a write operation. The entire offset byte, including the R/W bit, is taken from the OFFSET field of the CMD. This means that OFFSET[7] should be set to 1 if the slave expects a R/W bit. If the slave does not expect a R/W bit, this allows the first byte of a write to be completely specified in the OFFSET field, and a single byte write in that case can be executed without requiring any data to be loaded in to the FIFO.

Each subsequent byte is read from the FIFO and transmitted. The operation is terminated when the I^2C/ SPI Master brings the nCE signal high. Note that the MISO line is not used in a write operation and is held in the high impedance state by the I^2C/SPI Master.

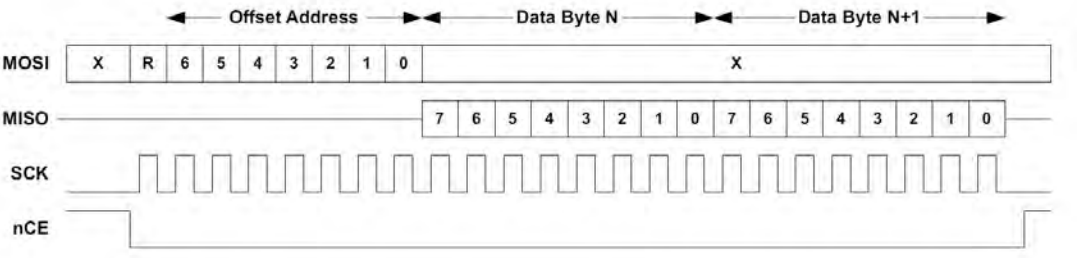
Figure 5-12: SPI Normal Write Operation (Single-byte Offset Address)



5.6.4 SPI Normal Read

Figure 5-13 shows the case of a Normal Read with a one-byte address offset operation, whereby a read operation is selected in the OPER field. The operation is initiated when the I^2C/SPI Master pulls one of the eight nCE signals low. At that point the I^2C/SPI Master begins driving the clock onto SCK and the address is transferred from the master to the slave just as it is in a write operation, but in this case the R/W bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the I^2C/SPI Master stops driving the MOSI line and begins loading the FIFO with the data on the MISO line. The transfer continues until the I^2C/SPI Master brings the nCE line high.

Figure 5-13: SPI Normal Read Operation

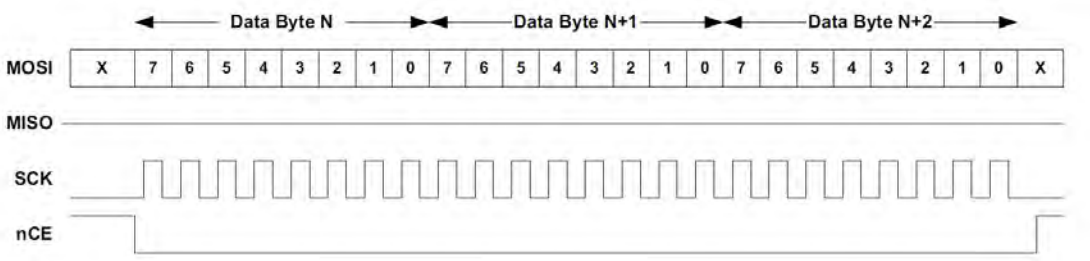


As with a Normal Write, the Offset Address byte including the R/W bit is taken from the offset field of CMD. If the slave expects a R/W bit, the msb of the offset must be set accordingly. This allows reads from devices which have different formats for the address byte.

5.6.5 SPI Raw Write

If a Raw Write is selected in the OPER field, the operation is similar to a Normal Write but the Offset Address byte is not sent and all data comes directly from the FIFO as shown in Figure 5-14 on page 153. The OFFSET field is not used in this case.

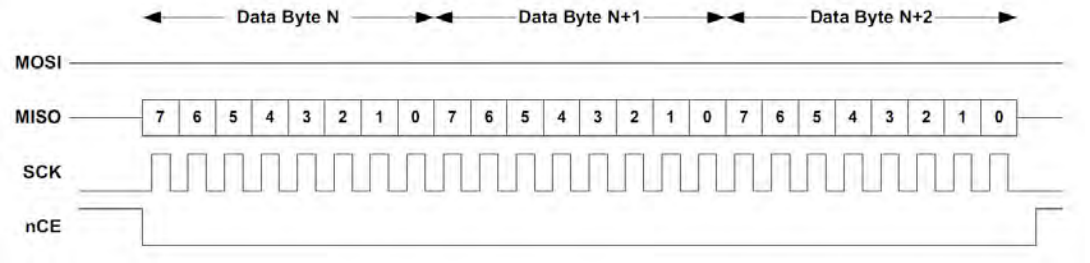
Figure 5-14: SPI Raw Write Operation



5.6.6 SPI Raw Read

If a Raw Read is selected in the OPER field, the operation is simply the data transfer portion of a Normal Read. All data goes directly to the FIFO as shown in Figure 5-15. The OFFSET field is not used in this case.

Figure 5-15: SPI Raw Read Operation



5.6.7 SPI 3-wire Mode

In 3-wire mode, the MOSI and MISO lines are shared on a single pin. As described in the previous sections, the MISO and MOSI lines are not driven at the same time, so 3-wire mode is equivalent to simply tying them together external to the Apollo2 SoC. 3-wire mode is configured by selecting the MxWIR3 alternative (x = 0 to 5 selecting the I^2C/SPI Master) in the GPIO Pad Multiplexor rather than the MxMOSI and MxMISO alternatives. Detailed configuration information is supplied in the *Section 8 GPIO and Pad Configuration Module on page 218*.

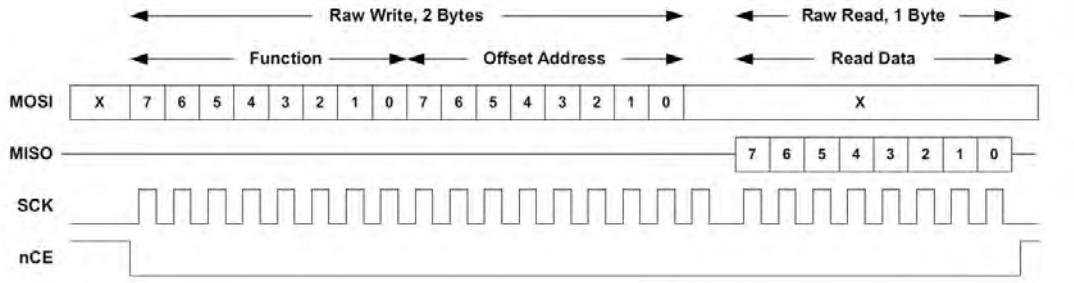
5.6.8 Complex SPI Operations

In some cases peripheral devices require more complex transaction sequences than those supported by a single Command. In order to support these transactions, the CONT bit may be set in the Command. In this case, the nCE pin selected by the Channel will remain asserted low at the end of the transaction, so that the next SPI operation will be seen as part of the same transaction. For example, there are peripheral devices which require both a Function and an Address Offset to be

transmitted at the beginning of a read. Implementing this can be done in several ways. One example as shown in Figure 5-16 is:

1. Execute a Raw SPI write of length 2, with the data bytes being the Function and Offset. Set the CONT bit in this Command so nCE remains asserted low.
2. Execute a Raw SPI Read of the desired transfer length. The data will then be read into the FIFO. The CONT bit is not set in this Command.

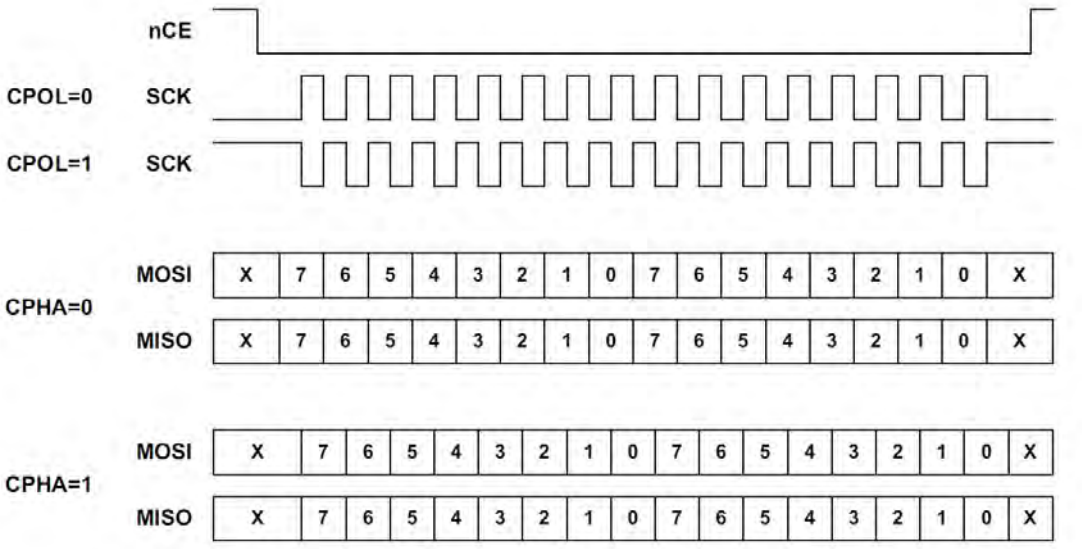
Figure 5-16: SPI Combined Operation



5.6.9 SPI Polarity and Phase

The Apollo2 SoC supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode, as defined by the SPOL and SPHA bits. Figure 5-17 shows how these two bits affect the interface signal behavior.

Figure 5-17: SPI CPOL and CPHA



If CPOL is 0, the clock SCK is normally low and positive pulses are generated during transfers. If CPOL is 1, SCK is normally high and negative pulses are generated during transfers.

If CPHA is 0, the data on the MOSI and MISO lines is sampled on the edge corresponding to the first SCK edge after nCE goes low (i.e. the rising edge if CPOL is 0 and the falling edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

If CPHA is 1, the data on the MOSI and MISO lines is sampled on the edge corresponding to the second SCK edge after nCE goes low (e.g., the falling edge if CPOL is 0 and the rising edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

The SPOL and SPHA bits may be changed between Commands if different slave devices have different requirements. In this case the bit should be set to 0 either before or at the same time as SPHA and SPOL are changed, and then set back to 1 before CMD is written.

5.7 Apollo2 SoC Bit Orientation

In both I^2C and SPI modes, the I^2C/SPI Master supports data transmission either LSB first or MSB first as configured by the LSB bit in the Command. If LSB is 0, data is transmitted and received MSB first. If LSB is 1, data is transmitted and received LSB first.

5.8 Full Duplex Operations

Some SPI slaves operate in full duplex mode, where data is transferred on both the MISO and MOSI wires at the same time. The I^2C/SPI Master supports this type of operation when the IOMSTRn_IOMCFG_FULLDUP bit is set.

When FULLDUP is set, the I^2C/SPI Master splits the standard 128-byte transmit/receive FIFO into a 64- byte transmit FIFO and a 64-byte receive FIFO. A normal or raw write Command is executed, and proceeds just as a normal write transfer. FIFOREM will report the remaining FIFO area as the remainder from 64 bytes. Software must not attempt to load more than 64 bytes into the write FIFO or an IACC error will be generated. The primary difference from a normal write operation is that data received on the MISO line will be loaded into the read FIFO, with the bytes aligned to the corresponding byte in the write FIFO. Software may read the read FIFO at any time, and should use the FIFOSIZ and./or FIFOREM registers to determine when data should be read. The FIFORTHR threshold value does not generate an interrupt.

If more than 64 bytes of data are written into the read FIFO without being read by software, the read FIFO will simply wrap around and overwrite the earlier read data. This means that if a long full duplex operation only returns data at the end, software does not need to continuously empty the read FIFO but can simply drain the FIFO when the write operation is complete.

5.9 SPI Flow Control

The I^{2}C/SPI Master supports flow control from the slave, which is controlled by several configuration bits. Either read or write (or both) flow control may be implemented. Read flow control is enabled by setting the IOMSTR_IOMCFG_RDFC bit, in which case the I^{2}C/SPI Master will check the state of the Flow Control IRQ pin, and if it is inactive the SPI clock will stop at the completion of the current byte transfer until it becomes active. The Flow Control IRQ can be any of the 50 pins as selected by the GPIO_IOMnIRQ register corresponding to the particular I^{2}C/SPI Master. The polarity of the active state of the Flow Control IRQ is selected by the IOMSTR_IOMCFGCFGn_RDFCPOL.

Write flow control is enabled by setting the IOMSTR_IOMCFGCFGn_WTFC bit, but in this case either the Flow Control IRQ or the state of the MISO line may be used for flow control, as selected by the IOMSTR_IOMCFGCFGn_WTFCIRQ bit. If IRQ is selected by setting a one, the clock control is identical to that described for reads above and the IRQ polarity is set by the IOMSTR_IOMCFGCFGn_WTFCPOL bit.

If MISO is selected by setting a zero in WTFCIRQ, the clock will be stopped if the MISO line is at the inactive polarity, which is set by the WTFCPOL bit.

Slave devices supporting flow control typically require specific states of the MOSI line prior to the start of a transfer. This state is controlled by the IOMSTR_IOMCFGCFGn_MOSIINV bit. If this bit is zero, MOSI will be driven to a 1 at the start of a write transaction and to a 0 at the start of a read transaction – this is the normal operation of devices with flow control support. If MOSIINV is set to one, these polarities will be inverted.

Flow control may be asserted either prior to the first byte transfer, which will delay the start of SCK, or within each byte transferred, which will pause SCK at the end of that byte. The examples below assume that WTFCPOL or RDFCPOL are set to 0.

Figure 5-18 shows the operation of flow control at the beginning of a write transfer or a normal read transfer which begins with an offset byte write. Either MISO or IRQ (selected by WTFCIRQ) must be deasserted low within ½ of the SCK period after nCE is asserted low in order to delay the clock. SCK will continue in its inactive state until MISO or IRQ is changed to the active state, and then will begin normal operation.

Figure 5-18: Flow Control at Beginning of a Write Transfer

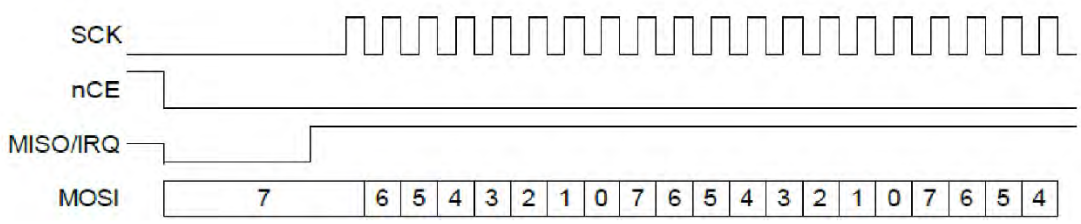


Figure 5-19 shows the operation of flow control at the beginning of a raw read transfer. IRQ must be deasserted low within ½ of the SCK period after nCE is asserted low in order to delay the clock. SCK will continue in its inactive state until IRQ is changed to the active state, and then will begin normal operation.

Figure 5-19: Flow Control at Beginning of a Raw Read Transfer

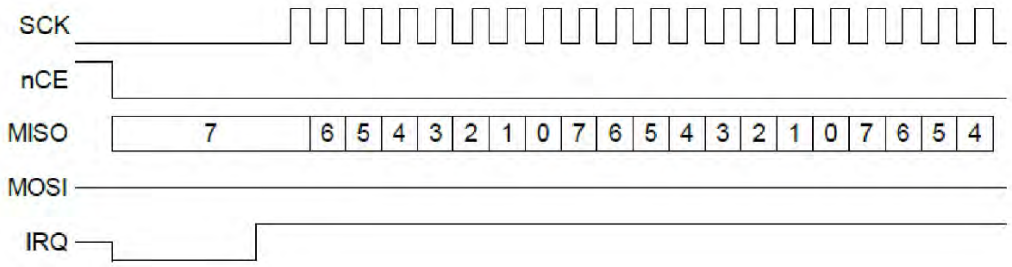


Figure 5-20 shows the operation of flow control in the middle of a write transfer. MISO or IRQ must be deasserted after the leading edge of SCK on the first bit of the byte (labeled 7) and before the falling edge of the 7th bit of the byte (labeled 1) in order to insure that SCK stops at the end of the byte. De-asserting MISO or IRQ outside of that window can produce unpredictable results. SCK will resume at some point after the assertion of MISO or IRQ.

Figure 5-20: Flow Control in the Middle of a Write Transfer

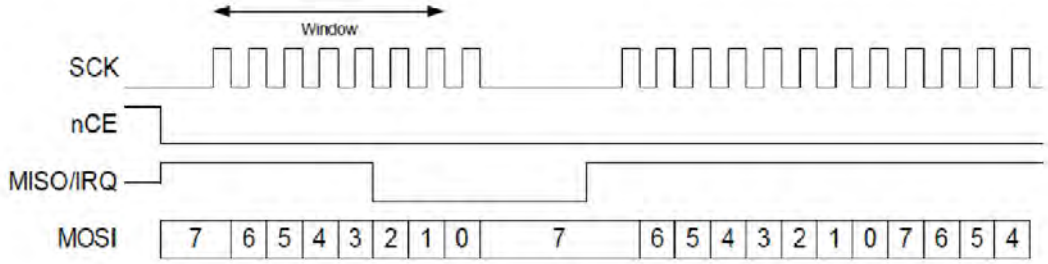
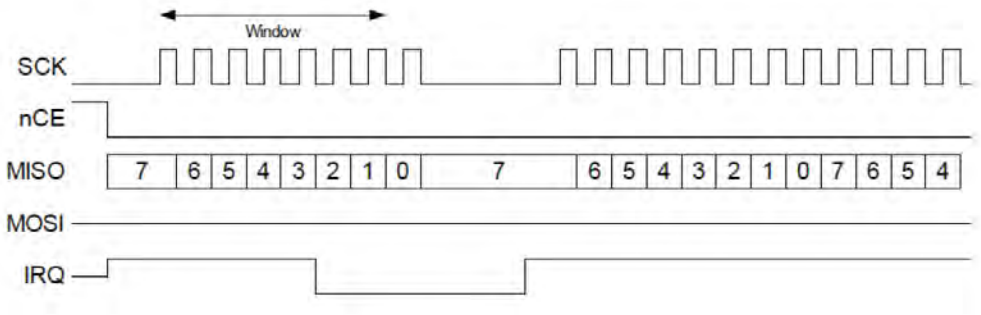


Figure 5-21 on page 158 shows the operation of flow control in the middle of a read transfer. IRQ must be deasserted after the leading edge of SCK on the first bit of the byte (labeled 7) and before the falling edge of the 7th bit of the byte (labeled 1) in order to insure that SCK stops at the end of the byte. De-asserting IRQ outside of that window can produce unpredictable results. SCK will resume at some point after the assertion of IRQ.

Figure 5-21: Flow Control in the Middle of a Read Transfer



5.10 Pre-read Control

The STARTRD field defines the number of bus clock cycles before the end of each byte where the IO read request occurs. For all I^2C frequencies and SPI frequencies below 16 MHz, the STARTRD field should be set to 0 to minimize the potential of the IO transfer holding off a bus access to the FIFO. For SPI frequencies of 16 MHz or 24 MHz, the STARTRD field must be set to a value of 2 to insure enough time for the IO preread.

5.11 Minimizing Power

Each I^2C/SPI Master has an interface enable bit IOMSTRn_IOMCFG_IFCEN. This bit should be kept at 0 whenever the interface is not being used in order to minimize power consumption. The FIFO cannot be accessed if IFCEN is 0, although all of the other registers are accessible.

5.12 IOMSTR Registers

I2C/SPI Master
INSTANCE 0 BASE ADDRESS:0x50004000
INSTANCE 1 BASE ADDRESS:0x50005000
INSTANCE 2 BASE ADDRESS:0x50006000
INSTANCE 3 BASE ADDRESS:0x50007000
INSTANCE 4 BASE ADDRESS:0x50008000
INSTANCE 5 BASE ADDRESS:0x50009000

5.12.1 Register Memory Map

Table 5-4: IOMSTR Register Map

Address(es)	Register Name	Description
0x50004000 0x50005000 0x50006000 0x50007000 0x50008000 0x50009000	FIFO	FIFO Access Port
0x50004100 0x50005100 0x50006100 0x50007100 0x50008100 0x50009100	FIFOPTR	Current FIFO Pointers
0x50004104 0x50005104 0x50006104 0x50007104 0x50008104 0x50009104	TLNGTH	Transfer Length
0x50004108 0x50005108 0x50006108 0x50007108 0x50008108 0x50009108	FIFOTHR	FIFO Threshold Configuration
0x5000410C 0x5000510C 0x5000610C 0x5000710C 0x5000810C 0x5000910C	CLKCFG	I/O Clock Configuration
0x50004110 0x50005110 0x50006110 0x50007110 0x50008110 0x50009110	CMD	Command Register
0x50004118 0x50005118 0x50006118 0x50007118 0x50008118 0x50009118	STATUS	Status Register

Table 5-4: IOMSTR Register Map *(Continued)*

Address(es)	Register Name	Description
0x5000411C 0x5000511C 0x5000611C 0x5000711C 0x5000811C 0x5000911C	CFG	I/O Master Configuration
0x50004200 0x50005200 0x50006200 0x50007200 0x50008200 0x50009200	INTEN	IO Master Interrupts: Enable
0x50004204 0x50005204 0x50006204 0x50007204 0x50008204 0x50009204	INTSTAT	IO Master Interrupts: Status
0x50004208 0x50005208 0x50006208 0x50007208 0x50008208 0x50009208	INTCLR	IO Master Interrupts: Clear
0x5000420C 0x5000520C 0x5000620C 0x5000720C 0x5000820C 0x5000920C	INTSET	IO Master Interrupts: Set

5.12.2 IOMSTR Registers

5.12.2.1 FIFO Register

FIFO Access Port
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x50004000
INSTANCE 1 ADDRESS: 0x50005000
INSTANCE 2 ADDRESS: 0x50006000
INSTANCE 3 ADDRESS: 0x50007000
INSTANCE 4 ADDRESS: 0x50008000
INSTANCE 5 ADDRESS: 0x50009000
FIFO Access Port

Table 5-5: FIFO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO																															

Table 5-6: FIFO Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFO	0x0	RW	FIFO access port.

5.12.2.2 FIFOPTR Register

Current FIFO Pointers
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x50004100
INSTANCE 1 ADDRESS: 0x50005100
INSTANCE 2 ADDRESS: 0x50006100
INSTANCE 3 ADDRESS: 0x50007100
INSTANCE 4 ADDRESS: 0x50008100
INSTANCE 5 ADDRESS: 0x50009100
Current FIFO Pointers

Table 5-7: FIFOPTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD								FIFOREM								RSVD								FIFOSIZ							

Table 5-8: FIFOPTR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:16	FIFOREM	0x0	RO	The number of bytes remaining in the FIFO (e.g., 128-FIFOSIZ if FULLDUP = 0 or 64-FIFOSIZ if FULL-DUP = 1).
15:8	RSVD	0x0	RO	RESERVED
7:0	FIFOSIZ	0x0	RO	The number of bytes currently in the FIFO.

5.12.2.3 TLNGTH Register

Transfer Length
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x50004104
INSTANCE 1 ADDRESS: 0x50005104
INSTANCE 2 ADDRESS: 0x50006104
INSTANCE 3 ADDRESS: 0x50007104
INSTANCE 4 ADDRESS: 0x50008104
INSTANCE 5 ADDRESS: 0x50009104
Transfer Length

Table 5-9: TLNGTH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																				TLNGTH											

Table 5-10: TLNGTH Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	RESERVED
11:0	TLNGTH	0x0	RO	Remaining transfer length.

5.12.2.4 FIFOTHR Register

FIFO Threshold Configuration
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x50004108
INSTANCE 1 ADDRESS: 0x50005108
INSTANCE 2 ADDRESS: 0x50006108
INSTANCE 3 ADDRESS: 0x50007108
INSTANCE 4 ADDRESS: 0x50008108
INSTANCE 5 ADDRESS: 0x50009108
FIFO Threshold Configuration

Table 5-11: FIFOTHR Register

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07	06 05 04 03 02 01 00
RSVD		FIFOWTHR	RSVD	FIFORTHR

Table 5-12: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
31:15	RSVD	0x0	RO	RESERVED
14:8	FIFOWTHR	0x0	RW	FIFO write threshold.
7	RSVD	0x0	RO	RESERVED
6:0	FIFORTHR	0x0	RW	FIFO read threshold.

5.12.2.5 CLKCFG Register

I/O Clock Configuration
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x5000410C
INSTANCE 1 ADDRESS: 0x5000510C
INSTANCE 2 ADDRESS: 0x5000610C
INSTANCE 3 ADDRESS: 0x5000710C
INSTANCE 4 ADDRESS: 0x5000810C
INSTANCE 5 ADDRESS: 0x5000910C
I/O Clock Configuration

Table 5-13: CLKCFG Register

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13	12	11	10 09 08	07 06 05 04 03 02 01 00
TOTPER	LOWPER	RSVD	DIVEN	DIV3	FSEL	RSVD

Table 5-14: CLKCFG Register Bits

Bit	Name	Reset	RW	Description
31:24	TOTPER	0x0	RW	Clock total count minus 1.
23:16	LOWPER	0x0	RW	Clock low count minus 1.
15:13	RSVD	0x0	RO	RESERVED
12	DIVEN	0x0	RW	Enable clock division by TOTPER. DIS = 0x0 - Disable TOTPER division. EN = 0x1 - Enable TOTPER division.
11	DIV3	0x0	RW	Enable divide by 3. DIS = 0x0 - Select divide by 1. EN = 0x1 - Select divide by 3.

Table 5-14: CLKCFG Register Bits

Bit	Name	Reset	RW	Description
10:8	FSEL	0x0	RW	Select the input clock frequency. MIN_PWR = 0x0 - Selects the minimum power clock. This setting should be used whenever the IOMSTR is not active. HFRC = 0x1 - Selects the HFRC as the input clock. HFRC_DIV2 = 0x2 - Selects the HFRC / 2 as the input clock. HFRC_DIV4 = 0x3 - Selects the HFRC / 4 as the input clock. HFRC_DIV8 = 0x4 - Selects the HFRC / 8 as the input clock. HFRC_DIV16 = 0x5 - Selects the HFRC / 16 as the input clock. HFRC_DIV32 = 0x6 - Selects the HFRC / 32 as the input clock. HFRC_DIV64 = 0x7 - Selects the HFRC / 64 as the input clock.
7:0	RSVD	0x0	RO	RESERVED

5.12.2.6 CMD Register

Command Register
OFFSET: 0x00000110
INSTANCE 0 ADDRESS: 0x50004110
INSTANCE 1 ADDRESS: 0x50005110
INSTANCE 2 ADDRESS: 0x50006110
INSTANCE 3 ADDRESS: 0x50007110
INSTANCE 4 ADDRESS: 0x50008110
INSTANCE 5 ADDRESS: 0x50009110
Command Register

Table 5-15: CMD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMD																															

Table 5-16: CMD Register Bits

Bit	Name	Reset	RW	Description
31:0	CMD	0x0	RW	This register holds the I/O Command.

5.12.2.7 STATUS Register

Status Register
OFFSET: 0x00000118
INSTANCE 0 ADDRESS: 0x50004118
INSTANCE 1 ADDRESS: 0x50005118
INSTANCE 2 ADDRESS: 0x50006118
INSTANCE 3 ADDRESS: 0x50007118
INSTANCE 4 ADDRESS: 0x50008118
INSTANCE 5 ADDRESS: 0x50009118
Status Register

Table 5-17: STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													IDLEST	CMDACT	ERR

Table 5-18: STATUS Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	IDLEST	0x0	RO	This bit indicates if the I/O state machine is IDLE. IDLE = 0x1 - The I/O state machine is in the idle state.
1	CMDACT	0x0	RO	This bit indicates if the I/O Command is active. ACTIVE = 0x1 - An I/O command is active.
0	ERR	0x0	RO	This bit indicates if an error interrupt has occurred. ERROR = 0x1 - An error has been indicated by the IOM.

5.12.2.8 CFG Register

I/O Master Configuration
OFFSET: 0x0000011C
INSTANCE 0 ADDRESS: 0x5000411C
INSTANCE 1 ADDRESS: 0x5000511C
INSTANCE 2 ADDRESS: 0x5000611C
INSTANCE 3 ADDRESS: 0x5000711C
INSTANCE 4 ADDRESS: 0x5000811C
INSTANCE 5 ADDRESS: 0x5000911C
I/O Master Configuration

Table 5-19: CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IFCEN	RSVD																RDFCPOL	WTFCPOL	WTFCIRQ	FCDEL	MOSIINV	RDFC	WTFC	RSVD		STARTRD		FULLDUP	SPHA	SPOL	IFCSEL

Table 5-20: CFG Register Bits

Bit	Name	Reset	RW	Description
31	IFCEN	0x0	RW	This bit enables the IO Master. DIS = 0x0 - Disable the IO Master. EN = 0x1 - Enable the IO Master.
30:15	RSVD	0x0	RO	RESERVED
14	RDFCPOL	0x0	RW	This bit selects the read flow control signal polarity. HIGH = 0x0 - Flow control signal high creates flow control. LOW = 0x1 - Flow control signal low creates flow control.

Table 5-20: CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
13	WTFCPOL	0x1	RW	This bit selects the write flow control signal polarity. HIGH = 0x0 - Flow control signal high creates flow control. LOW = 0x1 - Flow control signal low creates flow control.
12	WTFCIRQ	0x0	RW	This bit selects the write mode flow control signal. MISO = 0x0 - MISO is used as the write mode flow control signal. IRQ = 0x1 - IRQ is used as the write mode flow control signal.
11	FCDEL	0x0	RW	This bit must be left at the default value of 0.
10	MOSIINV	0x0	RW	This bit invewrts MOSI when flow control is enabled. NORMAL = 0x0 - MOSI is set to 0 in read mode and 1 in write mode. INVERT = 0x1 - MOSI is set to 1 in read mode and 0 in write mode.
9	RDFC	0x0	RW	This bit enables read mode flow control. DIS = 0x0 - Read mode flow control disabled. EN = 0x1 - Read mode flow control enabled.
8	WTFC	0x0	RW	This bit enables write mode flow control. DIS = 0x0 - Write mode flow control disabled. EN = 0x1 - Write mode flow control enabled.
7:6	RSVD	0x0	RO	RESERVED
5:4	STARTRD	0x0	RW	This bit selects the preread timing. PRERD0 = 0x0 - 0 read delay cycles. PRERD1 = 0x1 - 1 read delay cycles. PRERD2 = 0x2 - 2 read delay cycles. PRERD3 = 0x3 - 3 read delay cycles.
3	FULLDUP	0x0	RW	This bit selects full duplex mode. NORMAL = 0x0 - 128 byte FIFO in half duplex mode. FULLDUP = 0x1 - 64 byte FIFO in full duplex mode.
2	SPHA	0x0	RW	This bit selects SPI phase. SAMPLE_LEADING_EDGE = 0x0 - Sample on the leading (first) clock edge. SAMPLE_TRAILING_EDGE = 0x1 - Sample on the trailing (second) clock edge.
1	SPOL	0x0	RW	This bit selects SPI polarity. CLK_BASE_0 = 0x0 - The base value of the clock is 0. CLK_BASE_1 = 0x1 - The base value of the clock is 1.
0	IFCSEL	0x0	RW	This bit selects the I/O interface. I2C = 0x0 - Selects I2C interface for the I/O Master. SPI = 0x1 - Selects SPI interface for the I/O Master.

5.12.2.9 INTEN Register

IO Master Interrupts: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x50004200
INSTANCE 1 ADDRESS: 0x50005200
INSTANCE 2 ADDRESS: 0x50006200
INSTANCE 3 ADDRESS: 0x50007200
INSTANCE 4 ADDRESS: 0x50008200
INSTANCE 5 ADDRESS: 0x50009200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 5-21: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 5-22: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10	ARB	0x0	RW	This is the arbitration loss interrupt. This error occurs if another master collides with an IO Master transfer. Generally, the IOM started an operation but found SDA already low.
9	STOP	0x0	RW	This is the STOP command interrupt. A STOP bit was detected by the IOM.
8	START	0x0	RW	This is the START command interrupt. A START from another master was detected. Software must wait for a STOP before proceeding.
7	ICMD	0x0	RW	This is the illegal command interrupt. Software attempted to issue a CMD while another CMD was already in progress. Or an attempt was made to issue a non-zero-length write CMD with an empty FIFO.
6	IACC	0x0	RW	This is the illegal FIFO access interrupt. An attempt was made to read the FIFO during a write CMD. Or an attempt was made to write the FIFO on a read CMD.
5	WTLEN	0x0	RW	This is the WTLEN interrupt.
4	NAK	0x0	RW	This is the I^2C NAK interrupt. The expected ACK from the slave was not received by the IOM.
3	FOVFL	0x0	RW	This is the Write FIFO Overflow interrupt. An attempt was made to write the FIFO while it was full (e.g., while FIFOSIZ > 124).
2	FUNDFL	0x0	RW	This is the Read FIFO Underflow interrupt. An attempt was made to read FIFO when empty (e.g., while FIFOSIZ < 4).
1	THR	0x0	RW	This is the FIFO Threshold interrupt.
0	CMDCMP	0x0	RW	This is the Command Complete interrupt.

5.12.2.10 INTSTAT Register

IO Master Interrupts: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x50004204
INSTANCE 1 ADDRESS: 0x50005204
INSTANCE 2 ADDRESS: 0x50006204
INSTANCE 3 ADDRESS: 0x50007204
INSTANCE 4 ADDRESS: 0x50008204
INSTANCE 5 ADDRESS: 0x50009204
Read bits from this register to discover the cause of a recent interrupt.

Table 5-23: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 5-24: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10	ARB	0x0	RW	This is the arbitration loss interrupt. This error occurs if another master collides with an IO Master transfer. Generally, the IOM started an operation but found SDA already low.
9	STOP	0x0	RW	This is the STOP command interrupt. A STOP bit was detected by the IOM.
8	START	0x0	RW	This is the START command interrupt. A START from another master was detected. Software must wait for a STOP before proceeding.
7	ICMD	0x0	RW	This is the illegal command interrupt. Software attempted to issue a CMD while another CMD was already in progress. Or an attempt was made to issue a non-zero-length write CMD with an empty FIFO.
6	IACC	0x0	RW	This is the illegal FIFO access interrupt. An attempt was made to read the FIFO during a write CMD. Or an attempt was made to write the FIFO on a read CMD.
5	WTLEN	0x0	RW	This is the WTLEN interrupt.
4	NAK	0x0	RW	This is the I^2C NAK interrupt. The expected ACK from the slave was not received by the IOM.
3	FOVFL	0x0	RW	This is the Write FIFO Overflow interrupt. An attempt was made to write the FIFO while it was full (e.g., while FIFOSIZ > 124).
2	FUNDFL	0x0	RW	This is the Read FIFO Underflow interrupt. An attempt was made to read FIFO when empty (e.g., while FIFOSIZ < 4).
1	THR	0x0	RW	This is the FIFO Threshold interrupt.
0	CMDCMP	0x0	RW	This is the Command Complete interrupt.

5.12.2.11 INTCLR Register

IO Master Interrupts: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x50004208
INSTANCE 1 ADDRESS: 0x50005208
INSTANCE 2 ADDRESS: 0x50006208
INSTANCE 3 ADDRESS: 0x50007208
INSTANCE 4 ADDRESS: 0x50008208
INSTANCE 5 ADDRESS: 0x50009208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 5-25: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 5-26: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10	ARB	0x0	RW	This is the arbitration loss interrupt. This error occurs if another master collides with an IO Master transfer. Generally, the IOM started an operation but found SDA already low.
9	STOP	0x0	RW	This is the STOP command interrupt. A STOP bit was detected by the IOM.
8	START	0x0	RW	This is the START command interrupt. A START from another master was detected. Software must wait for a STOP before proceeding.
7	ICMD	0x0	RW	This is the illegal command interrupt. Software attempted to issue a CMD while another CMD was already in progress. Or an attempt was made to issue a non-zero-length write CMD with an empty FIFO.
6	IACC	0x0	RW	This is the illegal FIFO access interrupt. An attempt was made to read the FIFO during a write CMD. Or an attempt was made to write the FIFO on a read CMD.
5	WTLEN	0x0	RW	This is the WTLEN interrupt.
4	NAK	0x0	RW	This is the I^2C NAK interrupt. The expected ACK from the slave was not received by the IOM.
3	FOVFL	0x0	RW	This is the Write FIFO Overflow interrupt. An attempt was made to write the FIFO while it was full (e.g., while FIFOSIZ > 124).
2	FUNDFL	0x0	RW	This is the Read FIFO Underflow interrupt. An attempt was made to read FIFO when empty (e.g., while FIFOSIZ < 4).
1	THR	0x0	RW	This is the FIFO Threshold interrupt.
0	CMDCMP	0x0	RW	This is the Command Complete interrupt.

5.12.2.12 INTSET Register

IO Master Interrupts: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x5000420C
INSTANCE 1 ADDRESS: 0x5000520C
INSTANCE 2 ADDRESS: 0x5000620C
INSTANCE 3 ADDRESS: 0x5000720C
INSTANCE 4 ADDRESS: 0x5000820C
INSTANCE 5 ADDRESS: 0x5000920C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 5-27: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 5-28: INTSET Register Bits

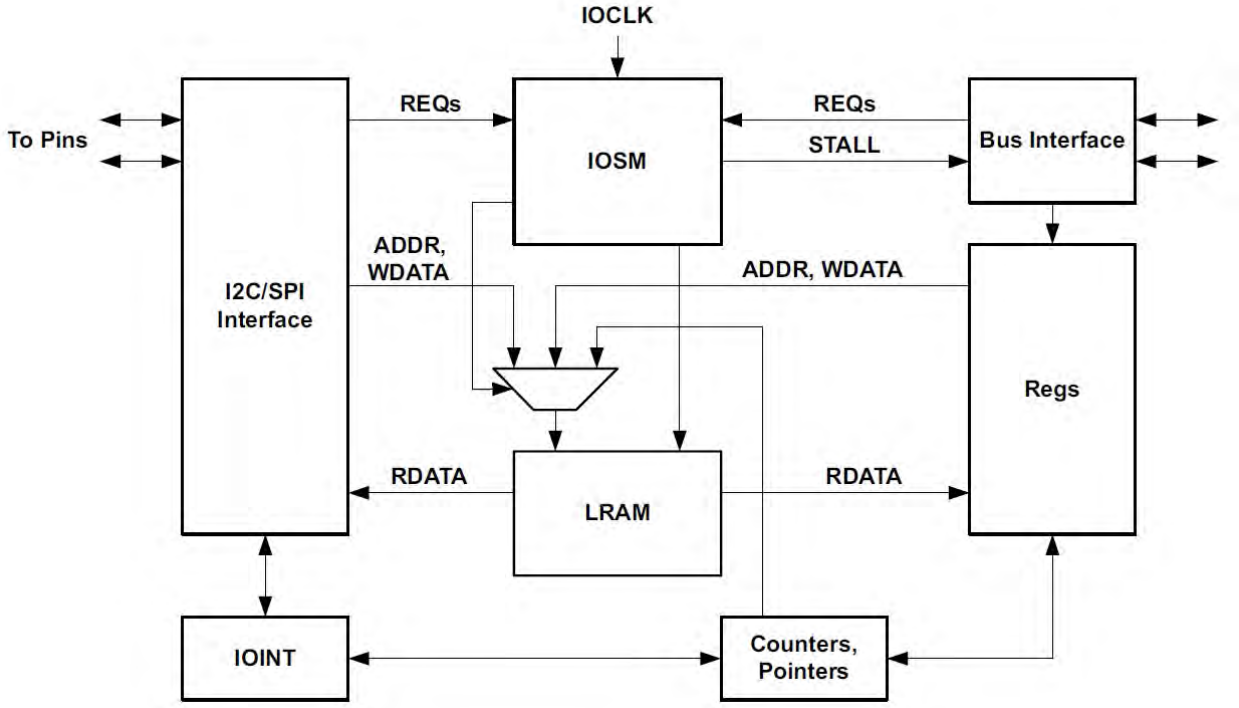
Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10	ARB	0x0	RW	This is the arbitration loss interrupt. This error occurs if another master collides with an IO Master transfer. Generally, the IOM started an operation but found SDA already low.
9	STOP	0x0	RW	This is the STOP command interrupt. A STOP bit was detected by the IOM.
8	START	0x0	RW	This is the START command interrupt. A START from another master was detected. Software must wait for a STOP before proceeding.
7	ICMD	0x0	RW	This is the illegal command interrupt. Software attempted to issue a CMD while another CMD was already in progress. Or an attempt was made to issue a non-zero-length write CMD with an empty FIFO.
6	IACC	0x0	RW	This is the illegal FIFO access interrupt. An attempt was made to read the FIFO during a write CMD. Or an attempt was made to write the FIFO on a read CMD.
5	WTLEN	0x0	RW	This is the WTLEN interrupt.
4	NAK	0x0	RW	This is the I^2C NAK interrupt. The expected ACK from the slave was not received by the IOM.
3	FOVFL	0x0	RW	This is the Write FIFO Overflow interrupt. An attempt was made to write the FIFO while it was full (e.g., while FIFOSIZ > 124).
2	FUNDFL	0x0	RW	This is the Read FIFO Underflow interrupt. An attempt was made to read FIFO when empty (e.g., while FIFOSIZ < 4).
1	THR	0x0	RW	This is the FIFO Threshold interrupt.
0	CMDCMP	0x0	RW	This is the Command Complete interrupt.

SECTION

6

I²C/SPI Slave Module

Figure 6-1: Block Diagram for the I²C/SPI Slave Module



6.1 Functional Overview

The I²C/SPI Slave (IOS) Module, shown in Figure 6-1, allows the Apollo2 SoC to function as a Slave in an I²C or SPI system. The I²C/SPI Slave operates in an independent fashion, so that the device may be placed in a sleep mode and still receive operations over the I/O interface. The Slave may be configured to generate an interrupt on specific references.

The I^{2}C/SPI Slave contains 256 bytes of RAM which is only accessible when the module is enabled. This RAM may be flexibly configured into three spaces: a block directly accessible via the I/O interface, a block which functions as a FIFO for read operations on the interface, and a block of generally accessible RAM used to store parameters during deep sleep mode.

In I^{2}C mode the Slave supports fully configurable 7 and 10-bit addressing with interface timing limits as specified in the Inter-Integrated Circuit (I^{2}C) Interface section of the Electricals chapter. In SPI mode, the Slave supports all polarity/phase combinations and interface frequencies as specified in the Serial Peripheral Interface (SPI) Slave Interface section.

6.2 Local RAM Allocation

The I^{2}C/SPI Slave is built around a 256-byte local RAM (LRAM), through which all data flows between the CPU AHB and the IO interface. The I^{2}C/SPI Slave supports a 128-byte offset space when accessed from the I/O interface.

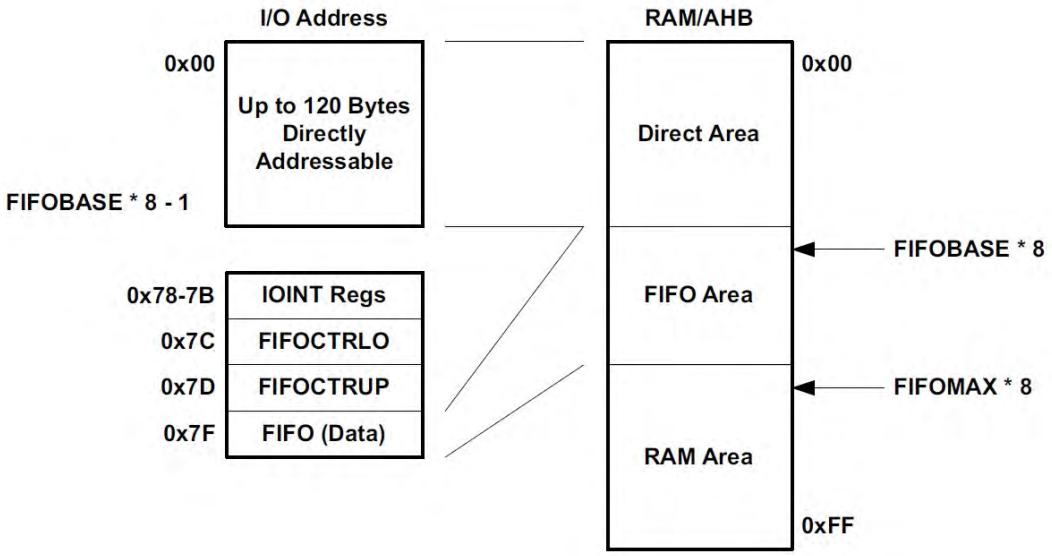
The LRAM is divided into three separate areas on 8-byte boundaries. These areas are:

1. A Direct Area for direct communication between the host and the MCU, which is mapped between the AHB address space and the I/O address space. This area is from LRAM address 0x00 to the address calculated from the 5-bit FIFOBASE field in the FIFO configuration register (FIFOCFG), minus 1. This 5-bit field (IOSLAVE_FIFOCFG_FIFOBASE) should contain a value that represents the start of the FIFO Area and, in so doing, defines the size of the Direct Area in 8-byte segments. Part of this area can be defined as IO Slave Read-only starting at any 8-byte segment defined by IOSLAVE_FIFOCFG_ROBASE and extending through the end of the Direct Area at FIFOBASE*8-1.
2. A FIFO Area which is used to stream data from the Apollo2 MCU. This memory is directly addressed from the AHB, but accessed from the I/O Interface using a single I/O address 0x7F as a streaming port. The FIFO area is from the LRAM address calculated from the value in the FIFOBASE field, FIFOBASE*8, to the LRAM address calculated from the value in the FIFOMAX field of the FIFOCFG register, IOSLAVE_FIFOCFG_FIFOMAX.The upper FIFO Area address is FIFOMAX*8-1. The maximum value for FIFOMAX is 0x20, which would result in an upper FIFO Area address of 0xFF.
3. A RAM Area which is accessible only from the AHB Slave. The RAM area is from the LRAM address calculated from the value in the FIFOMAX field of the FIFOCFG register, IOSLAVE_FIFOCFG_FIFOMAX, to address 0xFF. Setting FIFOMAX to 0x20 would result in a RAM area of zero size.

The data in the LRAM is maintained in Deep Sleep Mode.

Figure 6-2 below shows the LRAM address mapping between the I/O interface and the AHB.

Figure 6-2: I^{2}C/SPI Slave Module LRAM Addressing



6.3 Direct Area Functions

The Direct Area is used for direct communications between the interface Host and the Apollo2 SoC. The Host may write a register in this Register Access space, called REGACC, and read it back without requiring the CPU to wake up, so that very low power interactions are supported. In some cases, however, accesses require interaction with the CPU.

REGACC interrupts are mapped in the Direct Area and operate as follows. Each REGACC interrupt status bit will be set whenever there is a read or write over the I^{2}C or SPI interface in the Direct Area with an offset address which corresponds to a particular REGACC interrupt. Table 6-1 on page 174 below lists the offsets to memory locations within the Direct Area and corresponding interrupt bit settings in the REGACCINTSTAT register.

I/O writes to locations 0x0-0xF will set a corresponding interrupt flag in the REGACCINTSTAT register. These locations are typically used for specific commands to the Apollo2 SoC. Note that not all flags need generate an actual interrupt, so small multi-byte commands may be transmitted in this area. For example, a write to location 0x0 will set bit 31 of the REGACCINTSTAT register, a write to location 0x1 will set bit 30 of REGACCINTSTAT, and a write to location 0xF will set bit 16 of the REGACCINTSTAT register. The upper 16 REGACC interrupts are each generated on an access to the last byte of a 32-bit word, starting at 0x10.

I/O writes to locations 0x10 to 0x4F will set a corresponding interrupt flag in the REGACCINTSTAT register if the I/O address modulo 4 is 3 (e.g., addresses 0x13, 0x17, 0x1B, etc.). This allows larger transfers to be sent in a burst with a trigger being generated on the last write, and it also allows specifying a data buffer of any whole word size and have an interrupt generated on access to the last byte of the buffer. For example, a write to location 0x13 will set bit 15 of the REGACCINTSTAT register, a write to location 0x17 will set bit 14 of REGACCINTSTAT, and a write to location 0x4F will set bit 0 of the REGACCINTSTAT register.

Table 6-1 lists the offsets to memory locations within the Direct Address Space and corresponding interrupt bit settings in the REGACCINTSTAT register.

Table 6-1: Mapping of Direct Area Access Interrupts and Corresponding REGACCINTSTAT Bits

REGACCINTSTAT Bit	Direct Area Offset Address
31	0x0
30	0x1
29	0x2
28	0x3
27	0x4
26	0x5
25	0x6
24	0x7
23	0x8
22	0x9
21	0xA
20	0xB
19	0xC
18	0xD
17	0xE
16	0xF
15	0x13
14	0x17
13	0x1B
12	0x1F
11	0x23
10	0x27
9	0x2B
8	0x2F
7	0x33
6	0x37
5	0x3B
4	0x3F
3	0x43

Table 6-1: Mapping of Direct Area Access Interrupts and Corresponding REGACCINTSTAT Bits *(Continued)*

REGACCINTSTAT Bit	Direct Area Offset Address
2	0x47
1	0x4B
0	0x4F

The REGACCINTSTAT register provides status of the 32 individual write interrupts. If an interrupt is enabled and set, it shows as a high bit in this register. The highest priority REGACC bit is bit 31 (set on access to address 0x00), and the lowest priority is bit 0 (set on access to address 0x4F). The 5-bit IOSLAVE_PRENC register provides an encoded value of the highest priority of these interrupts to speed software decoding, and is therefore very useful for quickly servicing the highest priority REGACC interrupt (e.g., the one at the lowest offset address). The encoding works such that if interrupt 31 is set, PRENC will be 0. If interrupt 31 is not set and bit 30 is set, PRENC will be 1, and so on to the point where if bits 31-1 are not set and bit 0 is set PRENC will be 31. If no interrupts are set the value in PRENC is indeterminate.

The final special memory space within the Direct Area is a read-only area for the I/O Host, which is from I/O address (IOSLAVE_FIFOCFG_ROBASE * 8) to (FIFOBASE * 8 – 1). I/O writes to this address space will not change the LRAM, which allows the space to be used for returning status to the I/O Host. ROBASE should have a minimum value of 0x0A, representing a start address of 0x50 to allow space for special commands and burst writes in lower Direct Area space.

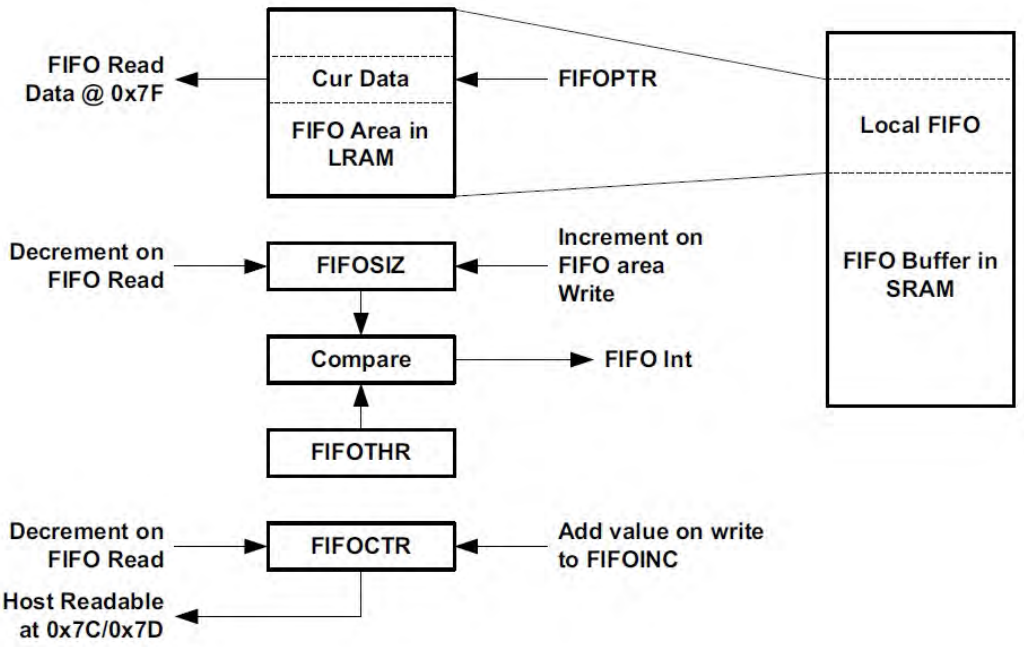
6.4 FIFO Area Functions

The FIFO is used to provide very efficient flow of data from the Apollo2 MCU to the I/O Host processor with minimal CPU interaction. A FIFO of up to 1023 bytes can be easily maintained by software, with the oldest bytes residing in the LRAM FIFO Area and the newer data being held in system SRAM and transferred to the I^2C/SPI Slave on demand. Several hardware features support this operation.

Figure 6-3 on page 176 shows the basic FIFO operation. The main FIFO is held in a buffer in SRAM, and the oldest data in that FIFO has been transferred to the FIFO Area of the I/O Slave. The IOSLAVE_FIFOPTR_FIFOPTR register points to the next byte to be read on the I/O interface. IOSLAVE_FIFOPTR_FIFOSIZ holds the current number of valid bytes in the FIFO on the I^2C/SPI Slave, and FIFOCTR holds the total number of bytes in the FIFO. The value in IOSLAVE_FIFOCTR may be read indirectly at any time by the Host processor via the FIFOCTRUP_FIFOCTRLO registers to determine if there is FIFO data available (and how much is currently in the FIFO). I/O Host access to the FIFO counter is at offset 0x7C/D.

WARNING: The host read of the FIFOCTR value via FIFOCTRUP_FIFOCTRLO is not synchronized to the write clock. So if the host read happens during a FIFOCTR update (either through a read-modify-write of FIFOCTR register or an automatic update because of a write to the FIFOINC register by the Slave CPU), it is possible for the count value to be out of sync, impacting the value read in either or both the upper (FIFOCTRUP) and lower (FIFOCTRLO) bytes. This is a very rare case, but proper code would have the host read the two registers for the FIFOCTR value multiple times until consecutive reads are the same.

Figure 6-3: I^{2}C/SPI Slave Module FIFO



When the host reads a byte from the FIFO, the data retrieved is pointed to by FIFOPTR, FIFOPTR is incremented and wraps around in the FIFO Area if it reaches FIFOMAX. FIFOSIZ and FIFOCTR are each decremented by one. The Host can read FIFOCTR and then read that many bytes without further checking. Note that this process can continue without requiring a CPU wakeup. If the Host attempts to read the FIFO when FIFOSIZ is 0, the FUNDFL interrupt flag is set in both the I^{2}C Slave interrupt block and in the Host interrupt block.

When FIFOSIZ drops below the configured threshold IOSLAVE_FIFOTHR the FSIZE interrupt flag is set and if enabled an interrupt is sent to the CPU which will wake it up. At that point, the CPU can move as much data from the SRAM FIFO to the I^{2}C/SPI Slave FIFO as possible in a single operation and then go back to sleep. Since the FIFO Area can be quite large, CPU wake-ups will be very infrequent. If a write to the

FIFOCTR which would increment the value beyond 1023 occurs, the FOVFL interrupt flag is set.

When some other process, such as a sensor read, produces new data for the FIFO, the CPU will add that data to the FIFO in SRAM, wrapping around as necessary. The IOSLAVE_FIFOINC register is then written with the number of bytes added to the FIFO, which is added to the FIFOCTR register in an atomic fashion. In this way the Host processor can always determine how much read data is available.

The FIFO interface offset 0x7F is treated uniquely by the I^2C/SPI Slave, in that an access to this address does not increment the Address Pointer. This allows the Host to initiate a burst read from address 0x7F of any length, and each read will supply the next byte in the FIFO.

6.5 Rearranging the FIFO

In normal operation the Host reads the oldest data from the FIFO, and the CPU writes new data onto the FIFO. In some cases it is desirable to modify this process, in particular for the FIFO to provide the newest data. The Apollo2 SoC supports such operation using a special control function.

If software desires to write the current sample to the front of the FIFO, it first checks the IOSLAVE_FUPD_IOREAD status bit to ensure that there is not a Host read operation from the FIFO underway. Once IOREAD is clear, software sets the IOSLAVE_-FUPD_FIFOUPD bit, writes the new sample data to the front of the FIFO and modifies the FIFOPTR to point to the new data. At that point the FIFOUPD bit is cleared.

If the Host attempts a FIFO read operation while the FIFOUPD is set, a RDERR interrupt will be generated to the Host and the FRDERR interrupt flag will be set. The Host must either poll the RDERR interrupt bit at the end of each operation or configure a hardware interrupt. Note that if the software does not support alternate FIFO ordering, the Host does not have to check the RDERR function.

6.6 Interface Interrupts

The CPU may also signal the Host via the IOINT interrupt, which may be connected to an Apollo2 SoC pin and driven to the Host. Eight interrupts are available to be combined into the IOINT interrupt, and the Host can enable, read, clear and set these interrupts via the I/O interface. Software on the CPU can set 6 of the interrupts (SWINT0 through SWINT5) to communicate a variety of situations to the Host, and the other two interrupts indicate errors such as an attempt by the Host to read the FIFO when it is empty. A CPU interrupt is generated whenever the Host writes any IOINT registers (for example, to clear an interrupt) so the CPU can manage the interrupt interaction.The I^2C/SPI Slave includes a mechanism to allow the Host CPU

and the Apollo2 SoC to each interrupt the other via a set of eight interrupts. The Host CPU accesses these interrupts via interface locations 0x78-0x7B, and the Apollo accesses these interrupts in the IOINTCTL Register.

The Host CPU may enable or disable any of the eight interrupts by writing the corresponding bit in the IOINTEN field of the IOINTCTL Register, which is accessed by the Host at interface location 0x78. The Host CPU may then clear or set any of the interrupts by writing a 1 to the corresponding bit of the clear (at location 0x7A) or set (at location 0x7B) registers. The current state of all eight interrupts may be read in the IOINT field at location 0x79. Note that this structure is identical to the standard Apollo2 MCU interrupts in all modules. The SoC can read the value of the eight interrupt enables in the IOINTEN field of IOINTCTL, and can read the values of the eight interrupt status bits in the IOINT field of the IOINTCTL register. These two fields are read only. Table 6-2 summarizes these I/O interface interrupts and how they can be controlled and read.

Table 6-2: I/O Interface Interrupt Control

RAM Location	IOINT Register[1]	Function	SoC Register_Field	Description
0x78	IOINTEN	I/O Interrupt Enable	IOINTCTL_IOINTEN (R/O)	Each interrupt can be individually enabled by I/O Host, but can only be read by the SoC
0x79	IOINT	I/O Interrupt State	IOINTCTL_IOINT (R/O)	State of each interrupt, set or cleared, can be read by either the I/O Host or by the SoC.
0x7A	IOINTCLR	I/O Interrupt Clear	IOINTCTL_IOINTCLR (W/O)	Each interrupt can be individually cleared by the I/O Host, but the SoC can (only) clear all of them at once.
0x7B	IOINTSET	I/O Interrupt Set	IOINTCTL_IOINTSET (W/O)	Each interrupt can be individually set by either the I/O Host or the SoC.

[1] Readable by the I/O Host.

The Apollo2 SoC software may set any of the eight interrupt status register bits by writing a 1 to the corresponding bit of the IOINTSET field of the IOINTCTL Register, and may clear all of the interrupts by writing a 1 to the IOINTCLR bit of the IOINTCTL register. This allows the SoC to generate a software interrupt to the Host device. In addition, a FIFO underflow interrupt FUNDFL in the I^2C/SPI Slave will set interrupt bit 7, and a FIFO read error interrupt FRDERR will set interrupt bit 6 of the IO interrupt status register IOINT. Note that the SoC software cannot write the IOINTEN register, so that IO interrupts are controlled completely by the Host processor.

If any of the IOINT interrupt bits are set and the corresponding bit in IOINTEN is set, an IOINT interrupt will be generated. If the GPIO configuration registers have configured PAD4 as IOINT, that interrupt will be driven directly onto PAD_IO[4]. This pin should be connected to an interrupt input pin of the Host interface device so that it can receive the interrupt and service it.

If the Host device writes to any of the interrupt register access locations (any location in 0x78-0x7B) the IOINTW interrupt will be set in the I^{2}C/SPI INTSTAT Register. This allows Apollo2 SoC software to receive a software interrupt from the Host device. Note that this interrupt will occur for all writes by the Host, including a write to clear an interrupt.

6.7 Command Completion Interrupts

Four interrupts in the I^{2}C/SPI Slave module are generated when the Host interface device completes a transfer. This allows the SoC to be easily awakened for any transfer from the Host while maximizing the time the SoC is in sleep mode. The XCMPWR interrupt is generated at the completion of a Host write transfer which includes addresses in the currently configured Direct Register space, and the XCMPRR interrupt is generated on the completion of a Host read transfer to that space. The XCMPWF interrupt is generated at the completion of a Host write transfer which includes the FIFO address 0x7F (although that is an invalid access), and the XCMPRF interrupt is generated at the completion of a Host read transfer which includes the FIFO address 0x7F.

NOTES:

- A write to 0x7F, which is the FIFO address, uses the address 0xFF since this includes the R/W bit in the upper (first) bit followed by the 7-bit Direct Register address (offset). The prescribed usage of IOS FIFO is only for READ from the host, and hence writing to the FIFO is generally an invalid operation. So, even though XCMPWF flag/interrupt is defined, it is likely never going to be used.
- A burst transfer which begins in the Direct Register address space and is long enough to cause the Address Pointer to be 0x7F can set both the Direct Register and FIFO interrupts, although that would in general be an invalid operation.

6.8 Host Address Space and Registers

The Host of the I/O interface can access 128 bytes in the I^{2}C/SPI Slave in either I^{2}C or SPI mode. Offsets 0x00 to 0x77 may be directly mapped to the Direct RAM Area. The remaining eight offset locations access hardware functions within the I^{2}C/SPI Slave. The R/W indicator is referring to accesses from the Host.

6.9 I²C Interface

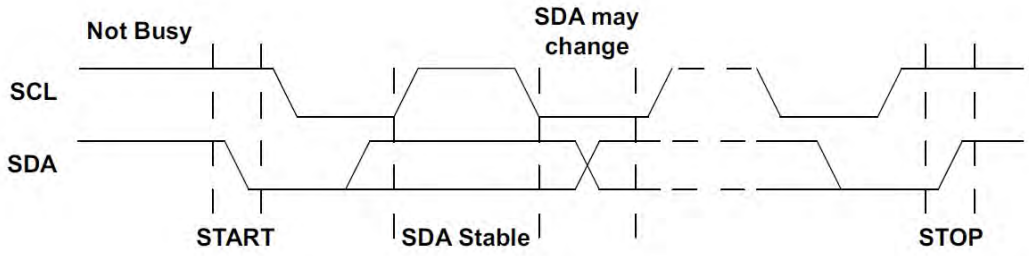
The Apollo2 SoC I²C Slave interface operates as a standard slave. The device is accessed at an address configured in the IOSLAVE_IOSCFG_I2CADDR field, and supports Fast Mode Plus (up to 1 MHz). Both 7-bit and 10-bit address modes are supported, as selected by IOSLAVE_IOSCFG_10BIT. The I²C interface consists of two lines: one bi-directional data line (SDA) and one clock line (SCL). Both the SDA and the SCL lines must be connected to a positive supply voltage via a pull-up resistor. By definition, a device that sends a message is called the "transmitter", and the device that accepts the message is called the "receiver". The device that controls the message transfer by driving SCL is called "master". The devices that are controlled by the master are called "slaves". The SoC's I²C Slave is always a slave device.

The following protocol has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high.
- Changes in the data line while the clock line is high will be interpreted as control signals.

A number of bus conditions have been defined (see Figure 6-4) and are described in the following sections.

Figure 6-4: Basic I²C Conditions



6.9.1 Bus Not Busy

Both SDA and SCL remain high.

6.9.2 Start Data Transfer

A change in the state of SDA from high to low, while SCL is high, defines the START condition. A START condition which occurs after a previous START but before a STOP is called a RESTART condition, and functions exactly like a normal STOP followed by a normal START.

6.9.3 Stop Data Transfer

A change in the state of SDA from low to high, while SCL is high, defines the STOP condition.

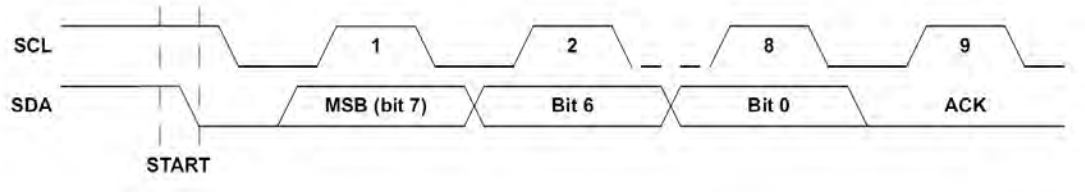
6.9.4 Data Valid

After a START condition, SDA is stable for the duration of the high period of SCL. The data on SDA may be changed during the low period of SCL. There is one clock pulse per bit of data. Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and STOP conditions is not limited. The information is transmitted bytewide and each receiver acknowledges with a ninth bit.

6.9.5 Acknowledge

Each byte of eight bits is followed by one Acknowledge (ACK) bit as shown in Figure 6-5. This Acknowledge bit is a low level driven onto SDA by the receiver, whereas the master generates an extra ACK related SCL pulse. A slave receiver which is addressed is obliged to generate an Acknowledge after the reception of each byte. Also, on a read transfer a master receiver must generate an Acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges must pull down the SDA line during the Acknowledge clock pulse in such a way that the SDA line is a stable low during the high period of the Acknowledge related SCL pulse. A master receiver must signal an end-of-data to the slave transmitter by not generating an Acknowledge (a NAK) on the last byte that has been clocked out of the slave. In this case, the transmitter must leave the data line high to enable the master to generate the STOP condition.

Figure 6-5: I^2C Acknowledge



6.9.6 Address Operation

In I^2C mode, the I^2C/SPI Slave supports either 7-bit or 10-bit addressing, selected by the 10BIT bit in the IOSCFG Register. Figure 6-6 on page 182 shows the operation in 7-bit mode in which the master addresses the Apollo2 SoC with a 7-bit address configured as 0xD2 in the CFG_I2CADDR field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of the CFG_I2CADDR field, the SoC is selected, the eighth bit indicate a write (RW = 0)

or a read (RW = 1) operation and the SoC supplies the ACK. The SoC ignores all other address values and does not respond with an ACK.

Figure 6-6: I2C 7-bit Address Operation

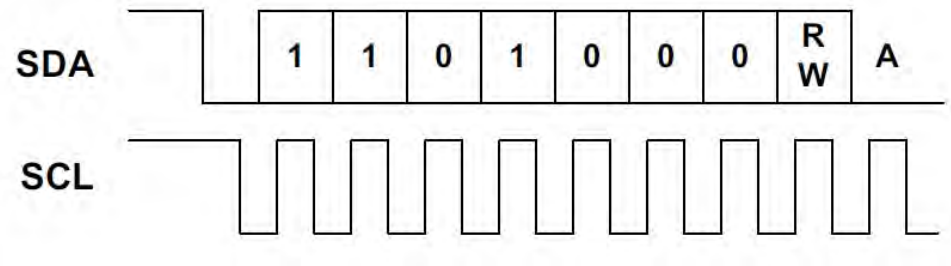
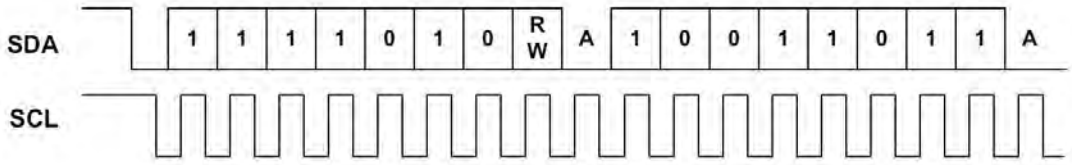


Figure 6-7 shows the operation with which the master addresses the SoC with a 10-bit address configured at 0x536. After the START condition, the 10-bit preamble 0b11110 is transmitted first, followed by the first two address bits and the eighth bit indicating a write (RW = 0) or a read (RW = 1) operation. If the upper two bits match the I2CADDR value, the I2C/SPI Slave supplies the ACK. The next transfer includes the lower 8 bits of the address, and if these bits also match I2CADDR the SoC again supplies the ACK. The I2C/SPI Slave ignores all other address values and does not respond with an ACK.

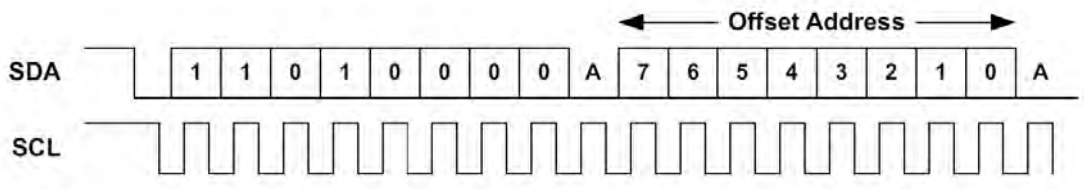
Figure 6-7: I2C 10-bit Address Operation



6.9.7 Offset Address Transmission

If the RW bit of the Address Operation indicates a write, the next byte transmitted from the master is the Offset Address as shown in Figure 6-8. This value is loaded into the Address Pointer of the I2C/SPI Slave.

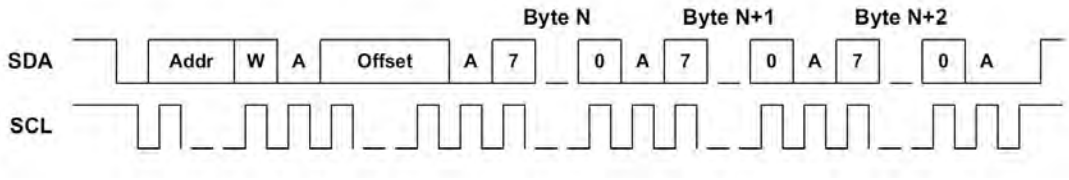
Figure 6-8: I2C Offset Address Transmission



6.9.8 Write Operation

In a write operation the master transmitter transmits to the Apollo2 SoC slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address as in Figure 6-8 on page 182. The next byte is written to the register selected by the Address Pointer (which was loaded with the Offset Address) and the Address Pointer is incremented. Subsequent transfers write bytes into successive registers until a STOP condition is received, as shown in Figure 6-9. Note that if the Address Pointer is at 0x7F, it will not increment on the write.

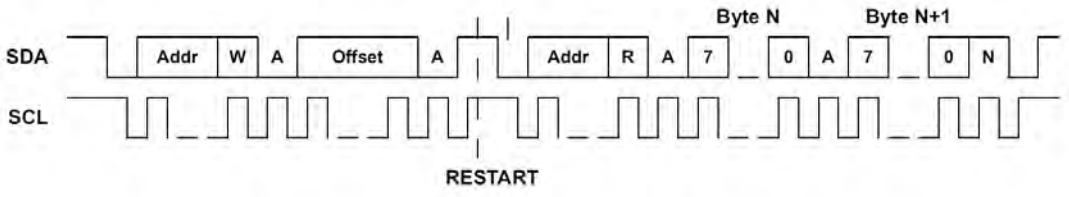
Figure 6-9: I^{2}C Write Operation



6.9.9 Read Operation

In a read operation, the master first executes an Offset Address Transmission to load the Address Pointer with the desired Offset Address. A subsequent operation will again issue the address of the SoC but with the RW bit as a 1 indicating a read operation. Figure 6-10 shows this transaction beginning with a RESTART condition, although a STOP followed by a START may also be used. After the address operation, the slave becomes the transmitter and sends the register value from the location pointed to by the Address Pointer, and the Address Pointer is incremented. Subsequent transactions produce successive register values, until the master receiver responds with a NAK and a STOP to complete the operation. Because the Address Pointer holds a valid register address, the master may initiate another read sequence at this point without performing another Offset Address operation. Note that if the Address Pointer is at 0x7F, it will not increment on the read.

Figure 6-10: I^{2}C Read Operation



6.9.10 General Address Detection

The I^{2}C/SPI Slave may be configured to detect an I^{2}C General Address (0x00) write. If this address is detected, the first data byte written is stored in the IOSLAVE_GA-

DATA Register and the GENAD interrupt flag is set. This allows software to create the appropriate response, which is typically to reset the I^{2}C/SPI Slave.

6.10 SPI Interface

The I^{2}C/SPI Slave includes a standard 3-wire or 4-wire SPI interface. The serial peripheral interface (SPI) bus is intended for synchronous communication between different ICs. 4-wire SPI consists of four signal lines: serial data input (MOSI), serial data output (MISO), serial clock (SCL) and an active low chip enable (nCE). The I^{2}C/SPI Slave may be connected to a master with a 3-wire SPI interface by configuring 3-wire mode in the pin configuration block of the GPIO module, which will tie MOSI and MISO together. By definition, a device that sends a message is called the “transmitter”, and the device that accepts the message is called the “receiver”. The device that controls the message transfer by driving SCL is called “master”. The devices that are controlled by the master are called “slaves”. The I^{2}C/SPI Slave SPI Slave is always a slave device.

The nCE input is used to initiate and terminate a data transfer. The SCL input is used to synchronize data transfer between the master and the slave devices via the MOSI (master to slave) and MISO (slave to master) lines. The SCL input, which is generated by the master, is active only during address and data transfer to any device on the SPI bus.

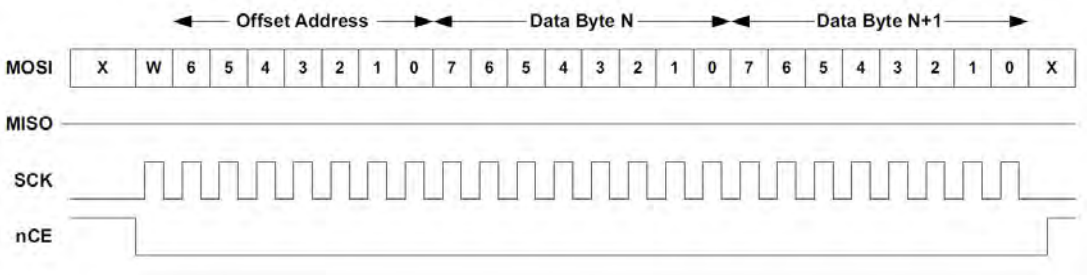
The I^{2}C/SPI Slave supports all SPI configurations of CPOL and CPHA using the SPOL configuration bit. There is one clock for each bit transferred. Address and data bits are transferred in groups of eight bits.

6.10.1 Write Operation

Figure 6-11 on page 185 shows a SPI write operation. The operation is initiated when the nCE signal to the SoC goes low. At that point an 8-bit Address byte is transmitted from the master on the MOSI line, with the upper RW bit indicating read (if 0) or write (if 1). In this example the RW bit is a one selecting a write operation, and the lower 7 bits of the Address byte contain the Offset Address, which is loaded into the Address Pointer of the I^{2}C/SPI Slave.

Each subsequent byte is loaded into the register selected by the Address Pointer, and the Address Pointer is incremented. The operation is terminated by the master by bringing the nCE signal high. Note that the MISO line is not used in a write operation and is held in the high impedance state by the I^{2}C/SPI Slave. Note also that if the Address Pointer is 0x7F, it does not increment on the read.

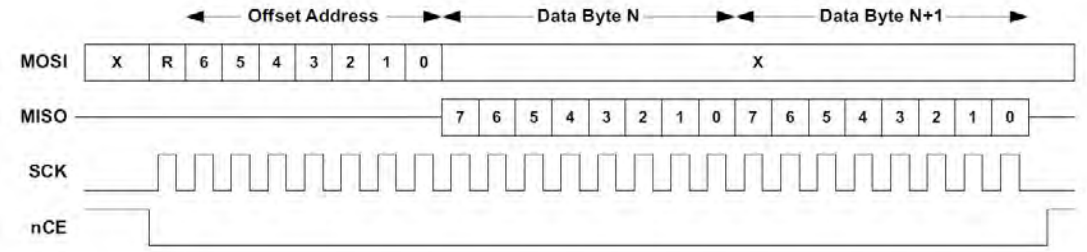
Figure 6-11: SPI White Operation



6.10.2 Read Operation

Figure 6-12 shows a read operation. The address is transferred from the master to the slave just as it is in a write operation, but in this case the RW bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the I^{2}C/SPI Slave begins driving data from the register selected by the Address Pointer onto the MISO line, bit 7 first, and the Address Pointer is incremented. The transfer continues until the master brings the nCE line high. Note that if the Address Pointer is 0x7F, it does not increment on the read.

Figure 6-12: SPI Read Operation



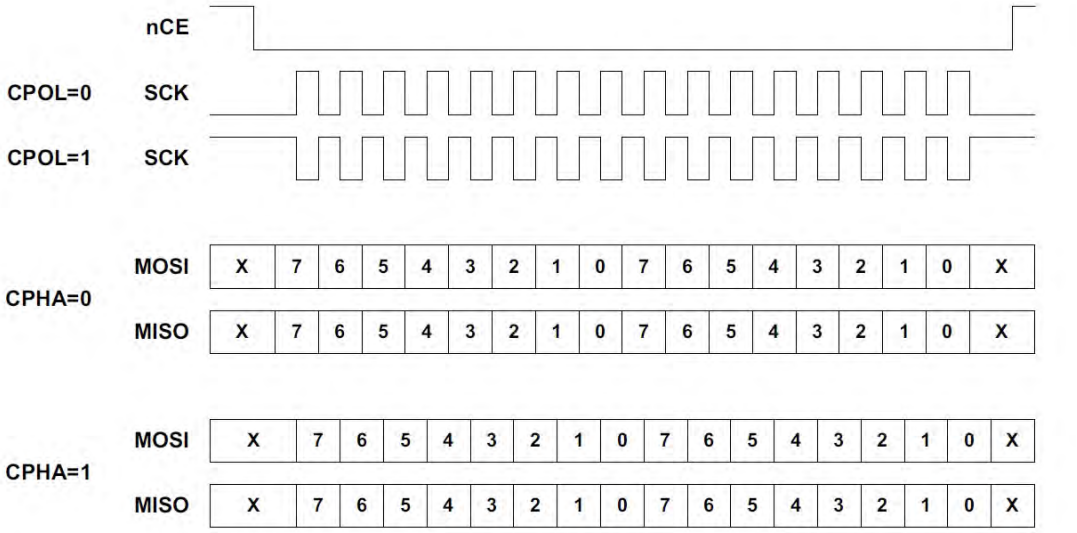
6.10.3 Configuring 3-wire vs. 4-wire SPI Mode

The I^{2}C/SPI Slave can operate in either 4-wire SPI mode, where the MISO and MOSI signals are on separate wires, or in 3-wire SPI mode where MISO and MOSI share a wire. This configuration is performed in the Pin Configuration module, and no configuration is necessary in the I^{2}C/SPI Slave itself.

6.10.4 SPI Polarity and Phase

The I^{2}C/SPI Slave supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode. Figure 6-13 on page 186 shows how these two bits affect the interface signal behavior.

Figure 6-13: SPI CPOL and CPHA



If CPOL is 0, the clock SCK is normally low and positive pulses are generated during transfers. If CPOL is 1, SCK is normally high and negative pulses are generated during transfers.

If CPHA is 0, the data on the MOSI and MISO lines is sampled on the edge corresponding to the first SCK edge after nCE goes low (i.e. the rising edge if CPOL is 0 and the falling edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

If CPHA is 1, the data on the MOSI and MISO lines is sampled on the edge corresponding to the second SCK edge after nCE goes low (i.e. the falling edge if CPOL is 0 and the rising edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

The I^2C/SPI Slave has only a single SPOL bit to control the polarity. If CPOL = CPHA, IOSLAVE_IOSCFG_SPOL must be set to 0. If CPOL ≠ CPHA, SPOL must be set to 1.

6.11 Bit Orientation

In both I^2C and SPI modes, the I^2C/SPI Slave supports data transmission either LSB first or MSB first as configured by the IOSLAVE_IOSCFG_LSB bit. If LSB is 0, data is transmitted and received MSB first. If LSB is 1, data is transmitted and received LSB first.

6.12 Wakeup Using the I²C/SPI Slave

The I²C/SPI Slave can continue to operate even if the SoC's CPU is in Sleep or Deep Sleep mode. The hardware will enable and disable the I²C/SPI Slave clock and oscillators as necessary. The only consideration in this environment is when the SoC is in a deep sleep mode, such that the HFRC Oscillator is powered down, and a master attempts to access the I²C/SPI Slave. In this case the HFRC Oscillator must be powered up before any is transferred to or from the internal RAM. This process takes roughly 5-10 us, and is initiated by nCE going low in SPI mode or by the detection of a START in I²C mode.

For I²C applications, the time delay is typically not relevant. At the fastest system clock of 1 MHz, the master must transfer 9 bits of address plus 9 bits of offset before any FIFO access can occur, and that is a minimum of 18 us. The clocks will have started prior to that point in every case.

For SPI applications with fast interface clocks (faster than 1 MHz), the master must be programmed to pull nCE low at least 10 us prior to sending the first clock. If a master is unable to control the timing of nCE in this way, then a GPIO interrupt can be configured to wake the MCU prior to initiating any SPI transfers.

There is no delay restriction if the SoC is in normal Sleep mode. In that case the HFRC is not powered down and the I²C/SPI Slave clock will start immediately when nCE goes low. Alternatively, the FRCHFRC bit may be set in the FRCHFRC Register in the CLK_GEN module. If this bit is set, the HFRC will continue to be active even if the SoC's CPU is in deep sleep mode, so that the I²C/SPI Slave can immediately begin transferring data independent of the SPI transfer rate. This will result in higher power because the HFRC remains active, so the FRCHFRC bit should only be set if it is known that a transfer is likely to begin prior to another interrupt.

6.13 IOSLAVE Registers

I2C/SPI Slave
INSTANCE 0 BASE ADDRESS:0x50000000

6.13.1 Register Memory Map

Table 6-3: IOSLAVE Register Map

Address(es)	Registered Name	Description
0x50000100	FIFOPTR	Current FIFO Pointer
0x50000104	FIFOCFG	FIFO Configuration
0x50000108	FIFOTHR	FIFO Threshold Configuration
0x5000010C	FUPD	FIFO Update Status
0x50000110	FIFOCTR	Overall FIFO Counter
0x50000114	FIFOINC	Overall FIFO Counter Increment
0x50000118	CFG	I/O Slave Configuration
0x5000011C	PRENC	I/O Slave Interrupt Priority Encode
0x50000120	IOINTCTL	I/O Interrupt Control
0x50000124	GENADD	General Address Data
0x50000200	INTEN	IO Slave Interrupts: Enable
0x50000204	INTSTAT	IO Slave Interrupts: Status
0x50000208	INTCLR	IO Slave Interrupts: Clear
0x5000020C	INTSET	IO Slave Interrupts: Set
0x50000210	REGACCINTEN	Register Access Interrupts: Enable
0x50000214	REGACCINTSTAT	Register Access Interrupts: Status
0x50000218	REGACCINTCLR	Register Access Interrupts: Clear
0x5000021C	REGACCINTSET	Register Access Interrupts: Set

6.13.2 IOSLAVE Registers

6.13.2.1 FIFOPTR Register

Current FIFO Pointer
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x50000100
Current FIFO Pointer

Table 6-4: FIFOPTR Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
RSVD	FIFOSIZ	FIFOPTR

Table 6-5: FIFOPTR Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:8	FIFOSIZ	0x0	RW	The number of bytes currently in the hardware FIFO.
7:0	FIFOPTR	0x0	RW	Current FIFO pointer.

6.13.2.2 FIFOCFG Register

FIFO Configuration
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x50000104
FIFO Configuration

Table 6-6: FIFOCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	ROBASE							RSVD								RSVD	FIFOMAX							RSVD	FIFOBASE						

Table 6-7: FIFOCFG Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:24	ROBASE	0x20	RW	Defines the read-only area. The IO Slave read-only area is situated in LRAM at (ROBASE*8) to (FIFOOBASE*8-1)
23:16	RSVD	0x0	RO	RESERVED
15:14	RSVD	0x0	RO	RESERVED
13:8	FIFOMAX	0x0	RW	These bits hold the maximum FIFO address in 8 byte segments. It is also the beginning of the RAM area of the LRAM. Note that no RAM area is configured if FIFOMAX is set to 0x1F.
7:5	RSVD	0x0	RO	RESERVED
4:0	FIFOBASE	0x0	RW	These bits hold the base address of the I/O FIFO in 8 byte segments.The IO Slave FIFO is situated in LRAM at (FIFOBASE*8) to (FIFOMAX*8-1).

6.13.2.3 FIFOTHR Register

FIFO Threshold Configuration
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x50000108
FIFO Threshold Configuration

Table 6-8: FIFOTHR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								FIFOTHR							

Table 6-9: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:0	FIFOTHR	0x0	RW	FIFO size interrupt threshold.

6.13.2.4 FUPD Register

FIFO Update Status
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x5000010C
FIFO Update Status

Table 6-10: FUPD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														IOREAD	FIFOUPD

Table 6-11: FUPD Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	IOREAD	0x0	RO	This bit field indicates an IO read is active.
0	FIFOUPD	0x0	RW	This bit indicates that a FIFO update is underway.

6.13.2.5 FIFOCTR Register

Overall FIFO Counter
OFFSET: 0x00000110
INSTANCE 0 ADDRESS: 0x50000110
Overall FIFO Counter

Table 6-12: FIFOCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								FIFOCTR							

Table 6-13: FIFOCTR Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9:0	FIFOCTR	0x0	RW	Virtual FIFO byte count.

6.13.2.6 FIFOINC Register

Overall FIFO Counter Increment
OFFSET: 0x00000114
INSTANCE 0 ADDRESS: 0x50000114
Overall FIFO Counter Increment

Table 6-14: FIFOINC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						FIFOINC									

Table 6-15: FIFOINC Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9:0	FIFOINC	0x0	WO	Increment the Overall FIFO Counter by this value on a write.

6.13.2.7 CFG Register

I/O Slave Configuration
OFFSET: 0x00000118
INSTANCE 0 ADDRESS: 0x50000118
I/O Slave Configuration

Table 6-16: CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IFCEN	RSVD											I2CADDR												RSVD			STARTRD	RSVD	LSB	SPOL	IFCSEL

Table 6-17: CFG Register Bits

Bit	Name	Reset	RW	Description
31	IFCEN	0x0	RW	IOSLAVE interface enable. DIS = 0x0 - Disable the IOSLAVE EN = 0x1 - Enable the IOSLAVE
30:20	RSVD	0x0	RO	RESERVED

Table 6-17: CFG Register Bits

Bit	Name	Reset	RW	Description
19:8	I2CADDR	0x0	RW	7-bit or 10-bit I2C device address. Bit 19 Selects 7-bit/10-bit address: 7BIT = 0x0, 10BIT = 0x1 Bit 18:16 Upper 3 bits for 10-bit address; not used for 7-bit address Bit 15:9 7-bit address or lower 7 bits of 10-bit address Bit 8 Not used for either address mode
7:5	RSVD	0x0	RO	RESERVED
4	STARTRD	0x0	RW	This bit holds the cycle to initiate an I/O RAM read. LATE = 0x0 - Initiate I/O RAM read late in each transferred byte. EARLY = 0x1 - Initiate I/O RAM read early in each transferred byte.
3	RSVD	0x0	RO	RESERVED
2	LSB	0x0	RW	This bit selects the transfer bit ordering. MSB_FIRST = 0x0 - Data is assumed to be sent and received with MSB first. LSB_FIRST = 0x1 - Data is assumed to be sent and received with LSB first.
1	SPOL	0x0	RW	This bit selects SPI polarity. SPI_MODES_0_3 = 0x0 - Polarity 0, handles SPI modes 0 and 3. SPI_MODES_1_2 = 0x1 - Polarity 1, handles SPI modes 1 and 2.
0	IFCSEL	0x0	RW	This bit selects the I/O interface. I2C = 0x0 - Selects I^2C interface for the IO Slave. SPI = 0x1 - Selects SPI interface for the IO Slave.

6.13.2.8 PRENC Register

I/O Slave Interrupt Priority Encode
OFFSET: 0x0000011C
INSTANCE 0 ADDRESS: 0x5000011C
I/O Slave Interrupt Priority Encode

Table 6-18: PRENC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																											PRENC				

Table 6-19: PRENC Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4:0	PRENC	0x0	RO	These bits hold the priority encode of the REGACC interrupts.

6.13.2.9 IOINTCTL Register

I/O Interrupt Control
OFFSET: 0x00000120
INSTANCE 0 ADDRESS: 0x50000120
I/O Interrupt Control

Table 6-20: IOINTCTL Register

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17	16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
IOINTSET	RSVD	IOINTCLR	IOINT	IOINTEN

Table 6-21: IOINTCTL Register Bits

Bit	Name	Reset	RW	Description
31:24	IOINTSET	0x0	WO	These bits set the IOINT interrupts when written with a 1.
23:17	RSVD	0x0	RO	RESERVED
16	IOINTCLR	0x0	WO	This bit clears all of the IOINT interrupts when written with a 1.
15:8	IOINT	0x0	RO	These bits read the IOINT interrupts.
7:0	IOINTEN	0x0	RO	These read-only bits indicate whether the IOINT interrupts are enabled.

6.13.2.10 GENADD Register

General Address Data
OFFSET: 0x00000124
INSTANCE 0 ADDRESS: 0x50000124
General Address Data

Table 6-22: GENADD Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
RSVD	GADATA

Table 6-23: GENADD Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:0	GADATA	0x0	RO	The data supplied on the last General Address reference.

6.13.2.11 INTEN Register

IO Slave Interrupts: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x50000200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 6-24: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						XCMPWR	XCMPWF	XCMPRR	XCMPRF	IOINTW	GENAD	FRDERR	FUNDFL	FOVFL	FSIZE

Table 6-25: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9	XCMPWR	0x0	RW	Transfer complete interrupt, write to register space.
8	XCMPWF	0x0	RW	Transfer complete interrupt, write to FIFO space.
7	XCMPRR	0x0	RW	Transfer complete interrupt, read from register space.
6	XCMPRF	0x0	RW	Transfer complete interrupt, read from FIFO space.
5	IOINTW	0x0	RW	IO Write interrupt.
4	GENAD	0x0	RW	I^2C General Address interrupt.
3	FRDERR	0x0	RW	FIFO Read Error interrupt.
2	FUNDFL	0x0	RW	FIFO Underflow interrupt.
1	FOVFL	0x0	RW	FIFO Overflow interrupt.
0	FSIZE	0x0	RW	FIFO Size interrupt.

6.13.2.12 INTSTAT Register

IO Slave Interrupts: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x50000204
Read bits from this register to discover the cause of a recent interrupt.

Table 6-26: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						XCMPWR	XCMPWF	XCMPRR	XCMPRF	IOINTW	GENAD	FRDERR	FUNDFL	FOVFL	FSIZE

Table 6-27: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9	XCMPWR	0x0	RW	Transfer complete interrupt, write to register space.
8	XCMPWF	0x0	RW	Transfer complete interrupt, write to FIFO space.
7	XCMPRR	0x0	RW	Transfer complete interrupt, read from register space.
6	XCMPRF	0x0	RW	Transfer complete interrupt, read from FIFO space.
5	IOINTW	0x0	RW	IO Write interrupt.

Table 6-27: INTSTAT Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
4	GENAD	0x0	RW	I^2C General Address interrupt.
3	FRDERR	0x0	RW	FIFO Read Error interrupt.
2	FUNDFL	0x0	RW	FIFO Underflow interrupt.
1	FOVFL	0x0	RW	FIFO Overflow interrupt.
0	FSIZE	0x0	RW	FIFO Size interrupt.

6.13.2.13 INTCLR Register

IO Slave Interrupts: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x50000208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 6-28: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						XCMPWR	XCMPWF	XCMPRR	XCMPRF	IOINTW	GENAD	FRDERR	FUNDFL	FOVFL	FSIZE

Table 6-29: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9	XCMPWR	0x0	RW	Transfer complete interrupt, write to register space.
8	XCMPWF	0x0	RW	Transfer complete interrupt, write to FIFO space.
7	XCMPRR	0x0	RW	Transfer complete interrupt, read from register space.
6	XCMPRF	0x0	RW	Transfer complete interrupt, read from FIFO space.
5	IOINTW	0x0	RW	IO Write interrupt.
4	GENAD	0x0	RW	I^2C General Address interrupt.
3	FRDERR	0x0	RW	FIFO Read Error interrupt.
2	FUNDFL	0x0	RW	FIFO Underflow interrupt.
1	FOVFL	0x0	RW	FIFO Overflow interrupt.
0	FSIZE	0x0	RW	FIFO Size interrupt.

6.13.2.14 INTSET Register

IO Slave Interrupts: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x5000020C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 6-30: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						XCMPWR	XCMPWF	XCMPRR	XCMPRF	IOINTW	GENAD	FRDERR	FUNDFL	FOVFL	FSIZE

Table 6-31: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9	XCMPWR	0x0	RW	Transfer complete interrupt, write to register space.
8	XCMPWF	0x0	RW	Transfer complete interrupt, write to FIFO space.
7	XCMPRR	0x0	RW	Transfer complete interrupt, read from register space.
6	XCMPRF	0x0	RW	Transfer complete interrupt, read from FIFO space.
5	IOINTW	0x0	RW	IO Write interrupt.
4	GENAD	0x0	RW	I^2C General Address interrupt.
3	FRDERR	0x0	RW	FIFO Read Error interrupt.
2	FUNDFL	0x0	RW	FIFO Underflow interrupt.
1	FOVFL	0x0	RW	FIFO Overflow interrupt.
0	FSIZE	0x0	RW	FIFO Size interrupt.

6.13.2.15 REGACCINTEN Register

Register Access Interrupts: Enable
OFFSET: 0x00000210
INSTANCE 0 ADDRESS: 0x50000210
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 6-32: REGACCINTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
REGACC																															

Table 6-33: REGACCINTEN Register Bits

Bit	Name	Reset	RW	Description
31:0	REGACC	0x0	RW	Register access interrupts.

6.13.2.16 REGACCINTSTAT Register

Register Access Interrupts: Status
OFFSET: 0x00000214
INSTANCE 0 ADDRESS: 0x50000214

Read bits from this register to discover the cause of a recent interrupt.

Table 6-34: REGACCINTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
REGACC																															

Table 6-35: REGACCINTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:0	REGACC	0x0	RW	Register access interrupts.

6.13.2.17 REGACCINTCLR Register

Register Access Interrupts: Clear
OFFSET: 0x00000218
INSTANCE 0 ADDRESS: 0x50000218
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 6-36: REGACCINTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
REGACC																															

Table 6-37: REGACCINTCLR Register Bits

Bit	Name	Reset	RW	Description
31:0	REGACC	0x0	RW	Register access interrupts.

6.13.2.18 REGACCINTSET Register

Register Access Interrupts: Set
OFFSET: 0x0000021C
INSTANCE 0 ADDRESS: 0x5000021C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 6-38: REGACCINTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
REGACC																															

Table 6-39: REGACCINTSET Register Bits

Bit	Name	Reset	RW	Description
31:0	REGACC	0x0	RW	Register access interrupts.

6.14 Host Side Address Space and Register

6.14.1 Host Address Space and Registers

The Host of the I/O interface can access 128 bytes in the I^2C/SPI Slave in either I^2C or SPI mode. Offsets 0x00 to 0x77 may be directly mapped to the Direct RAM Area. The remaining eight offset locations access hardware functions within the I^2C/SPI Slave. The R/W indicator refers to accesses from the Host.

6.14.1.1 HOST_IER Register

Host Interrupt Enable
OFFSET: 0x78
This register enables the FIFO read interrupts.

Table 6-40: HOST_IER Register

07	06	05	04	03	02	01	00
FUNDFLEN	RDERREN	SWINT5EN	SWINT4EN	SWINT3EN	SWINT2EN	SWINT1EN	SWINT0EN

Table 6-41: HOST_IER Register Bits

Bit	Name	Reset	RW	Description
7	FUNDFLEN	0x0	RW	If 1, enable an interrupt that triggers when the FIFO underflows
6	RDERREN	0x0	RW	If 1, enable the interrupt which occurs when the Host attempts to access the FIFO when read access is locked
5	SWINT5EN	0x0	RW	If 1, enable software interrupt 5
4	SWINT4EN	0x0	RW	If 1, enable software interrupt 4
3	SWINT3EN	0x0	RW	If 1, enable software interrupt 3
2	SWINT2EN	0x0	RW	If 1, enable software interrupt 2
1	SWINT1EN	0x0	RW	If 1, enable software interrupt 1
0	SWINT0EN	0x0	RW	If 1, enable software interrupt 0

6.14.1.2 HOST_ISR Register

Host Interrupt Status Register
OFFSET: 0x79
The host uses this register to read interrupt status.

Table 6-42: HOST_ISR Register

07	06	05	04	03	02	01	00
FUNDFLSTAT	RDERRSTAT	SWINT5STAT	SWINT4STAT	SWINT3STAT	SWINT2STAT	SWINT1STAT	SWINT0STAT

Table 6-43: HOST_ISR Register Bits

Bit	Name	Reset	RW	Description
7	FUNDFLSTAT	0x0	RO	This bit is set by writing a 1 to bit 31 of the IOINTCTL Register, or if the Host attempts a FIFO read when FIFOCTR is 0.
6	RDERRSTAT	0x0	RO	This bit is set by writing a 1 to bit 30 of the IOINTCTL Register, or if the Host attempts a FIFO read when the FIFOUPD bit is a 1.
5	SWINT5STAT	0x0	RO	This bit is set by writing a 1 to bit 29 of the IOINTCTL Register.
4	SWINT4STAT	0x0	RO	This bit is set by writing a 1 to bit 28 of the IOINTCTL Register.
3	SWINT3STAT	0x0	RO	This bit is set by writing a 1 to bit 27 of the IOINTCTL Register.
2	SWINT2STAT	0x0	RO	This bit is set by writing a 1 to bit 26 of the IOINTCTL Register.
1	SWINT1STAT	0x0	RO	This bit is set by writing a 1 to bit 25 of the IOINTCTL Register.
0	SWINT0STAT	0x0	RO	This bit is set by writing a 1 to bit 24 of the IOINTCTL Register.

NOTE: All bits are cleared by a write to the IOINTCLR bit of the IOINTCTL Register.

6.14.1.3 HOST_WCR Register

Host Interrupt Write-to-Clear Register
OFFSET: 0x7A
Write a 1 to a bit in this register to clear a pending interrupt.

Table 6-44: HOST_WCR Register

07	06	05	04	03	02	01	00
FUNDFLWC	RDERRWC	SWINT5WC	SWINT4WC	SWINT3WC	SWINT2WC	SWINT1WC	SWINT0WC

Table 6-45: HOST_WCR Register Bits

Bit	Name	Reset	RW	Description
7	FUNDFLWC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit FUNDFLSTAT
6	RDERRWC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit RDERRSTAT
5	SWINT5WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT5STAT
4	SWINT4WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT4STAT

Table 6-45: HOST_WCR Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
3	SWINT3WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT3STAT
2	SWINT2WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT2STAT
1	SWINT1WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT1STAT
0	SWINT0WC	0x0	WO	Writing a 1 to this bit will clear the pending interrupt status bit SWINT0STAT

6.14.1.4 HOST_WCS Register

Host Interrupt Write-to-Set Register
OFFSET: 0x7B
Write a 1 to a bit in this register to set the status bit of a pending interrupt.

Table 6-46: HOST_WCS Register

07	06	05	04	03	02	01	00
FUNDFLWS	RDERRWS	SWINT5WS	SWINT4WS	SWINT3WS	SWINT2WS	SWINT1WS	SWINT0WS

Table 6-47: HOST_WCS Register Bits

Bit	Name	Reset	RW	Description
7	FUNDFLWS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit FUNDFLSTAT
6	RDERRWS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit RDERRSTAT
5	SWINT5WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT5STAT
4	SWINT4WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT4STAT
3	SWINT3WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT3STAT
2	SWINT2WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT2STAT
1	SWINT1WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT1STAT
0	SWINT0WS	0x0	WO	Writing a 1 to this bit will set the pending interrupt status bit SWINT0STAT

6.14.1.5 FIFOCTRLO Register

FIFOCTR Low Byte
OFFSET: 0x7C
This register allows the host to read the lower eight bits of the FIFOCTR register.

Table 6-48: FIFOCTRLO Register

07	06	05	04	03	02	01	00
FIFOCTRLO							

Table 6-49: FIFOCTRLO Register Bits

Bit	Name	Reset	RW	Description
7:0	FIFOCTRLO	0x0	RO	Reads the lower eight bits of FIFOCTR.

6.14.1.6 FIFOCTRUP Register

FIFOCTR Upper Byte
OFFSET: 0x7D
This register allows the host to read the upper two bits of the FIFOCTR register.

Table 6-50: FIFOCTRUP Register

07	06	05	04	03	02	01	00
RSVD						FIFOCTRUP	

Table 6-51: FIFOCTRUP Register Bits

Bit	Name	Reset	RW	Description
1:0	FIFOCTRUP	0x0	RO	Reads the upper two bits of FIFOCTR

6.14.1.7 FIFO Register

FIFO Read Data
OFFSET: 0x7F
Read this register for FIFO data.

Table 6-52: FIFO Register

07	06	05	04	03	02	01	00
FIFO							

Table 6-53: FIFO Register Bits

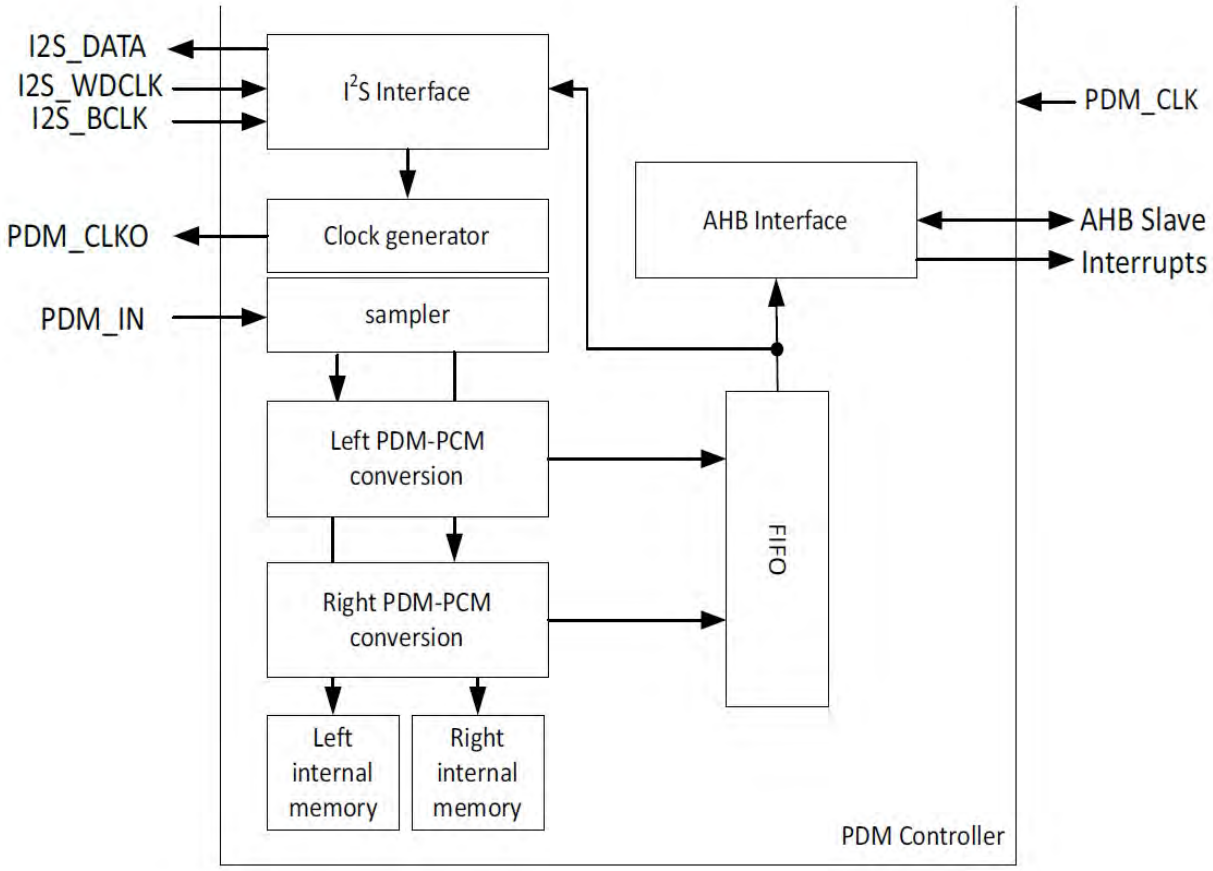
Bit	Name	Reset	RW	Description
7:0	FIFO	0x0	RO	Reads the top byte of the FIFO

SECTION

7

PDM/I²S Module

Figure 7-1: Block Diagram for PDM Module



7.1 Features

The PDM module provides support for low power Pulse-Density Modulated (PDM) to Pulse-Code Modulated (PCM) conversion and optional I^2S slave interface for external host processor communication.

The PDM controller generates the clock output to interface to 1 (mono) or 2 (stereo) PDM-based digital microphones. The PDM input data is sampled on the rising (left/mono) and falling (right/stereo) edges of PDM clock. The controller supports 16-bit PCM output sampling at 8/16 KHz. The single bit pulse-density modulated (PDM) bit stream data is converted into pulse-code modulated (PCM) data and provides an optional I^2S serial audio/voice data format. The converted PCM data is stored in an asynchronous FIFO where is can then be retrieved by the SoC CPU via the AHB slave interface.

The PDM controller includes the following features:

- Stereo or mono PDM input
- 16-bit PCM digital output
- I^2S slave interface output (optional)
- Support for variable PDM output clock rates (750-768 KHz, 1.5-1.536 MHz, 3-3.072 MHz: output clock depends on source clock from I^2S or SoC)
- 64x decimation of PDM bit stream input to PCM output
- Sampling rate: 8 KHz, 16 KHz (additional sample rates are supported as needed)
- AHB slave interface for register control, status programming and PCM FIFO data access

7.2 Functional Overview

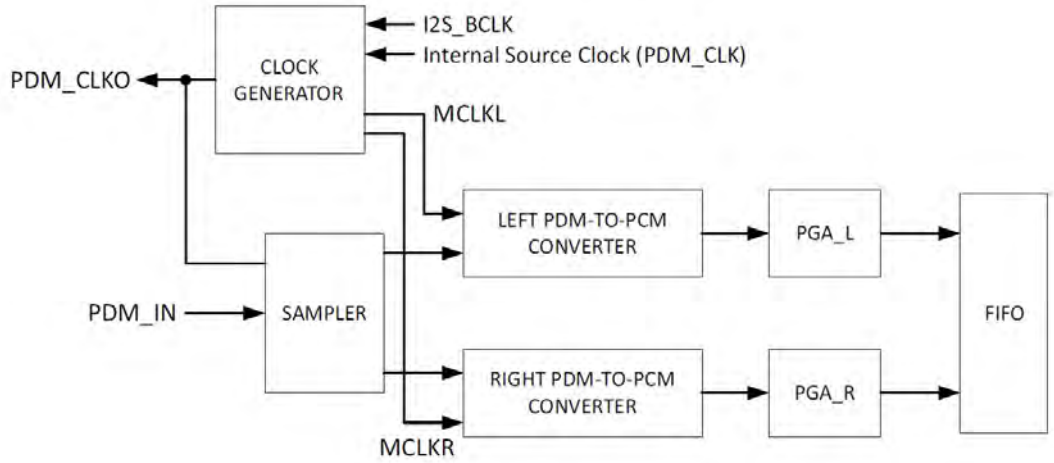
The Apollo2 SoC integrates a PDM controller which has two modes of operation: low power mode and normal mode. The low power mode is intended for wake-on-voice/keyword detect operation. A low frequency PDM clock is generated to the microphone (requires digital microphone that supports low power operation). Once a keyword is detected, the SoC generates a wake event to enter normal mode. In normal mode, higher PDM frequencies are supported to process audio/voice as needed for voice recording, voice calls, etc.

7.2.1 PDM-to-PCM Conversion

The PDM-to-PCM core IP converts PDM bit stream data into 16-bit PCM data through internal data sampling, filtering, and PGA amplification. The controller may be operated at stereo or mono mode in normal operation, system reset or power down mode when not in use. Each mode can be programmed through registers.

The basic PCM conversion flow is shown in Figure 7-2.

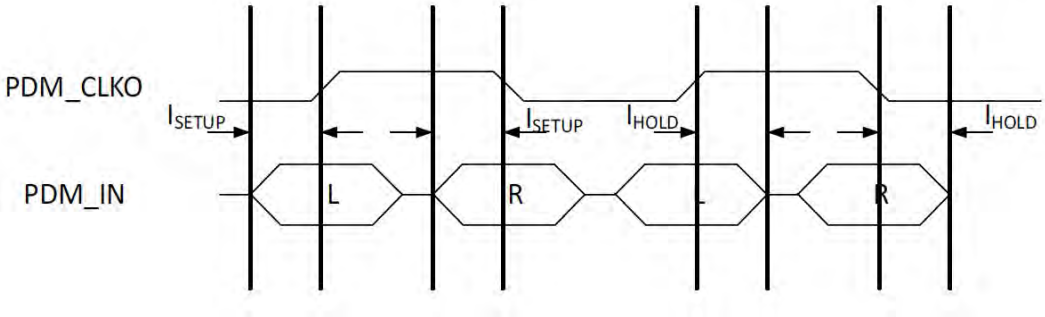
Figure 7-2: Stereo PDM to PCM Conversion Path



7.2.2 Clock Generation

The PDM module generates the clock which is supplied on the PDM_CLKO pin to the PDM source, and is shown in Figure 7-3.

Figure 7-3: PDM Clock Timing Diagram



There are two sources for this clock, which are selected by the VCFG_SELAP register bit. If SELAP is 0, this clock is an internally generated clock which is selected by the VCFG_PDMCLKSEL field and can range from 12 MHz to 187.5 KHz, and is enabled by setting the PCFG_PDMCLK bit. These clock selections are derived from the internal 48 MHz HFRC oscillator and therefore will have some frequency variation. If SELAP is 1, this clock is supplied externally on the I2S_BCLK pin. The input clock is used as the clock of the internal PDM logic, and therefore the lowest acceptable frequency should be selected to minimize power. The PDM logic includes separately clocked sections for each of the left and right channels.

The input clock is divided by 1, 2, 3 or 4 as selected by the PCFG_MCLKDIV field to produce the PDM_CLKO output.

NOTE: If achieving a nominal 50% duty cycle PDM output clock is important, then using a clock divider of divide-by-3 (MCKDIV3) for PCFG_MCLKDIV should be avoided as the resulting divided clock has a duty cycle of 67%, not the expected 50%. The other PCFG_MCLKDIV settings, MCKDIV1, MCKDIV2 and MCKDIV4 can be used to generate an output clock close to 50% duty cycle. See Table 7-1 for reference.

The following equations are for reference showing the relationship between SINC_RATE, MCLKDIV, sample rate and OSR.

$F_{PDM\_CLK} = F_S$ X 2 X SINC_RATE X MCLKDIV

$F_{PDM\_CLKO} = F_S$ X 2 X SINC_RATE

OSR = F_{PDM_CLKO} / F_S = 2 X SINC_RATE

NOTE: SINC_RATE must not be set higher than 64, even though the field is 7-bits wide.

The PDM module also requires a system clock to operate, which is enabled by the VCFG_IOCLKEN register bit. This bit should be kept at 0 whenever the PDM is not capturing input data to minimize power consumption.

The serial PDM input data is oversampled by a value specified in the PCFG_SINCRATE register field to produce the PCM data. The resulting PCM data rate is the PDM_CLKO frequency divided by the SINCRATE value and divided by 2. The table below shows some examples of frequency selection.

Table 7-1: PDM Clock Output Reference Table

F_S (kHz)	Duty Cycle (%)	F_{PDM_CLKO}(kHz)	OSR	MCLKDIV	SINC_RATE	Clock Source
7.8125	50	750	96	MCKDIV1	48	750kHz (MCU HFRC)
15.625	50	750	48	MCKDIV1	24	750kHz (MCU HFRC)
15.625	50	1500	96	MCKDIV1	48	1.5MHz (MCU HFRC)
7.8125	67	1000	128	MCKDIV3	64	3MHz (MCU HFRC)
8	50	768	96	MCKDIV1	48	768kHz (external I2S_BCLK)
16	50	768	48	MCKDIV1	24	768kHz (external I2S_BCLK)

The PDM controller also includes separate clock gates for left and right channel. This allows for lower power operation in mono microphone configuration.

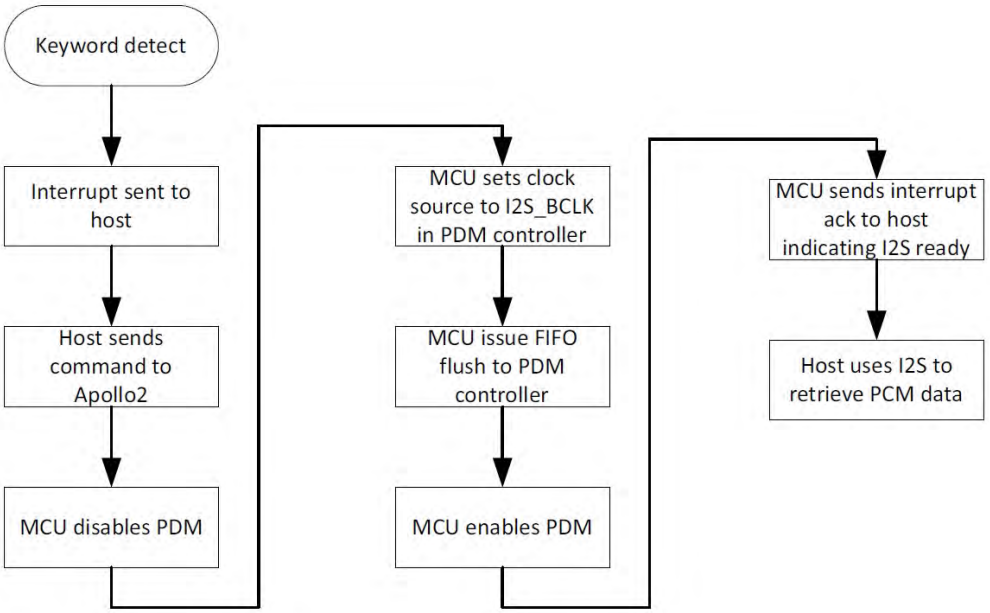
7.2.3 Clock Switching

The Apollo2 SoC supports dual-mode clock sourcing for PDM microphone operation. The first mode is clock sourcing from the SoC directly (via divided down HFRC reference). The second mode is clock sourcing from an external host via the I2S_BCLK.

The scenario for switching clock sources is if a higher accuracy clock is required based on the audio sampling requirements. The SoC clock source is based off of an RC oscillator which has intrinsic jitter that affects the quality of the resulting clock. For general voice command processing, the quality of the clock is sufficient. However, for voice recording/playback scenarios, this could manifest as pitch/noise problems. In a scenario where the SoC is used for voice/keyword detect, upon detection, the SoC can generate notification to the external host. The external host can then send a command to the SoC to switch clock source.

Clock switching requires careful orchestration since the PDM controller will continue to collect/process samples during this transition. The flow below is an example of how this transition can be handled.

Figure 7-4: PDM Clock Source Switching Flow



7.2.4 Operating Modes

The PDM module can operate in a variety of modes selected by the CHSET, PCMPACK and LRSWAP register fields, as shown in Table 7-2 on page 207. The FIFO Data Format column shows the PCM data that will be presented on each 32-bit read from the FIFO, in two 16-bit segments. “L0” indicates the first 16-bit sample from

the left channel, “L1” indicates the second left channel sample, “R0” indicates the first 16-bit sample from the right channel, etc.

Table 7-2: PDM Operating Modes and Data Formats

Mode	CHSET	PCMPACK	LRSWAP	31 - FIFO Data Format - 0		MCLKL	MCLKR
Mono Left Packed	01	1	N/A	L1	L0	En	Dis
				L3	L2		
Mono Right Packed	10	1	N/A	R1	R0	Dis	En
				R3	R2		
Stereo Packed	11	1	0	R0	L0	En	En
				R1	L1		
Stereo Packed Swapped	11	1	1	L0	R0	En	En
				L1	R1		
Mono Left Unpacked	01	0	N/A	0000	L0	En	Dis
				0000	L1		
Mono Right Unpacked	10	0	N/A	0000	R0	Dis	En
				0000	R1		
Stereo Unpacked	11	0	0	0000	L0	0000	R0
				En	En		
Stereo Unpacked Swapped	11	0	1	0000	R0	En	En
				0000	L0		
Disabled	00	N/A	N/A	0000	0000	Dis	Dis
				0000	0000		

The MCLKL and MCLKR columns indicate whether the left and right channel clocks are enabled or disabled.

7.2.5 FIFO Control and Interrupts

The PCM data is retrieved from the PDM module through a 256-word FIFO, read at the FRD Register. The number of words currently in the FIFO (0 to 256) is read in the FR_FIFOCNT field. If the FLUSH Register is written (with any value) FIFOCNT is set to 0 and any data in the FIFO is discarded. Each read from the FRD Register will decrement the FIFOCNT value, and FIFOCNT will be incremented each time new PCM data is written into the FIFO.

There are three interrupts which are generated based on the number of words in the FIFO. The UNDFL interrupt is generated if software reads from the FRD register when FIFOCNT is 0. The OVF interrupt is generated if PCM data is received when FIFOCNT is 256. The THR interrupt is set if PCM data is received and FIFOCNT is greater than or equal to the value in the FTHR_FIFOTHR Register field.

7.2.6 Digital Volume Gain

The PDM controller supports digital volume control with a range from -12dB to +10.5 dB in steps of 1.5 dB. It is programmed by register PGA_L and PGA_R for both left and right channels.

Table 7-3: Digital Volume Control

Port Name	Default	Description
PGA_L[3:0]	0000	Left Channel PGA Gain: +1.5dB/step 7→ +10.5dB; 6→ +9.0dB; 5→ +7.5dB; 4→ +6.0dB; 3→ +4.5dB; 2→ +3.0dB; 1→ +1.5dB; 0→ 0.0dB; 15→ -1.5dB; 14→-3.0dB; 13→ -4.5dB; 12→ -6.0dB; 11? -7.5dB; 10→ -9.0dB; 9→ -10.5dB; 8→ -12dB
PGA_R[3:0]	0000	Right Channel PGA Gain: +1.5dB/step 7→ +10.5dB; 6→ +9.0dB; 5→ +7.5dB; 4→ +6.0dB; 3→ +4.5dB; 2→ +3.0dB; 1→ +1.5dB; 0→ 0.0dB; 15→ -1.5dB; 14→ -3.0dB; 13→ -4.5dB; 12→-6.0dB; 11→ -7.5dB; 10→ -9.0dB; 9→ -10.5dB; 8→ -12dB

7.2.7 Low Pass Filter (LPF)

The controller's internal low pass filters attenuate the out-of-band noise at pre-defined bandwidth and corners.

Table 7-4: LPF Digital Filter Parameters

Parameter	Min	Typ	Max	Units
Pass band corner frequency		0.41		Fs
Pass band ripple	-1		1	dB
Stop band corner frequency	0.59			Fs
Stop band rejection		-60		dB

7.2.8 High Pass Filter

The discrete-time filter transfer function of the high pass filter as a function of the PDM_PCFG_HPCUTOFF field is:

$$H(Z) = (1 - Z^{-1}) / [1 - (1 - 2^{-(HPCUTOFF-8)}) Z^{-1}]$$

If HPCUTOFF = 1011, the high pass filter can be formulated by the polynomial:

$$H(Z) = (1 - Z^{-1}) / [1 - 0.875Z^{-1}]$$

At 16 KHz, the HPCUTOFF settings result in high pass corner frequencies as shown in Table 7-5.

Table 7-5: LPF High Pass Corner Frequency as a Function of HPCUTOFF

HPCUTOFF	High Pass Corner Frequency (Hz)
0x0 - 0x7	INVALID
0x8	1843
0x9	1049
0xA	572
0xB	301
0xC	155
0D	79
0E	40
0F	20

7.3 I²S Slave Interface

The PDM controller supports an optional I²S slave interface for PCM serial data output. This enables support for an external host controller to receive the serial output data from the converted PDM stream. In I²S slave mode, the MSB of I2S_DAT PCM data is available on the second rising edge of I2S_BCLK following an I2S_WCLK transition. The other bits up to the LSB are sent in order. The word length is 16 bits so there will be 16 bits of unused I2S_BCLK cycles between the LSB of one sample data and the MSB of the next one. The I2S_WCLK is always 32 clock cycles/phase.

Figure 7-5: I²S Interface Data Format Timing

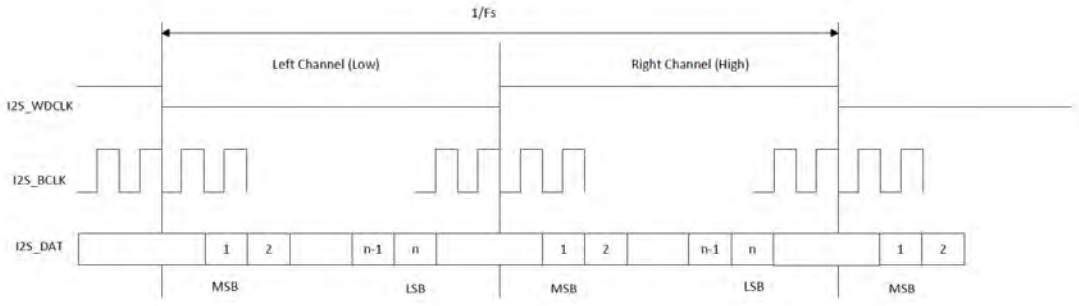
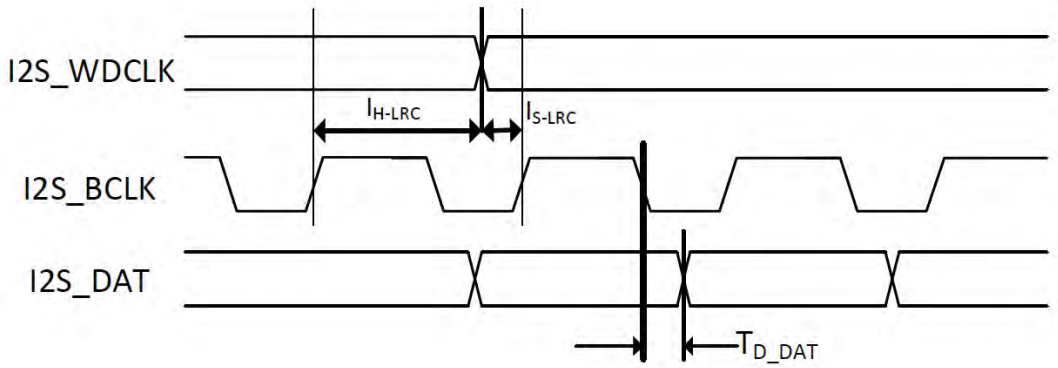


Figure 7-6: I2S Interface Setup and Hold Timing Diagram



7.4 PDM Registers

PDM Audio
INSTANCE 0 BASE ADDRESS:0x50011000

7.4.1 6.4.1 Register Memory Map

Table 7-6: PDM Register Map

Address(es)	Registered Name	Description
0x50011000	PCFG	PDM Configuration Register
0x50011004	VCFG	Voice Configuration Register
0x50011008	FR	Voice Status Register
0x5001100C	FRD	FIFO Read
0x50011010	FLUSH	FIFO Flush
0x50011014	FTHR	FIFO Threshold
0x50011200	INTEN	IO Master Interrupts: Enable
0x50011204	INTSTAT	IO Master Interrupts: Status
0x50011208	INTCLR	IO Master Interrupts: Clear
0x5001120C	INTSET	IO Master Interrupts: Set

7.4.2 PDM Registers

7.4.2.1 PCFG Register

PDM Configuration Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x50011000
PDM Configuration Register

Table 7-7: PCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LRSWAP	PGARIGHT				PGALEFT				RSVD				MCLKDIV		SINCRATE							ADCHPD	HPCUTOFF				CYCLES			SOFTMUTE	PDMCORE

Table 7-8: PCFG Register Bits

Bit	Name	Reset	RW	Description
31	LRSWAP	0x0	RW	Left/right channel swap. EN = 0x1 - Swap left and right channels (FIFO Read RIGHT_LEFT). NOSWAP = 0x0 - No channel swapping (IFO Read LEFT_RIGHT).
30:27	PGARIGHT	0x0	RW	Right channel PGA gain. M15DB = 0xF - -1.5 db gain. M300DB = 0xE - -3.0 db gain. M45DB = 0xD - -4.5 db gain. M60DB = 0xC - -6.0 db gain. M75DB = 0xB - -7.5 db gain. M90DB = 0xA - -9.0 db gain. M105DB = 0x9 - -10.5 db gain. M120DB = 0x8 - -12.0 db gain. P105DB = 0x7 - 10.5 db gain. P90DB = 0x6 - 9.0 db gain. P75DB = 0x5 - 7.5 db gain. P60DB = 0x4 - 6.0 db gain. P45DB = 0x3 - 4.5 db gain. P30DB = 0x2 - 3.0 db gain. P15DB = 0x1 - 1.5 db gain. 0DB = 0x0 - 0.0 db gain.

Table 7-8: PCFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
26:23	PGALEFT	0x0	RW	Left channel PGA gain. M15DB = 0xF - -1.5 db gain. M300DB = 0xE - -3.0 db gain. M45DB = 0xD - -4.5 db gain. M60DB = 0xC - -6.0 db gain. M75DB = 0xB - -7.5 db gain. M90DB = 0xA - -9.0 db gain. M105DB = 0x9 - -10.5 db gain. M120DB = 0x8 - -12.0 db gain. P105DB = 0x7 - 10.5 db gain. P90DB = 0x6 - 9.0 db gain. P75DB = 0x5 - 7.5 db gain. P60DB = 0x4 - 6.0 db gain. P45DB = 0x3 - 4.5 db gain. P30DB = 0x2 - 3.0 db gain. P15DB = 0x1 - 1.5 db gain. 0DB = 0x0 - 0.0 db gain.
22:19	RSVD	0x0	RO	This bit field is reserved for future use.
18:17	MCLKDIV	0x0	RW	PDM_CLK frequency divisor. MCKDIV4 = 0x3 - Divide input clock by 4 MCKDIV3 = 0x2 - Divide input clock by 3 MCKDIV2 = 0x1 - Divide input clock by 2 MCKDIV1 = 0x0 - Divide input clock by 1
16:10	SINCRATE	0x30	RW	SINC decimation rate.
9	ADCHPD	0x1	RW	High pass filter control. EN = 0x1 - Enable high pass filter. DIS = 0x0 - Disable high pass filter.
8:5	HPCUTOFF	0xb	RW	High pass filter coefficients.
4:2	CYCLES	0x1	RW	Number of clocks during gain-setting changes.
1	SOFTMUTE	0x0	RW	Soft mute control. EN = 0x1 - Enable Soft Mute. DIS = 0x0 - Disable Soft Mute.
0	PDMCORE	0x1	RW	Data Streaming Control. EN = 0x1 - Enable Data Streaming. DIS = 0x0 - Disable Data Streaming.

7.4.2.2 VCFG Register

Voice Configuration Register
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x50011004
Voice Configuration Register

Table 7-9: VCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IOCLKEN	RSTB	PDMCLKSEL			PDMCLK	RSVD					I2SMODE	BCLKINV	RSVD	DMICKDEL	SELAP	RSVD							PCMPACK	RSVD			CHSET	RSVD			

Table 7-10: VCFG Register Bits

Bit	Name	Reset	RW	Description
31	IOCLKEN	0x0	RW	Enable the IO clock. DIS = 0x0 - Disable FIFO read. EN = 0x1 - Enable FIFO read.
30	RSTB	0x0	RW	Reset the IP core. RESET = 0x0 - Reset the core. NORM = 0x1 - Enable the core.
29:27	PDMCLKSEL	0x0	RW	Select the PDM input clock. DISABLE = 0x0 - Static value. 12MHz = 0x1 - PDM clock is 12 MHz. 6MHz = 0x2 - PDM clock is 6 MHz. 3MHz = 0x3 - PDM clock is 3 MHz. 1_5MHz = 0x4 - PDM clock is 1.5 MHz. 750KHz = 0x5 - PDM clock is 750 KHz. 375KHz = 0x6 - PDM clock is 375 KHz. 187KHz = 0x7 - PDM clock is 187.5 KHz.
26	PDMCLK	0x0	RW	Enable the serial clock. DIS = 0x0 - Disable serial clock. EN = 0x1 - Enable serial clock.
25:21	RSVD	0x0	RO	This bit field is reserved for future use.
20	I2SMODE	0x0	RW	I^2S interface enable. DIS = 0x0 - Disable I^2S interface. EN = 0x1 - Enable I^2S interface.
19	BCLKINV	0x0	RW	I^2S BCLK input inversion. INV = 0x0 - BCLK inverted. NORM = 0x1 - BCLK not inverted.
18	RSVD	0x0	RO	This bit field is reserved for future use.
17	DMICKDEL	0x0	RW	PDM clock sampling delay. 0CYC = 0x0 - No delay. 1CYC = 0x1 - 1 cycle delay.
16	SELAP	0x0	RW	Select PDM input clock source. I^2S = 0x1 - Clock source from I^2S BCLK. INTERNAL = 0x0 - Clock source from internal clock generator.
15:9	RSVD	0x0	RO	This bit field is reserved for future use.
8	PCMPACK	0x0	RW	PCM data packing enable. DIS = 0x0 - Disable PCM packing. EN = 0x1 - Enable PCM packing.
7:5	RSVD	0x0	RO	This bit field is reserved for future use.

Table 7-10: VCFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
4:3	CHSET	0x1	RW	Set PCM channels. DIS = 0x0 - Channel disabled. LEFT = 0x1 - Mono left channel. RIGHT = 0x2 - Mono right channel. STEREO = 0x3 - Stereo channels.
2:0	RSVD	0x0	RO	This bitfield is reserved for future use.

7.4.2.3 FR Register

Voice Status Register
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x50011008
Voice Status Register

Table 7-11: FR Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	
RSVD	FIFOCNT

Table 7-12: FR Register Bits

Bit	Name	Reset	RW	Description
31:9	RSVD	0x0	RO	This bit field is reserved for future use.
8:0	FIFOCNT	0x0	RO	Valid 32-bit entries currently in the FIFO.

7.4.2.4 FRD Register

FIFO Read
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x5001100C
FIFO Read

Table 7-13: FRD Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
FIFOREAD

Table 7-14: FRD Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFOREAD	0x0	RO	FIFO read data.

7.4.2.5 FLUSH Register

FIFO Flush
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x50011010
FIFO Flush

Table 7-15: FLUSH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															FIFOFLUSH

Table 7-16: FLUSH Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bit field is reserved for future use.
0	FIFOFLUSH	0x0	WO	FIFO FLUSH.

7.4.2.6 FTHR Register

FIFO Threshold
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x50011014
FIFO Threshold

Table 7-17: FTHR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								FIFOTHR							

Table 7-18: FTHR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7:0	FIFOTHR	0xc0	RW	FIFO interrupt threshold.

7.4.2.7 INTEN Register

IO Master Interrupts: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x50011200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 7-19: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													UNDFL	OVF	THR

Table 7-20: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	UNDFL	0x0	RW	This is the FIFO underflow interrupt.
1	OVF	0x0	RW	This is the FIFO overflow interrupt.
0	THR	0x0	RW	This is the FIFO threshold interrupt.

7.4.2.8 INTSTAT Register

IO Master Interrupts: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x50011204
Read bits from this register to discover the cause of a recent interrupt.

Table 7-21: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													UNDFL	OVF	THR

Table 7-22: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	UNDFL	0x0	RW	This is the FIFO underflow interrupt.
1	OVF	0x0	RW	This is the FIFO overflow interrupt.
0	THR	0x0	RW	This is the FIFO threshold interrupt.

7.4.2.9 INTCLR Register

IO Master Interrupts: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x50011208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 7-23: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													UNDFL	OVF	THR

Table 7-24: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	UNDFL	0x0	RW	This is the FIFO underflow interrupt.
1	OVF	0x0	RW	This is the FIFO overflow interrupt.
0	THR	0x0	RW	This is the FIFO threshold interrupt.

7.4.2.10 INTSET Register

IO Master Interrupts: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x5001120C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 7-25: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																													UNDFL	OVF	THR

Table 7-26: INTSET Register Bits

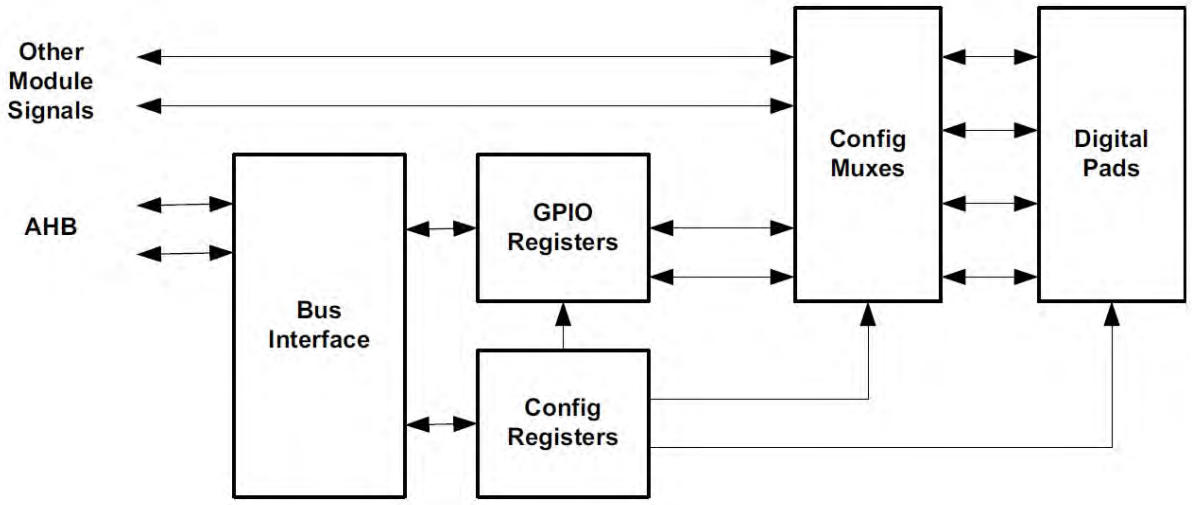
Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	UNDFL	0x0	RW	This is the FIFO underflow interrupt.
1	OVF	0x0	RW	This is the FIFO overflow interrupt.
0	THR	0x0	RW	This is the FIFO threshold interrupt.

SECTION

8

GPIO and Pad Configuration Module

Figure 8-1: Block Diagram for the General Purpose I/O (GPIO) Module



8.1 Functional Overview

The General Purpose I/O and Pad Configuration (GPIO) Module, shown in Figure 8-1, controls connections to up to 50 digital/analog pads. Each pad may be connected to a variety of module interface signals, with all pad input and output selection and control managed by the GPIO module. In addition, any pad may function as a general purpose input and/or output pad which may be configured for a variety of external functions. Each GPIO may be configured to generate an interrupt when a transition occurs on the input.

NOTE: Once the PADKEY is written, it should be explicitly cleared (with a non-key value) after GPIO configuration register updates are complete.

8.2 Pad Configuration Functions

The REG_GPIO_PADREGy (y = A to M) registers are used to control the function of each pad. Note that the REG_GPIO_PADKEY Register must be set to the value 0x73 in order to write the PADREGn registers. The REG_GPIO_PADREGy_PADnFNCSEL (n = 0 to 49) field selects one of up to eight signals to be used for each pad, as shown in Figure 8-2 on page 220. Functions are grouped by module, with the color coding shown in Figure 8-3 on page 221. This figure also defines the pad type for each configuration. The Special Pad Types are defined in Table 8-2 on page 221. Note that the CSP package only supports pads 0 through 23, 26, 28-29, 39-41, 44 and 47-49, which are indicated by an 'X' in the CSP PKG column of Figure 8-2 on page 220.

The REG_GPIO_PADREGy_PADnSTRNG and the REG_GPIO_ALTPADCFGy_PADn_DS1 bits control the drive strength of the pad. Nominal drive strengths of 2, 4, 8 or 12 mA can be selected with the setting of these two bits according to Table 8-1.

Table 8-1: Drive Strength Control Bits

ALTPADCFGy_PADn_DS1	PADREGy_PADnSTRNG	Nominal Drive Strength (mA)
0	0	2
0	1	4
1	0	8
1	1	12

IMPORTANT: Some of the pads on the Apollo2 SoC are limited to 4 mA maximum drive strength, and do not support the two higher drive strengths of 8 mA and 12 mA. The GPIO that do NOT support the higher drive strengths are: 3, 4, 6, 9, 11, 14-21, 24, 25, 30-37, 40, 41, 43, and 49. For these pads, setting the REG_GPIO_ALTPADCFGy_PADn_DS1 bit to 1 will put the pad in 4 mA drive mode regardless of the setting of the REG_GPIO_PADREGy_PADnSTRNG bit.

For all pads except for pad 20, REG_GPIO_PADREGy_PADnPULL bit enables a weak pull-up on the pad when set to one. For pad 20, the REG_GPIO_PADREGy_PAD20PULL bit enables a weak pull-down on the pad when set to one. The REG_GPIO_PADREGy_PADnINPEN bit must be set to enable the pad input, and should be left clear whenever the pad is not used in order to eliminate any leakage current in the pad. Pads 22 and 41 have selectable high side power switch transistors to provide ~1 Ω switches to VDDH.

Pad 4 has a selectable low side power switch transistor to provide a ~1 Ω switch to VSS. A high side power switch is enabled by setting the REG_GPIO_PADREGF_PAD22PWRUP or REG_GPIO_PADREGK_PAD41PWRUP bit, and a low side switch

is enabled by setting the REG_GPIO_PADREGB_PAD4PWRDN bit. Once enabled, the switches operate in parallel with the normal pad function.

Pads 0, 1, 5, 6, 8, 9, 25, 27, 39, 40, 42, 43, 48 and 49 include optional pull-up resistors for use in I^2C mode, to eliminate the need for external resistors. If the pull-up is enabled by the PADnPULL bit, the REG_GPIO_PADREGy_PADnRSEL field selects the size of the pull-up resistor as shown in Table 8-3 on page 222.

Figure 8-2: Apollo2 SoC Pad Function Mapping

	PADnFNCSEL								CSP PKG
Pad	0	1	2	3	4	5	6	7	
0	SLSCL	SLSCK	CLKOUT	GPIO00	M0SCKLB	M2SCK	M0SCLLB	M2SCL	X
1	SLSDA	SLMISO	UART0TX	GPIO01	M0MISOLB	M2MISO	M0SDALB	M2SDA	X
2	SLWIR3	SLMOSI	UART0RX	GPIO02	M0MOSILB	M2MOSI	M0WIR3LB	M2WIR3	X
3	UA0RTS	SLnCE	M1nCE4	GPIO03	M0nCE4LB	M2nCE0	TRIG1	I2S_WCLK	X
4	UA0CTS	SLINT	M0nCE5	GPIO04	SLINTGF	M2nCE5	CLKOUT	32KHz_XT	X
5	M0SCL	M0SCK	UA0RTS	GPIO05	M1SCKLB	-	M0SCLLB	M1nCE2	X
6	M0SDA	M0MISO	UA0CTS	GPIO06	SLMISOLB	M1nCE0	SLSDALB	I2S_DAT	X
7	M0WIR3	M0MOSI	CLKOUT	GPIO07	TRIG0	UART0TX	SLWIR3LB	M1nCE1	X
8	M1SCL	M1SCK	M0nCE4	GPIO08	M2nCE4	M1SCKLB	UART1TX	M1SCLLB	X
9	M1SDA	M1MISO	M0nCE5	GPIO09	M4nCE5	SLMISOLB	UART1RX	SLSDALB	X
10	M1WIR3	M1MOSI	M0nCE6	GPIO10	M2nCE6	UA1RTS	M4nCE4	SLWIR3LB	X
11	ADCSE2	M0nCE0	CLKOUT	GPIO11	M2nCE7	UA1CTS	UART0RX	PDM_DATA	X
12	ADCD0NSE9	M1nCE0	TCTA0	GPIO12	CLKOUT	PDM_CLK	UA0CTS	UART1TX	X
13	ADCD0PSE8	M1nCE1	TCTB0	GPIO13	M2nCE3	-	UA0RTS	UART1RX	X
14	ADCD1P	M1nCE2	UART1TX	GPIO14	M2nCE1	-	SWDCK	32KHz_XT	X
15	ADCD1N	M1nCE3	UART1RX	GPIO15	M2nCE2	-	SWDIO	SWO	X
16	ADCSE0	M0nCE4	TRIG0	GPIO16	M2nCE3	CMPIN0	UART0TX	UA1RTS	X
17	CMPRF1	M0nCE1	TRIG1	GPIO17	M4nCE3	-	UART0RX	UA1CTS	X
18	CMPIN1	M0nCE2	TCTA1	GPIO18	M4nCE1	-	UART1TX	32KHz_XT	X
19	CMPRF0	M0nCE3	TCTB1	GPIO19	TCTA1	-	UART1RX	I2S_BCLK	X
20	SWDCK	M1nCE5	TCTA2	GPIO20	UART0TX	UART1TX	-	-	X
21	SWDIO	M1nCE6	TCTB2	GPIO21	UART0RX	UART1RX	-	-	X
22	UART0TX	M1nCE7	TCTA3	GPIO22	PDM_CLK	-	TCTB1	SWO	X
23	UART0RX	M0nCE0	TCTB3	GPIO23	PDM_DATA	CMPOUT	TCTB1	-	X
24	M2nCE1	M0nCE1	CLKOUT	GPIO24	M5nCE0	TCTA1	I2S_BCLK	SWO	
25	-	M0nCE2	TCTA0	GPIO25	M2SDA	M2MISO	SLMISOLB	SLSDALB	
26	-	M0nCE3	TCTB0	GPIO26	M2nCE0	TCTA1	M5nCE1	M3nCE0	X
27	-	M1nCE4	TCTA1	GPIO27	M2SCL	M2SCK	M2SCKLB	M0SCLLB	
28	I2S_WCLK	M1nCE5	TCTB1	GPIO28	M2WIR3	M2MOSI	M5nCE3	SLWIR3LB	X
29	ADCSE1	M1nCE6	TCTA2	GPIO29	UA0CTS	UA1CTS	M4nCE0	PDM_DATA	X
30	-	M1nCE7	TCTB2	GPIO30	UART0TX	UA1RTS	-	I2S_DAT	
31	ADCSE3	M0nCE4	TCTA3	GPIO31	UART0RX	TCTB1	-	-	
32	ADCSE4	M0nCE5	TCTB3	GPIO32	-	TCTB1	-	-	
33	ADCSE5	M0nCE6	32KHz_XT	GPIO33	-	M3nCE7	TCTB1	SWO	
34	ADCSE6	M0nCE7	M2nCE3	GPIO34	CMPRF2	M3nCE1	M4nCE0	M5nCE2	
35	ADCSE7	M1nCE0	UART1TX	GPIO35	M4nCE6	TCTA1	UA0RTS	M3nCE2	
36	TRIG1	M1nCE1	UART1RX	GPIO36	32KHz_XT	M2nCE0	UA0CTS	M3nCE3	
37	TRIG2	M1nCE2	UA0RTS	GPIO37	M3nCE4	M4nCE1	PDM_CLK	TCTA1	
38	TRIG3	M1nCE3	UA0CTS	GPIO38	M3WIR3	M3MOSI	M4nCE7	SLWIR3LB	
39	UART0TX	UART1TX	CLKOUT	GPIO39	M4SCL	M4SCK	M4SCKLB	M4SCLLB	X
40	UART0RX	UART1RX	TRIG0	GPIO40	M4SDA	M4MISO	SLMISOLB	SLSDALB	X
41	M2nCE1	CLKOUT	SWO	GPIO41	M3nCE5	M5nCE7	M4nCE2	UA0RTS	X
42	M2nCE2	M0nCE0	TCTA0	GPIO42	M3SCL	M3SCK	M4SCKLB	M0SCLLB	
43	M2nCE4	M0nCE1	TCTB0	GPIO43	M3SDA	M3MISO	SLMISOLB	SLSDALB	
44	UA1RTS	M0nCE2	TCTA1	GPIO44	M4WIR3	M4MOSI	M5nCE6	SLWIR3LB	X
45	UA1CTS	M0nCE3	TCTB1	GPIO45	M4nCE3	M3nCE6	M5nCE5	SWO	
46	32KHz_XT	M0nCE4	TCTA2	GPIO46	TCTA1	M5nCE4	M4nCE4	SWO	
47	M2nCE5	M0nCE5	TCTB2	GPIO47	M5WIR3	M5MOSI	M4nCE5	SLWIR3LB	X
48	M2nCE6	M0nCE6	TCTA3	GPIO48	M5SCL	M5SCK	M5SCKLB	M5SCLLB	X
49	M2nCE7	M0nCE7	TCTB3	GPIO49	M5SDA	M5MISO	SLMISOLB	SLSDALB	X

Figure 8-3: Pad Function Color and Symbol Code

Color/ Symbol	Function	Pad Type
	ADC Signals	Analog or Input, as indicated by [A] or [I] respectively
	I^2C/SPI Slave Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively,
	I^2C/SPI Master 0 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	I^2C/SPI Master 1 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	I^2C/SPI Master 2 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	I^2C/SPI Master 3 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	I^2C/SPI Master 4 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	I^2C/SPI Master 5 Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	GPIO Signals	Controlled by GPIO Configuration
	Counter/Timer Signals	Controlled by CTIMER Configuration
	UART0 Signals	Input or Push-pull output, as indicated by [I] or [O] respectively
	UART1 Signals	Input or Push-pull output, as indicated by [I] or [O] respectively
	Audio Signals	Input or Push-pull output, as indicated by [I] or [O] respectively
	CLKOUT Signals	Push-pull Output
	Loopback Connections	Tri-state
	Miscellaneous Signals	Input Special or Push-pull output, as indicated by [I], [S] or [O] respectively
*	High-side power switch	Pads with a (*) (pads 22 and 41) have selectable high side power switch transistors to provide ~1 Ω switches to VDDH.
**	Low-side power switch	Pads with a (**) (pad 4) have selectable low side power switch transistors to provide ~1 Ω switches to VSS.

Table 8-2: Special Pad Types

Pad	PADnFNCSEL	Name	Pad Type[1]
0	7	M2SCL	Open Drain*
1	0	SLSDA	Bidirectional Open Drain
1	7	M2SDA	Bidirectional Open Drain*
2	0	SLWIR3	Bidirectional Tri-state*

Table 8-2: Special Pad Types *(Continued)*

Pad	PADnFNCSEL	Name	Pad Type[1]
2	7	M2WIR3	Bidirectional Tri-state*
5	0	M0SCL	Open Drain
6	0	M0SDA	Bidirectional Open Drain
7	0	M0WIR3	Bidirectional Tri-state
8	0	M1SCL	Open Drain
9	0	M1SDA	Bidirectional Open Drain
10	0	M1WIR3	Bidirectional Tri-state
15	6	SWDIO	Bidirectional Tri-state
21	0	SWDIO	Bidirectional Tri-state
25	4	M2SDA	Bidirectional Open Drain
27	4	M2SCL	Open Drain
28	4	M2WIR3	Bidirectional Tri-state
38	4	M3WIR3	Bidirectional Tri-state
39	4	M4SCL	Open Drain
40	4	M4SDA	Bidirectional Open Drain
42	4	M3SCL	Open Drain
43	4	M3SDA	Bidirectional Open Drain
44	4	M4WIR3	Bidirectional Tri-state
47	4	M5WIR3	Bidirectional Tri-state
48	4	M5SCL	Open Drain
49	4	M5SDA	Bidirectional Open Drain

[1] Pad types with * have option for I^2C pull-up resistor.

Table 8-3: I^2C Pull-up Resistor Selection

RSEL[1:0]	Pull-up Resistor
00	1.5 kΩ
01	6 kΩ
10	12 kΩ
11	24 kΩ

8.3 General Purpose I/O (GPIO) Functions

For each pad, if the PADnFNCSEL field is set to 0x3 the pad is connected to the corresponding GPIO signal. This section describes the configuration functions specific to GPIO pads.

8.3.1 Configuring the GPIO Functions

Each GPIO must be configured in the REG_GPIO_CFGy (y = A to G) Registers as an input and/or output before using. Note that the PADKEY Register must be set to the value 0x73 in order to write the REG_GPIO_CFGy Registers. Each output may be push-pull, open drain, disabled, or tri-stated as selected by the REG_GPIO_CFGy_GPIOnOUTCFG field. If the output is configured as push-pull, the pad will be driven with the corresponding bit in the REG_GPIO_WTy (y = A or B) Register. If the output is configured as open drain, the pad will be pulled low if the corresponding bit in the WTy Register is a 0, and will be floating if the corresponding bit in the WTy Register is a 1. If the output is configured as tri-state, the pad will be driven with the corresponding bit in the WTy Register if the corresponding bit in the REG_GPIO_ENy Register is a 1. If the bit in ENy is a 0, the output will be floating.

8.3.2 Reading from a GPIO Pad

All GPIO inputs are readable at all times, even if the pad is not configured as a GPIO. The current values of pads 0 to 31 are read in the REG_GPIO_RDA Register, and the current values of pads 32 to 49 are read in the REG_GPIO_RDB Register. If the REG_GPIO_CFGy_GPIOnINCFG bit is set for a GPIO, it will always read as zero.

8.3.3 Writing to a GPIO Pad

The GPIO pad outputs are controlled by the REG_GPIO_WTA/B Registers and the REG_GPIO_ENA/B Registers. Each of these registers may be directly written and read. Because each GPIO is often an independent function, the capability also exists to set or clear one or more bits without having to perform a read-modify-write operation. If the REG_GPIO_WTSA/B Register is written, the corresponding bit in WTA/B will be set if the write data is 1, otherwise the WTA/B bit will not be changed. If the REG_GPIO_WTCA/B Register is written, the corresponding bit in WTA/B will be cleared if the write data is 1, otherwise the WTA/B bit will not be changed.

If a GPIO pad is configured for tri-state output mode, the ENA/B Register controls the enabling of each bit. These registers may be directly written, and individual bits may be set or cleared by writing the ENSA/B or ENCA/B Registers with a 1 in the desired bit position.

8.3.4 GPIO Interrupts

Each GPIO pad can be configured to generate an interrupt on a high-to-low transition or a low-to-high transition, as selected by setting the REG_GPIO_CFGy_GPIOnINTD bit. This interrupt will be generated even if the pad is not configured as a GPIO in the Pad Configuration logic.

Each interrupt is enabled, disabled, cleared or set with a standard set of interrupt registers REG_GPIO_INT0EN, REG_GPIO_INT0STAT, REG_GPIO_INT0CLR and REG_GPIO_INT0SET for GPIO pads 0 to 31, and Registers REG_GPIO_INT1EN, REG_GPIO_INT1STAT, REG_GPIO_INT1CLR and REG_GPIO_INT1SET for GPIO pads 32 to 49.

Note that these interrupts get mapped to different IRQs and hence respective interrupts need to be enabled in NVIC and serviced accordingly.

8.4 Pad Connection Summary

Figure 8-4 on page 225 shows the detailed implementation of each pad. Each element will be described in detail.

8.4.1 Output Selection

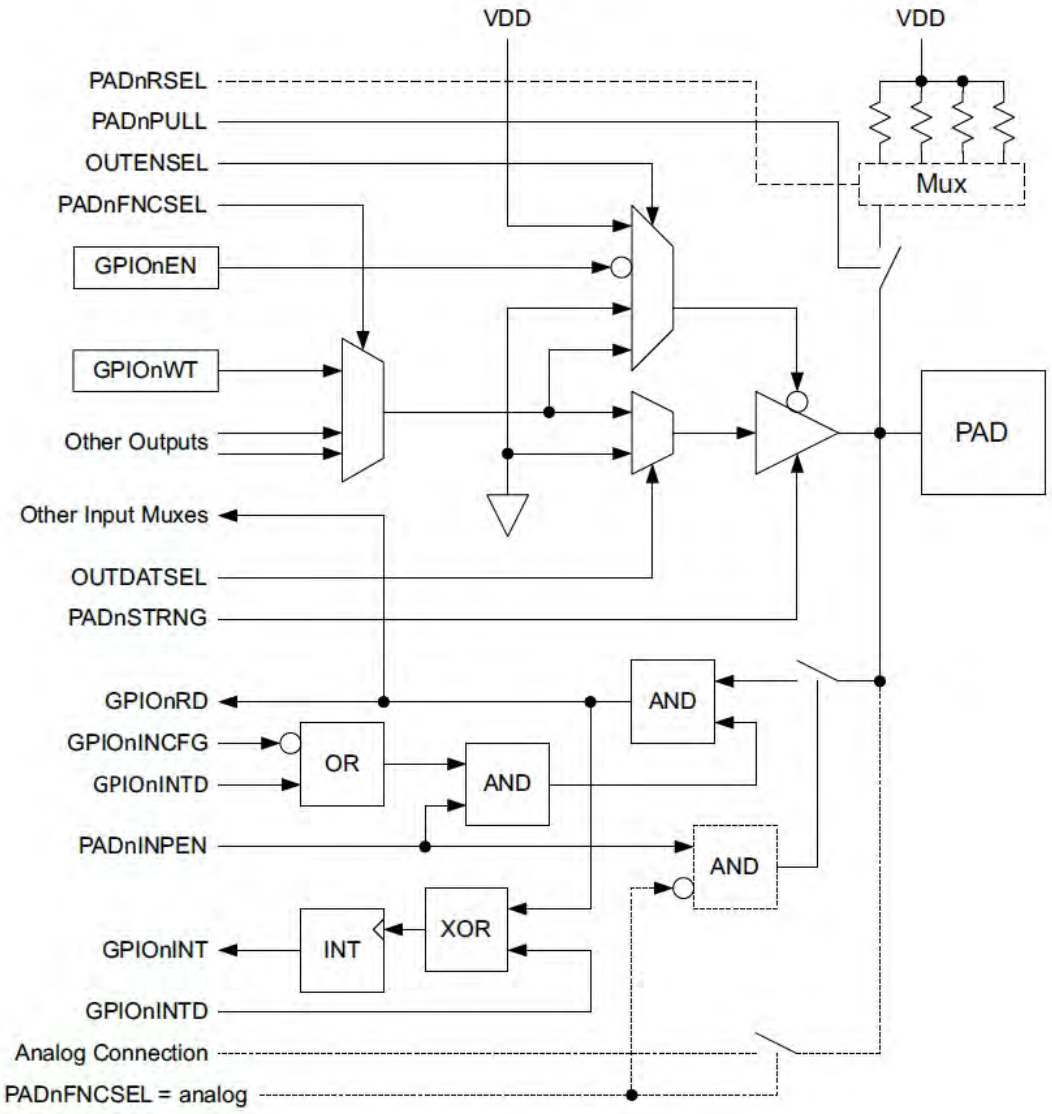
There is a multiplexer which selects the module signal to be driven to the output based on REG_GPIO_PADREGy_PADnFNCSEL (y = A to M (n = 0 to 49)) field. This implements the multiplexing shown in Figure 8-2 on page 220 for output pads. For all pads, a PADnFNCSEL value of 0x3 selects the value in the corresponding GPIO_WTy register bit.

8.4.2 Output Control

The pad driver for each pad has a data input and an output enable input. Each of these controls is selected from among several alternatives based on the OUTDATSEL and OUTENSEL signals which are controlled by the selection of the output type as shown in Table 8-2 on page 221 and Table 8-3 on page 222.

OUTDATSEL normally selects the data from the output multiplexer, but if the pad is configured as Open Drain the data input is selected to be low.

Figure 8-4: Pad Connection Details



OUTENSEL normally selects a ground signal to keep the pad driver enabled. If the pad is configured to be Open Drain, the pad enable is driven with the data from the output multiplexer. If the pad is configured as a GPIO (PADnFNCSEL = 0x3) and the GPIO drive type is tri-state (GPIOnOUTCFG = 0x3), the pad enable is driven with the inverse of the corresponding GPIO bit in the GPIO_ENx register. If the pad is not configured as an output, the pad enable is forced high to turn the driver off.

The drive strength of each pad driver is configured as described in *Section 8.2 Pad Configuration Functions on page 219*.

8.4.3 Input Control

The input circuitry of the pad may be disabled by clearing the PADnINPEN bit. This configuration should always be set if the pad input is not being used, as it prevents unnecessary current consumption if the pad voltage happens to float to a level between VDD and Ground. If PADnINPEN is 0, the pad will always read as a 0.

If PADnINPEN is set, the pad input then goes to two places. It is driven to the selected module signal as selected in Figure 8-2 on page 220. In addition, the pad input can always be read from the GPIO_RDx register unless the pad is configured as a GPIO (PADnFNCSEL = 0x3) and GPIOnINCFG is high, which will force the GPIO_RD input to be a zero. The ability to always read the pad value is very useful in some diagnostic cases.

The pad input is always sent to the GPIO interrupt logic, and a pad transition in the direction selected by GPIOnINTD will set the GPIOn_INT flip-flop. Note that this interrupt will be set even if the pad is not configured as a GPIO, which may be useful in detecting functions. As an example, this could be used to generate an interrupt when the I^2C/SPI Slave nCE signal is driven low by the Interface Host.

8.4.4 Pull-up Control

If PADnPULL is high, a pull-up resistor is connected between the pad and VDDH, except for pad 20, where PADnPULL connects the resistor to VSS rather than VDDH.

The fourteen pads which can be I^2C/SPI Master output drivers (0, 1, 5, 6, 8, 9, 25, 27, 39, 40, 42, 43, 48 and 49) contain the additional circuitry required for this functionality. In this case one of four different pull-up resistors are selected among options for the PADnRSEL field.

8.4.5 Analog Pad Configuration

Pads which may have analog connections (11-19, 29 and 31-35) include the circuitry shown with the dotted lines of Figure 8-4 on page 225. If the pad is configured in analog mode (reference the analog input function selections in Table 2-1 on page 43), the pad is connected directly to the particular analog module signal. In addition, OUTENSEL is forced high to disable the pad output, and the input of the pad is disabled independent of the value of PADnINPEN.

8.5 Module-specific Pad Configuration

The following sections describe in detail how to configure the pads for each module function.

8.5.1 Implementing IO Master Connections

The six IO Master modules must be correctly connected to the appropriate pads in order to operate.

8.5.1.1 High Speed IO Master 0 I²C Connection

I²C mode of IO Master 0 uses pad 5 as SCL and pad 6 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-4. The PAD5INPEN and PAD6INPEN bits must be set. If the internal I²C pull-up resistors are to be used, PAD5PULL and PAD6PULL should be set, and the PAD5RSEL and PAD6RSEL fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD5PULL and PAD6PULL should be cleared.

Table 8-4: IO Master 0 I²C Configuration

Field	Value
PAD5FNCSEL	0
PAD6FNCSEL	0

8.5.1.2 IO Master 1 I²C Connection

I²C mode of IO Master 1 uses pad 8 as SCL and pad 9 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-5. The PAD8INPEN and PAD9INPEN bits must be set. If the internal I²C pull-up resistors are to be used, PAD8PULL and PAD9PULL should be set, and the PAD8RSEL and PAD9RSEL fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD8PULL and PAD9PULL should be cleared.

Table 8-5: IO Master 1 I²C Configuration

Field	Value
PAD8FNCSEL	0
PAD9FNCSEL	0

8.5.1.3 IO Master 2 I²C Connection

I²C mode of IO Master 2 uses pad 0 (or pad 27) as SCL and pad 1 (or pad 25) as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-6 on page 228. The PAD0INPEN (or PAD27INPEN) and PAD1INPEN (or PAD25INPEN) bits must be set. If the internal I²C pull-up resistors are to be used, PAD0PULL (or PAD27PULL) and PAD1PULL (or PAD25PULL) should be set, and the PAD0RSEL (or

PAD27RSEL) and PAD1RSEL (or PAD25RSEL) fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD0PULL (or PAD27PULL) and PAD1PULL (or PAD25PULL) should be cleared.

Table 8-6: IO Master 2 I^{2}C Configuration

Field	Value
PAD0FNCSEL	7
PAD1FNCSEL	7
PAD27FNCSEL	4
PAD25FNCSEL	4

8.5.1.4 IO Master 3 I^{2}C Connection

I^{2}C mode of IO Master 3 uses pad 42 as SCL and pad 43 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-7. The PAD42INPEN and PAD43INPEN bits must be set. If the internal I^{2}C pull-up resistors are to be used, PAD42PULL and PAD43PULL should be set, and the PAD42RSEL and PAD43RSEL fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD42PULL and PAD43PULL should be cleared.

Table 8-7: IO Master 3 I^{2}C Configuration

Field	Value
PAD42FNCSEL	4
PAD43FNCSEL	4

8.5.1.5 High Speed IO Master 4 I^{2}C Connection

I^{2}C mode of IO Master 4 uses pad 39 as SCL and pad 40 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-8. The PAD39INPEN and PAD40INPEN bits must be set. If the internal I^{2}C pull-up resistors are to be used, PAD39PULL and PAD40PULL should be set, and the PAD39RSEL and PAD40RSEL fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD39PULL and PAD40PULL should be cleared.

Table 8-8: IO Master 4 I^{2}C Configuration

Field	Value
PAD39FNCSEL	4
PAD40FNCSEL	4

8.5.1.6 IO Master 5 I2C Connection

I2C mode of IO Master 5 uses pad 48 as SCL and pad 49 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-9. The PAD48INPEN and PAD49INPEN bits must be set. If the internal I2C pull-up resistors are to be used, PAD48PULL and PAD49PULL should be set, and the PAD48RSEL and PAD49RSEL fields should be set to select the desired pull-up resistor size as shown in Table 8-3 on page 222. If external pull-up resistors are used, PAD48PULL and PAD49PULL should be cleared.

Table 8-9: IO Master 5 I2C Configuration

Field	Value
PAD48FNCSEL	4
PAD49FNCSEL	4

8.5.1.7 High Speed IO Master 0 4-wire SPI Connection

Four-wire SPI mode of IO Master 0 uses pad 5 as SCK, pad 6 as MISO and pad 7 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-10. The PAD5INPEN and PAD6INPEN bits must be set.PAD5PULL, PAD6PULL and PAD7PULL should be cleared. A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-11. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-10: IO Master 0 4-wire SPI Configuration

Field	Value
PAD5FNCSEL	1
PAD6FNCSEL	1
PAD7FNCSEL	1

Table 8-11: IO Master 0 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD11FNCSEL	1	0	11
PAD23FNCSEL	1	0	23
PAD42FNCSEL	1	0	42
PAD17FNCSEL	1	1	17
PAD24FNCSEL	1	1	24
PAD43FNCSEL	1	1	43
PAD18FNCSEL	1	2	18
PAD25FNCSEL	1	2	25
PAD44FNCSEL	1	2	44

Table 8-11: IO Master 0 4-wire SPI nCE Configuration *(Continued)*

Field	Value	nCE Selection	Pad Used
PAD19FNCSEL	1	3	19
PAD26FNCSEL	1	3	26
PAD45FNCSEL	1	3	45
PAD8FNCSEL	2	4	8
PAD16FNCSEL	1	4	16
PAD31FNCSEL	1	4	31
PAD46FNCSEL	1	4	46
PAD4FNCSEL	2	5	4
PAD9FNCSEL	2	5	9
PAD32FNCSEL	1	5	32
PAD47FNCSEL	1	5	47
PAD10FNCSEL	2	6	10
PAD33FNCSEL	1	6	33
PAD48FNCSEL	1	6	48
PAD34FNCSEL	1	7	34
PAD49FNCSEL	1	7	49

8.5.1.8 IO Master 1 4-wire SPI Connection

Four-wire SPI mode of IO Master 1 uses pad 8 as SCK, pad 9 as MISO and pad 10 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-12. The PAD8INPEN and PAD9INPEN bits must be set. PAD8PULL, PAD9PULL and PAD10PULL should be cleared.

Table 8-12: IO Master 1 4-wire SPI Configuration

Field	Value
PAD8FNCSEL	1
PAD9FNCSEL	1
PAD10FNCSEL	1

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-13. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-13: IO Master 1 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD6FNCSEL	5	0	6
PAD12FNCSEL	1	0	12

Table 8-13: IO Master 1 4-wire SPI nCE Configuration *(Continued)*

Field	Value	nCE Selection	Pad Used
PAD35FNCSEL	1	0	35
PAD7FNCSEL	7	1	7
PAD13FNCSEL	1	1	13
PAD36FNCSEL	1	1	36
PAD5FNCSEL	7	2	5
PAD14FNCSEL	1	2	14
PAD37FNCSEL	1	2	37
PAD15FNCSEL	1	3	15
PAD38FNCSEL	1	3	38
PAD3FNCSEL	2	4	3
PAD27FNCSEL	1	4	27
PAD20FNCSEL	1	5	20
PAD28FNCSEL	1	5	28
PAD21FNCSEL	1	6	21
PAD29FNCSEL	1	6	29
PAD22FNCSEL	1	7	22
PAD30FNCSEL	1	7	30

8.5.1.9 IO Master 2 4-wire SPI Connection

Four-wire SPI mode of IO Master 2 uses pad 0 (or pad 27) as SCK, pad 2 (or pad 28) as MOSI and pad 1 (or pad 25) as MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-14. The PAD0INPEN and PAD2INPEN bits must be set. PAD0PULL, PAD2PULL and PAD1PULL should be cleared.

Table 8-14: IO Master 2 4-wire SPI Configuration

Field	Value
PAD0FNCSEL	5
PAD27FNCSEL	5
PAD2FNCSEL	5
PAD28FNCSEL	5
PAD1FNCSEL	5
PAD25FNCSEL	5

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-15 on page 232. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-15: IO Master 2 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD3FNCSEL	5	0	3
PAD26FNCSEL	4	0	26
PAD36FNCSEL	5	0	36
PAD14FNCSEL	4	1	14
PAD24FNCSEL	0	1	24
PAD41FNCSEL	0	1	41
PAD15FNCSEL	4	2	15
PAD42FNCSEL	0	2	42
PAD13FNCSEL	4	3	13
PAD16FNCSEL	4	3	16
PAD34FNCSEL	2	3	34
PAD8FNCSEL	4	4	8
PAD43FNCSEL	0	4	43
PAD4FNCSEL	5	5	4
PAD47FNCSEL	0	5	47
PAD10FNCSEL	4	6	10
PAD48FNCSEL	0	6	48
PAD11FNCSEL	4	7	11
PAD49FNCSEL	0	7	49

8.5.1.10 IO Master 3 4-wire SPI Connection

Four-wire SPI mode of IO Master 3 uses pad 42 as SCK, pad 43 as MISO and pad 38 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-16. The PAD42INPEN and PAD38INPEN bits must be set. PAD38PULL, PAD42PULL and PAD43PULL should be cleared.

Table 8-16: IO Master 3 4-wire SPI Configuration

Field	Value
PAD38FNCSEL	5
PAD42FNCSEL	5
PAD43FNCSEL	5

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-17 on page 233. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-17: IO Master 3 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD26FNCSEL	7	0	26
PAD34FNCSEL	5	1	34
PAD35FNCSEL	7	2	35
PAD36FNCSEL	7	3	36
PAD37FNCSEL	4	4	37
PAD41FNCSEL	4	5	41
PAD45FNCSEL	5	6	45
PAD33FNCSEL	5	7	33

8.5.1.11 High Speed IO Master 4 4-wire SPI Connection

Four-wire SPI mode of IO Master 4 uses pad 39 as SCK, pad 40 as MISO and pad 44 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-18. The PAD39INPEN and PAD44INPEN bits must be set. PAD39PULL, PAD40PULL and PAD44PULL should be cleared.

Table 8-18: IO Master 4 4-wire SPI Configuration

Field	Value
PAD39FNCSEL	5
PAD40FNCSEL	5
PAD44FNCSEL	5

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-19. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-19: IO Master 4 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD29FNCSEL	6	0	29
PAD34FNCSEL	6	0	34
PAD18FNCSEL	4	1	18
PAD37FNCSEL	5	1	37
PAD41FNCSEL	6	2	41
PAD17FNCSEL	4	3	17
PAD45FNCSEL	4	3	45
PAD10FNCSEL	6	4	10
PAD46FNCSEL	6	4	46

Table 8-19: IO Master 4 4-wire SPI nCE Configuration *(Continued)*

Field	Value	nCE Selection	Pad Used
PAD9FNCSEL	4	5	9
PAD47FNCSEL	6	5	47
PAD35FNCSEL	4	6	35
PAD38FNCSEL	6	7	38

8.5.1.12 IO Master 5 4-wire SPI Connection

Four-wire SPI mode of IO Master 5 uses pad 48 as SCK, pad 49 as MISO and pad 47 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-20. The PAD48INPEN and PAD47INPEN bits must be set. PAD48PULL, PAD49PULL and PAD47PULL should be cleared.

Table 8-20: IO Master 5 4-wire SPI Configuration

Field	Value
PAD48FNCSEL	5
PAD49FNCSEL	5
PAD47FNCSEL	5

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-21. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 8-21: IO Master 5 4-wire SPI nCE Configuration

Field	Value	nCE Selection	Pad Used
PAD24FNCSEL	4	0	24
PAD26FNCSEL	6	1	26
PAD34FNCSEL	7	2	34
PAD28FNCSEL	6	3	28
PAD46FNCSEL	5	4	46
PAD45FNCSEL	6	5	45
PAD44FNCSEL	6	6	44
PAD41FNCSEL	5	7	41

8.5.1.13 High Speed IO Master 0 3-wire SPI Connection

Three-wire SPI mode of IO Master 0 uses pad 5 as SCK and pad 7 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-22 on page 235. The PAD5INPEN and PAD7INPEN bits must be set. PAD5PULL and PAD7PULL should be cleared. Pad 6 may be used for other functions.

Table 8-22: IO Master 0 3-wire SPI Configuration

Field	Value
PAD5FNCSEL	1
PAD7FNCSEL	0

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-11 on page 229. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.14 IO Master 1 3-wire SPI Connection

Three-wire SPI mode of IO Master 1 uses pad 8 as SCK and pad 10 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-23. The PAD8INPEN and PAD10INPEN bits must be set. PAD8PULL and PAD10PULL should be cleared. Pad 9 may be used for other functions.

Table 8-23: IO Master 1 3-wire SPI Configuration

Field	Value
PAD8FNCSEL	1
PAD10FNCSEL	0

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-13 on page 230. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.15 IO Master 2 3-wire SPI Connection

Three-wire SPI mode of IO Master 2 uses pad 0 (or pad 27) as SCK and pad 2 (or pad 28) as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-24. The PAD0INPEN (or PAD27INPEN) and PAD2INPEN (or PAD28INPEN) bits must be set. PAD0PULL (or PAD27PULL) and PAD2PULL (or PAD28PULL) should be cleared. Pad 1 or Pad 25 may be used for other functions.

Table 8-24: IO Master 2 3-wire SPI Configuration

Field	Value
PAD0FNCSEL	5
PAD27FNCSEL	5
PAD2FNCSEL	7
PAD28FNCSEL	4

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-15 on page 232. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.16 IO Master 3 3-wire SPI Connection

Three-wire SPI mode of IO Master 3 uses pad 42 as SCK and pad 38 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-25. The PAD42INPEN and PAD38INPEN bits must be set. PAD42PULL and PAD38PULL should be cleared. Pad 43 may be used for other functions.

Table 8-25: IO Master 3 3-wire SPI Configuration

Field	Value
PAD42FNCSEL	5
PAD38FNCSEL	4

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-17 on page 233. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.17 High Speed IO Master 4 3-wire SPI Connection

Three-wire SPI mode of IO Master 4 uses pad 39 as SCK and pad 44 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-26. The PAD39INPEN and PAD44INPEN bits must be set. PAD39PULL and PAD44PULL should be cleared. Pad 40 may be used for other functions.

Table 8-26: IO Master 0 3-wire SPI Configuration

Field	Value
PAD39FNCSEL	5
PAD44FNCSEL	4

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-19 on page 233. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.18 IO Master 5 3-wire SPI Connection

Three-wire SPI mode of IO Master 5 uses pad 48 as SCK and pad 47 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-27 on page 237. The PAD48INPEN and PAD47INPEN bits must be set. PAD48PULL and PAD47PULL should be cleared. Pad 49 may be used for other functions.

Table 8-27: IO Master 0 3-wire SPI Configuration

Field	Value
PAD48FNCSEL	5
PAD47FNCSEL	4

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 8-21 on page 234. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

8.5.1.19 SPI Flow Control Connections

SPI Flow Control in interrupt mode requires an external pin to be specified as the interrupt pin. This is accomplished by configuring the desired pin in the IOMxIRQ register (x = 0 to 5).

8.5.2 Implementing IO Slave Connections

The IO Master module must be correctly connected to the appropriate pads in order to operate.

8.5.2.1 IO Slave I^2C Connection

I^2C mode of the IO Slave uses pad 0 as SCL and pad 1 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-28. The PAD0INPEN and PAD1INPEN bits must be set. PAD0PULL and PAD1PULL should be cleared.

Table 8-28: IO Slave I^2C Configuration

Field	Value
PAD0FNCSEL	0
PAD1FNCSEL	0

8.5.2.2 IO Slave 4-wire SPI Connection

Four-wire SPI mode of the IO Slave uses pad 0 as SCK, pad 1 as MISO, pad 2 as MOSI and pad 3 as nCE. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-29 on page 238. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD1PULL, PAD2PULL and PAD3PULL should be cleared.

Table 8-29: IO Slave 4-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD1FNCSEL	1
PAD2FNCSEL	1
PAD3FNCSEL	1

8.5.2.3 IO Slave 3-wire SPI Connection

Three-wire SPI mode of the IO Slave uses pad 0 as SCK, pad 2 as MISO/MOSI and pad 3 as nCE. This mode is configured by setting the PADnFNCSEL fields as shown in Table 8-30. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD2PULL and PAD3PULL should be cleared. Pad 1 may be used for other functions.

Table 8-30: IO Slave 3-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD2FNCSEL	0
PAD3FNCSEL	1

8.5.2.4 IO Slave Interrupt Connection

The IO Slave can be configured to generate an interrupt output under a variety of internal conditions. If this function is used, the interrupt will be generated on pad 4. PAD4FNCSEL must be set to 1, and PAD4INPEN and PAD4PULL should be cleared.

8.5.2.5 Implementing Internal I^2C and SPI Loopback

Loopback involves connecting one of the 6 I^2C/SPI Masters to the I^2C/SPI Slave. In order to make this connection, two configurations are required, some of which are specific to the functional mode selected (I^2C, 3-wire SPI or 4-wire SPI). In all cases, the LOOPBACK register must be loaded with the value to select the desired I^2C/SPI Master (0 through 5).

8.5.2.5.1 I^2C Loopback

The Slave is configured in I^2C loopback mode by setting PAD0FNCSEL to 6 (MxSCLLB) and by setting PAD1FNCSEL to 6 (MxSDALB). The selected Master is configured as shown in Table 8-31 on page 239 below.

Table 8-31: I^2C Loopback

Master	PADxFNCSEL	Value	PADxFNCSEL	Value
0	PAD5FNCSEL	6	PAD6FNCSEL	6
1	PAD8FNCSEL	7	PAD9FNCSEL	7
2	PAD27FNCSEL	7	PAD25FNCSEL	7
3	PAD42FNCSEL	7	PAD43FNCSEL	7
4	PAD39FNCSEL	7	PAD40FNCSEL	7
5	PAD48FNCSEL	7	PAD49FNCSEL	7

8.5.2.5.2 3-wire SPI Loopback

The Slave is configured in 3-wire SPI loopback mode by setting PAD0FNCSEL to 4 (MxSCKLB), setting PAD2FNCSEL to 6 (MxWIR3LB) and setting PAD3FNCSEL to 4 (MxnCELB). The selected Master is configured as shown in the table below. Note that the SPI Channel selection is not relevant, as any of the eight Channels will loop nCE back to the Slave.

Table 8-32: 3-wire SPI Loopback

Master	PADxFNCSEL	Value	PADxFNCSEL	Value
0	PAD5FNCSEL	4	PAD7FNCSEL	6
1	PAD8FNCSEL	5	PAD10FNCSEL	7
2	PAD27FNCSEL	6	PAD28FNCSEL	7
3	PAD42FNCSEL	6	PAD38FNCSEL	7
4	PAD39FNCSEL	6	PAD44FNCSEL	7
5	PAD48FNCSEL	6	PAD47FNCSEL	7

8.5.2.5.3 4-wire SPI Loopback

The Slave is configured in 4-wire SPI loopback mode by setting PAD0FNCSEL to 4 (MxSCKLB), setting PAD1FNCSEL to 4 (MxMISOLB)), setting PAD2FNCSEL to 4 (MxMOSILB) and setting PAD3FNCSEL to 4 (MxnCELB). The selected Master is configured as shown in the table below. Note that the SPI Channel selection is not relevant, as any of the eight Channels will loop nCE back to the Slave.

Table 8-33: 4-wire SPI Loopback

Master	PADxFNCSEL	Value	PADxFNCSEL	Value
0	PAD5FNCSEL	4	PAD6FNCSEL	4
1	PAD8FNCSEL	5	PAD9FNCSEL	5
2	PAD27FNCSEL	6	PAD25FNCSEL	6
3	PAD42FNCSEL	6	PAD43FNCSEL	6

Table 8-33: 4-wire SPI Loopback *(Continued)*

Master	PADxFNCSEL	Value	PADxFNCSEL	Value
4	PAD39FNCSEL	6	PAD40FNCSEL	6
5	PAD48FNCSEL	6	PAD49FNCSEL	6

8.5.2.6 Loopback of the I²C/SPI Slave Interrupt

The I²C/SPI Slave interrupt can be exercised in loopback mode by setting PAD4FNCSEL to 4. This will cause the slave interrupt to be driven into the GPIO4 pin where it can be checked by software.

8.5.3 Implementing Counter/Timer Connections

Each Counter/Timer can optionally count pulses from an input pad, or generate pulses on an output pad. Table 8-34 shows the PADnFNCSEL settings to connect each Counter/Timer to the appropriate pad. If the pad is used as an input, the PADnINPEN bit should be set, otherwise it should be cleared. The PADnPULL bit may be set if the input signal is open drain.

Table 8-34: Counter/Timer Pad Configuration

Field	Value	Ctr/Timer	Pad
PAD12FNCSEL	2	A0	12
PAD25FNCSEL	2	A0	25
PAD42FNCSEL	2	A0	42
PAD13FNCSEL	2	B0	13
PAD26FNCSEL	2	B0	26
PAD43FNCSEL	2	B0	43
PAD18FNCSEL	2	A1	18
PAD19FNCSEL	4	A1	19
PAD24FNCSEL	5	A1	24
PAD26FNCSEL	5	A1	26
PAD27FNCSEL	2	A1	27
PAD35FNCSEL	5	A1	35
PAD37FNCSEL	7	A1	37
PAD44FNCSEL	2	A1	44
PAD45FNCSEL	7	A1	45
PAD46FNCSEL	4	A1	46
PAD19FNCSEL	2	B1	19
PAD22FNCSEL	6	B1	22
PAD23FNCSEL	6	B1	22

Table 8-34: Counter/Timer Pad Configuration *(Continued)*

Field	Value	Ctr/Timer	Pad
PAD28FNCSEL	2	B1	28
PAD31FNCSEL	5	B1	31
PAD32FNCSEL	5	B1	32
PAD33FNCSEL	6	B1	33
PAD45FNCSEL	2	B1	45
PAD20FNCSEL	2	A2	20
PAD29FNCSEL	2	A2	29
PAD46FNCSEL	2	A2	46
PAD21FNCSEL	2	B2	21
PAD30FNCSEL	2	B2	30
PAD47FNCSEL	2	B2	47
PAD22FNCSEL	2	A3	22
PAD31FNCSEL	2	A3	31
PAD48FNCSEL	2	A3	48
PAD23FNCSEL	2	B3	23
PAD32FNCSEL	2	B3	32
PAD49FNCSEL	2	B3	49

8.5.4 Implementing UART Connections

The UART signals can be connected to a variety of pads.

8.5.4.1 UART0 TX/RX Connections

The UART0 data signals TX and RX may each be connected to several pads. Note that TX and RX are selected independently. Table 8-35 shows the connections for TX, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 8-36 on page 242 shows the connections for RX, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

Table 8-35: UART0 TX Configuration

Field	Value	Pad
PAD1FNCSEL	2	1
PAD7FNCSEL	5	7
PAD16FNCSEL	6	16
PAD20FNCSEL	4	20
PAD22FNCSEL	0	22

Table 8-35: UART0 TX Configuration *(Continued)*

Field	Value	Pad
PAD30FNCSEL	4	30
PAD39FNCSEL	0	39

Table 8-36: UART0 RX Configuration

Field	Value	Pad
PAD2FNCSEL	2	2
PAD11FNCSEL	6	11
PAD17FNCSEL	6	17
PAD21FNCSEL	4	21
PAD23FNCSEL	0	23
PAD31FNCSEL	4	31
PAD40FNCSEL	0	40

8.5.4.2 UART0 RTS/CTS Connections

The UART modem control signals RTS and CTS may each be connected to one of two pads. Note that RTS and CTS are selected independently. Table 8-37 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 8-38 shows the connections for CTS, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

Table 8-37: UART0 RTS Configuration

Field	Value	Pad
PAD3FNCSEL	0	3
PAD5FNCSEL	2	5
PAD13FNCSEL	6	13
PAD35FNCSEL	6	35
PAD37FNCSEL	2	37
PAD41FNCSEL	7	41

Table 8-38: UART0 CTS Configuration

Field	Value	Pad
PAD4FNCSEL	0	4
PAD6FNCSEL	2	6
PAD12FNCSEL	6	12
PAD29FNCSEL	4	29

Table 8-38: UART0 CTS Configuration *(Continued)*

Field	Value	Pad
PAD36FNCSEL	6	36
PAD38FNCSEL	2	38

8.5.4.3 UART1 TX/RX Connections

The UART data signals TX and RX may each be connected to several pads. Note that TX and RX are selected independently. Table 8-39 shows the connections for RX, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

Table 8-39: UART1 RX Configuration

Field	Value	Pad
PAD9FNCSEL	6	9
PAD13FNCSEL	7	13
PAD15FNCSEL	2	15
PAD19FNCSEL	6	19
PAD21FNCSEL	5	21
PAD36FNCSEL	2	36
PAD40FNCSEL	1	40

8.5.4.4 UART1 RTS/CTS Connections

The UART modem control signals RTS and CTS may each be connected to one of two pads. Note that RTS and CTS are selected independently. Table 8-40 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 8-41 on page 244 shows the connections for CTS, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

Table 8-40: UART1 RTS Configuration

Field	Value	Pad
PAD10FNCSEL	5	10
PAD16FNCSEL	7	16
PAD30FNCSEL	5	30
PAD44FNCSEL	0	44

Table 8-41: UART1 CTS Configuration

Field	Value	Pad
PAD11FNCSEL	5	11
PAD17FNCSEL	7	17
PAD29FNCSEL	5	29
PAD45FNCSEL	0	45

8.5.5 Implementing Audio Connections

The Audio signals can be connected to a variety of pads.

8.5.5.1 PDM Connections

The PDM CLK and DATA signals may each be connected to several pads. Note that CLK and DATA are selected independently. Table 8-42 shows the connections for PDM CLK, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 8-43 shows the connections for PDM DATA, which must have the corresponding PADnINPEN field set and the corresponding PADnPULL field clear.

Table 8-42: PDM CLK Configuration

Field	Value	Pad
PAD12FNCSEL	5	12
PAD22FNCSEL	4	22
PAD37FNCSEL	6	37

Table 8-43: PDM DATA Configuration

Field	Value	Pad
PAD11FNCSEL	7	11
PAD23FNCSEL	4	23
PAD29FNCSEL	7	31

8.5.5.2 I^2S Connections

The I^2S BCLK, WCLK and DAT signals may each be connected to one of several pads. Note that BCLK, WCLK and DAT are selected independently. Table 8-44 on page 245 shows the connections for I^2S BCLK, which should have the corresponding PADnINPEN set and the corresponding PADnPULL field clear. Table 8-45 on page 245 shows the connections for I^2S WCLK, which should have the corresponding PADnINPEN set and the PADnPULL field clear. Table 8-46 on page 245 shows the con-

nections for I^{2}S DAT, which must have the corresponding PADnINPEN and PADnPULL fields clear.

Table 8-44: I^{2}S BCLK Configuration

Field	Value	Pad
PAD19FNCSEL	7	19
PAD24FNCSEL	6	24

Table 8-45: I^{2}S WCLK Configuration

Field	Value	Pad
PAD3FNCSEL	7	3
PAD28FNCSEL	0	28

Table 8-46: I^{2}S DAT Configuration

Field	Value	Pad
PAD6FNCSEL	7	6
PAD30FNCSEL	7	30

8.5.6 Implementing GPIO Connections

Each pad of the Apollo2 SoC can be configured as a GPIO port by setting PADnFNCSEL to 3. PADnINPEN and PADnPULL must be set appropriately depending on the specific GPIO function.

8.5.7 Implementing CLKOUT Connections

The flexible clock output of the Clock Generator module, CLKOUT, may be configured on several pads as shown in Table 8-47. PADnINPEN and PADnPULL should be cleared in each case.

Table 8-47: CLKOUT Configuration

Field	Value	Pad
PAD4FNCSEL	6	4
PAD11FNCSEL	2	11
PAD12FNCSEL	4	12
PAD24FNCSEL	2	24
PAD39FNCSEL	2	39
PAD41FNCSEL	1	41

8.5.8 Implementing 32kHz CLKOUT Connections

In addition to the CLKOUT mux output, there is also a dedicated 32 kHz clock output. This clock is primarily for leveraging the 32 kHz oscillator clock from Apollo2 MCU. This clock output may be configured on several pads as shown in Table 8-48. PADnINPEN and PADnPULL should be cleared in each case.

Table 8-48: 32 kHz CLKOUT Configuration

Field	Value	Pad
PAD4FNCSEL	7	4
PAD14FNCSEL	7	14
PAD18FNCSEL	7	18
PAD33FNCSEL	2	33
PAD36FNCSEL	4	36
PAD46FNCSEL	0	46

8.5.9 Implementing ADC Connections

Three types of pad connections may be made for the ADC module. Up to twelve pads may be selected from and configured as the analog inputs, as shown in Table 8-49. The ADCREF reference voltage input is supplied on a dedicated input pin.

If an external digital trigger is desired, up to eight selectable pad choices may be selected from and configured, as shown in Table 8-50 on page 247. For the trigger inputs, PADnINPEN must be set. For other inputs, PADnINPEN should be cleared. PADnPULL should be cleared except in the case of an open drain trigger input.

Table 8-49: ADC Analog Input Configuration

Field	Value	Input	Pad
PAD16FNCSEL	0	ADCSE0	16
PAD29FNCSEL	0	ADCSE1	29
PAD11FNCSEL	0	ADCSE2	11
PAD31FNCSEL	0	ADCSE3	31
PAD32FNCSEL	0	ADCSE4	32
PAD33FNCSEL	0	ADCSE5	33
PAD34FNCSEL	0	ADCSE6	34
PAD35FNCSEL	0	ADCSE7	35
PAD13FNCSEL	0	ADCD0M/SE8	13
PAD12FNCSEL	0	ADCD0P/SE9	12
PAD14FNCSEL	0	ADCD1P	14
PAD15FNCSEL	0	ADCD1M	15

Table 8-50: ADC Trigger Input Configuration

Field	Value	Input	Pad
PAD7FNCSEL	4	TRIG0	7
PAD16FNCSEL	2	TRIG0	16
PAD40FNCSEL	2	TRIG0	40
PAD3FNCSEL	6	TRIG1	3
PAD17FNCSEL	2	TRIG1	17
PAD36FNCSEL	0	TRIG1	36
PAD37FNCSEL	0	TRIG2	37
PAD38FNCSEL	0	TRIG3	38

8.5.10 Implementing Voltage Comparator Connections

Two types of pad connections may be made for the Voltage Comparator (VCOMP) module. Three reference voltages may be used for the comparator negative input as shown in Table 8-51. The voltage to be applied to the comparator positive input are shown in Table 8-52. In each case PADnINPENn and PADnPULL should be cleared. Note that for CMPRF2, this pin is muxed with ADCSE6 allowing for the same reference input to be used for both ADC and VCOMP operations. Additionally, CMPIN0 is muxed with ADCSE0 allowing for the same input to be used for both ADC and VCOMP operations.

Table 8-51: Voltage Comparator Reference Configuration

Field	Value	Input	Pad
PAD19FNCSEL	0	CMPRF0	19
PAD17FNCSEL	0	CMPRF1	17
PAD34FNCSEL	4	CMPRF2	34

NOTE: If voltage comparator and ADC operation are concurrently sampling the CMPIN0/ADCSE0 input, quality of the sample may be degraded and cannot be guaranteed. It is recommended that voltage comparator and ADC operations are sampled independently (time sliced) to avoid any signal quality loss.

Table 8-52: Voltage Comparator Input Configuration

Field	Value	Input	Pad
PAD16FNCSEL	5	CMPIN0	16
PAD18FNCSEL	0	CMPIN1	18

8.5.11 Implementing the Software Debug Port Connections

The software debug clock (SWDCK) and data (SWDIO) must be connected on pads 20 and 21 respectively. PAD20FNCSEL and PAD21FNCSEL must be set to 0, PAD20INPEN and PAD21INPEN must be set, and PAD20PULL and PAD21PULL must be set, which results in a default state of SWDCK low and SWDIO high. Pads 14 and 15 can alternatively be used for SWDCK and SWDIO functionality, respectively. These pads are, however, not selected by default. Using pads 14 and 15 requires PAD14FNCSEL to be set to 6 and PAD15FNCSEL to be set to 6, PAD14INPEN and PAD15INPEN to be set, and PAD14PULL and PAD15PULL to be set.

The optional continuous output signal SWO may be configured on a variety of pads as shown in Table 8-53, and PADnINPEN and PADnPULL should be cleared for the selected pad.

Table 8-53: SWO Configuration

Field	Value	Pad
PAD15FNCSEL	7	15
PAD22FNCSEL	7	22
PAD24FNCSEL	7	24
PAD30FNCSEL	6	30
PAD33FNCSEL	7	33
PAD41FNCSEL	2	41
PAD45FNCSEL	7	45
PAD46FNCSEL	7	46

8.6 GPIO Registers

General Purpose IO
INSTANCE 0 BASE ADDRESS:0x40010000
This is the detailed description of the general purpose I/O (GPIO) block, as well as for the PAD multi-plexor. Note that GPIO interrupt bits are edge triggered.

WARNING: If an interrupt bit is cleared while the combination of polarity and input are still asserted then this bit will not set again.

8.6.1 Register Memory Map

Table 8-54: GPIO Register Map

Address(es)	Registered Name	Description
0x40010000	PADREGA	Pad Configuration Register A
0x40010004	PADREGB	Pad Configuration Register B
0x40010008	PADREGC	Pad Configuration Register C
0x4001000C	PADREGD	Pad Configuration Register D
0x40010010	PADREGE	Pad Configuration Register E
0x40010014	PADREGF	Pad Configuration Register F
0x40010018	PADREGG	Pad Configuration Register G
0x4001001C	PADREGH	Pad Configuration Register H
0x40010020	PADREGI	Pad Configuration Register I
0x40010024	PADREGJ	Pad Configuration Register J
0x40010028	PADREGK	Pad Configuration Register K
0x4001002C	PADREGL	Pad Configuration Register L
0x40010030	PADREGM	Pad Configuration Register M
0x40010040	CFGA	GPIO Configuration Register A
0x40010044	CFGB	GPIO Configuration Register B
0x40010048	CFGC	GPIO Configuration Register C
0x4001004C	CFGD	GPIO Configuration Register D
0x40010050	CFGE	GPIO Configuration Register E
0x40010054	CFGF	GPIO Configuration Register F
0x40010058	CFGG	GPIO Configuration Register G
0x40010060	PADKEY	Key Register for all pad configuration registers
0x40010080	RDA	GPIO Input Register A
0x40010084	RDB	GPIO Input Register B
0x40010088	WTA	GPIO Output Register A
0x4001008C	WTB	GPIO Output Register B
0x40010090	WTSA	GPIO Output Register A Set
0x40010094	WTSB	GPIO Output Register B Set
0x40010098	WTCA	GPIO Output Register A Clear

Table 8-54: GPIO Register Map *(Continued)*

Address(es)	Registered Name	Description
0x4001009C	WTCB	GPIO Output Register B Clear
0x400100A0	ENA	GPIO Enable Register A
0x400100A4	ENB	GPIO Enable Register B
0x400100A8	ENSA	GPIO Enable Register A Set
0x400100AC	ENSB	GPIO Enable Register B Set
0x400100B4	ENCA	GPIO Enable Register A Clear
0x400100B8	ENCB	GPIO Enable Register B Clear
0x400100BC	STMRCAP	STIMER Capture Control
0x400100C0	IOM0IRQ	IOM0 Flow Control IRQ Select
0x400100C4	IOM1IRQ	IOM1 Flow Control IRQ Select
0x400100C8	IOM2IRQ	IOM2 Flow Control IRQ Select
0x400100CC	IOM3IRQ	IOM3 Flow Control IRQ Select
0x400100D0	IOM4IRQ	IOM4 Flow Control IRQ Select
0x400100D4	IOM5IRQ	IOM5 Flow Control IRQ Select
0x400100D8	LOOPBACK	IOM to IOS Loopback Control
0x400100DC	GPIOOBS	GPIO Observation Mode Sample register
0x400100E0	ALTPADCFGA	Alternate Pad Configuration reg0 (Pads 3,2,1,0)
0x400100E4	ALTPADCFGB	Alternate Pad Configuration reg1 (Pads 7,6,5,4)
0x400100E8	ALTPADCFGC	Alternate Pad Configuration reg2 (Pads 11,10,9,8)
0x400100EC	ALTPADCFGD	Alternate Pad Configuration reg3 (Pads15,14,13,12)
0x400100F0	ALTPADCFGE	Alternate Pad Configuration reg4 (Pads 19,18,17,16)
0x400100F4	ALTPADCFGF	Alternate Pad Configuration reg5 (Pads 23,22,21,20)
0x400100F8	ALTPADCFGG	Alternate Pad Configuration reg6 (Pads 27,26,25,24)
0x400100FC	ALTPADCFGH	Alternate Pad Configuration reg7 (Pads 31,30,29,28)
0x40010100	ALTPADCFGI	Alternate Pad Configuration reg8 (Pads 35,34,33,32)
0x40010104	ALTPADCFGJ	Alternate Pad Configuration reg9 (Pads 39,38,37,36)
0x40010108	ALTPADCFGK	Alternate Pad Configuration reg10 (Pads 43,42,41,40)
0x4001010C	ALTPADCFGL	Alternate Pad Configuration reg11 (Pads 47,46,45,44)
0x40010110	ALTPADCFGM	Alternate Pad Configuration reg12 (Pads 49,48)
0x40010200	INT0EN	GPIO Interrupt Registers 31-0: Enable
0x40010204	INT0STAT	GPIO Interrupt Registers 31-0: Status
0x40010208	INT0CLR	GPIO Interrupt Registers 31-0: Clear
0x4001020C	INT0SET	GPIO Interrupt Registers 31-0: Set
0x40010210	INT1EN	GPIO Interrupt Registers 49-32: Enable
0x40010214	INT1STAT	GPIO Interrupt Registers 49-32: Status
0x40010218	INT1CLR	GPIO Interrupt Registers 49-32: Clear
0x4001021C	INT1SET	GPIO Interrupt Registers 49-32: Set

8.6.2 GPIO Registers

8.6.2.1 PADREGA Register

Pad Configuration Register A
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40010000
This register controls the pad configuration controls for PAD3 through PAD0.
Writes to this register must be unlocked by the PADKEY register.

Table 8-55: PADREGA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD3FNCSEL			PAD3STRNG	PAD3INPEN	PAD3PULL	RSVD		PAD2FNCSEL			PAD2STRNG	PAD2INPEN	PAD2PULL	PAD1RSEL		PAD1FNCSEL			PAD1STRNG	PAD1INPEN	PAD1PULL	PAD0RSEL		PAD0FNCSEL			PAD0STRNG	PAD0INPEN	PAD0PULL

Table 8-56: PADREGA Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD3FNCSEL	0x3	RW	Pad 3 function select UA0RTS = 0x0 - Configure as the UART0 RTS output SLnCE = 0x1 - Configure as the IOSLAVE SPI nCE signal M1nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR1 GPIO3 = 0x3 - Configure as GPIO3 MxnCELB = 0x4 - Configure as the IOSLAVE SPI nCE loopback signal from IOMSTRx M2nCE0 = 0x5 - Configure as the SPI channel 0 nCE signal from IOMSTR2 TRIG1 = 0x6 - Configure as the ADC Trigger 1 signal I2S_WCLK = 0x7 - Configure as the PDM I^2S Word Clock input
26	PAD3STRNG	0x0	RW	Pad 3 drive strength. LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD3INPEN	0x0	RW	Pad 3 input enable. DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD3PULL	0x0	RW	Pad 3 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED

Table 8-56: PADREGA Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
21:19	PAD2FNCSEL	0x3	RW	Pad 2 function select SLWIR3 = 0x0 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO signal SLMOSI = 0x1 - Configure as the IOSLAVE SPI MOSI signal UART0RX = 0x2 - Configure as the UART0 RX input GPIO2 = 0x3 - Configure as GPIO2 MxMOSILB = 0x4 - Configure as the IOSLAVE SPI MOSI loopback signal from IOMSTRx M2MOSI = 0x5 - Configure as the IOMSTR2 SPI MOSI output signal MxWIR3LB = 0x6 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO loopback signal from IOMSTRx M2WIR3 = 0x7 - Configure as the IOMSTR2 SPI 3-wire MOSI/MISO signal
18	PAD2STRNG	0x0	RW	Pad 2 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD2INPEN	0x0	RW	Pad 2 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD2PULL	0x0	RW	Pad 2 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	PAD1RSEL	0x0	RW	Pad 1 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pullu-p is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
13:11	PAD1FNCSEL	0x3	RW	Pad 1 function select SLSDA = 0x0 - Configure as the IOSLAVE I^2C SDA signal SLMISO = 0x1 - Configure as the IOSLAVE SPI MISO signal UART0TX = 0x2 - Configure as the UART0 TX output signal GPIO1 = 0x3 - Configure as GPIO1 MxMISOLB = 0x4 - Configure as the IOSLAVE SPI MISO loopback signal from IOMSTRx M2MISO = 0x5 - Configure as the IOMSTR2 SPI MISO input signal MxSDALB = 0x6 - Configure as the IOSLAVE I^2C SDA loopback signal from IOMSTRx M2SDA = 0x7 - Configure as the IOMSTR2 I^2C Serial data I/O signal
10	PAD1STRNG	0x0	RW	Pad 1 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD1INPEN	0x0	RW	Pad 1 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD1PULL	0x0	RW	Pad 1 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	PAD0RSEL	0x0	RW	Pad 0 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms

Table 8-56: PADREGA Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
5:3	PAD0FNCSEL	0x3	RW	Pad 0 function select SLSCL = 0x0 - Configure as the IOSLAVE I^2C SCL signal SLSCK = 0x1 - Configure as the IOSLAVE SPI SCK signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO0 = 0x3 - Configure as GPIO0 MxSCKLB = 0x4 - Configure as the IOSLAVE SPI SCK loopback signal from IOMSTRx M2SCK = 0x5 - Configure as the IOMSTR2 SPI SCK output MxSCLLB = 0x6 - Configure as the IOSLAVE I^2C SCL loopback signal from IOMSTRx M2SCL = 0x7 - Configure as the IOMSTR2 I^2C SCL clock I/O signal
2	PAD0STRNG	0x0	RW	Pad 0 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD0INPEN	0x0	RW	Pad 0 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD0PULL	0x0	RW	Pad 0 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.2 PADREGB Register

Pad Configuration Register B
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40010004
This register controls the pad configuration controls for PAD7 through PAD4.
Writes to this register must be unlocked by the PADKEY register.

Table 8-57: PADREGB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	PAD7FNCSEL				PAD7STRNG	PAD7INPEN	PAD7PULL	PAD6RSEL		PAD6FNCSEL			PAD6STRNG	PAD6INPEN	PAD6PULL	PAD5RSEL		PAD5FNCSEL			PAD5STRNG	PAD5INPEN	PAD5PULL	PAD4PWRDN	RSVD	PAD4FNCSEL			PAD4STRNG	PAD4INPEN	PAD4PULL

Table 8-58: PADREGB Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD7FNCSEL	0x3	RW	Pad 7 function select M0WIR3 = 0x0 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO signal M0MOSI = 0x1 - Configure as the IOMSTR0 SPI MOSI signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO7 = 0x3 - Configure as GPIO7 TRIG0 = 0x4 - Configure as the ADC Trigger 0 signal UART0TX = 0x5 - Configure as the UART0 TX output signal SLWIR3LB = 0x6 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO loopback signal from IOSLAVE M1nCE1 = 0x7 - Configure as the SPI channel 1 nCE signal from IOMSTR1
26	PAD7STRNG	0x0	RW	Pad 7 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD7INPEN	0x0	RW	Pad 7 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD7PULL	0x0	RW	Pad 7 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	PAD6RSEL	0x0	RW	Pad 6 pullup resistor selection. PULL1_5K = 0x0 - Pullup is ~1.5 KOhms PULL6K = 0x1 - Pullup is ~6 KOhms PULL12K = 0x2 - Pullup is ~12 KOhms PULL24K = 0x3 - Pullup is ~24 KOhms
21:19	PAD6FNCSEL	0x3	RW	Pad 6 function select M0SDA = 0x0 - Configure as the IOMSTR0 I^2C SDA signal M0MISO = 0x1 - Configure as the IOMSTR0 SPI MISO signal UA0CTS = 0x2 - Configure as the UART0 CTS input signal GPIO6 = 0x3 - Configure as GPIO6 SLMISOLB = 0x4 - Configure as the IOMSTR0 SPI MISO loopback signal from IOSLAVE M1nCE0 = 0x5 - Configure as the SPI channel 0 nCE signal from IOMSTR1 SLSDALB = 0x6 - Configure as the IOMSTR0 I^2C SDA loopback signal from IOSLAVE I2S_DAT = 0x7 - Configure as the PDM I^2S Data output signal
18	PAD6STRNG	0x0	RW	Pad 6 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD6INPEN	0x0	RW	Pad 6 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD6PULL	0x0	RW	Pad 6 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

Table 8-58: PADREGB Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
15:14	PAD5RSEL	0x0	RW	Pad 5 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
13:11	PAD5FNCSEL	0x3	RW	Pad 5 function select M0SCL = 0x0 - Configure as the IOMSTR0 I^2C SCL signal M0SCK = 0x1 - Configure as the IOMSTR0 SPI SCK signal UA0RTS = 0x2 - Configure as the UART0 RTS signal output GPIO5 = 0x3 - Configure as GPIO5 M0SCKLB = 0x4 - Configure as the IOMSTR0 SPI SCK loopback signal from IOSLAVE RSV = 0x5 - Reserved for future use M0SCLLB = 0x6 - Configure as the IOMSTR0 I^2C SCL loopback signal from IOSLAVE M1nCE2 = 0x7 - Configure as the SPI Channel 2 nCE signal from IOMSTR1
10	PAD5STRNG	0x0	RW	Pad 5 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD5INPEN	0x0	RW	Pad 5 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD5PULL	0x0	RW	Pad 5 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7	PAD4PWRDN	0x0	RW	Pad 4 VSS power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (switch to GND)
6	RSVD	0x0	RO	RESERVED
5:3	PAD4FNCSEL	0x3	RW	Pad 4 function select UA0CTS = 0x0 - Configure as the UART0 CTS input signal SLINT = 0x1 - Configure as the IOSLAVE interrupt out signal M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0 GPIO4 = 0x3 - Configure as GPIO4 SLINTGP = 0x4 - Configure as the IOSLAVE interrupt loopback signal M2nCE5 = 0x5 - Configure as the SPI channel 5 nCE signal from IOMSTR2 CLKOUT = 0x6 - Configure as the CLKOUT signal 32khz_XT = 0x7 - Configure as the 32kHz crystal output signal
2	PAD4STRNG	0x0	RW	Pad 4 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD4INPEN	0x0	RW	Pad 4 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD4PULL	0x0	RW	Pad 4 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.3 PADREGC Register

Pad Configuration Register C
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40010008
This register controls the pad configuration controls for PAD11 through PAD8.
Writes to this register must be unlocked by the PADKEY register.

Table 8-59: PADREGC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	PAD11FNCSEL				PAD11STRNG	PAD11INPEN	PAD11PULL	RSVD		PAD10FNCSEL			PAD10STRNG	PAD10INPEN	PAD10PULL	PAD9RSEL		PAD9FNCSEL			PAD9STRNG	PAD9INPEN	PAD9PULL	PAD8RSEL		PAD8FNCSEL			PAD8STRNG	PAD8INPEN	PAD8PULL

Table 8-60: PADREGC Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD11FNCSEL	0x3	RW	Pad 11 function select ADCSE2 = 0x0 - Configure as the analog input for ADC single ended input 2 M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0 CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO11 = 0x3 - Configure as GPIO11 M2nCE7 = 0x4 - Configure as the SPI channel 7 nCE signal from IOMSTR2 UA1CTS = 0x5 - Configure as the UART1 CTS input signal UART0RX = 0x6 - Configure as the UART0 RX input signal PDM_DATA = 0x7 - Configure as the PDM Data input signal
26	PAD11STRNG	0x0	RW	Pad 11 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD11INPEN	0x0	RW	Pad 11 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD11PULL	0x0	RW	Pad 11 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED

Table 8-60: PADREGC Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
21:19	PAD10FNCSEL	0x3	RW	Pad 10 function select M1WIR3 = 0x0 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO signal M1MOSI = 0x1 - Configure as the IOMSTR1 SPI MOSI signal M0nCE6 = 0x2 - Configure as the SPI channel 6 nCE signal from IOMSTR0 GPIO10 = 0x3 - Configure as GPIO10 M2nCE6 = 0x4 - Configure as the SPI channel 6 nCE signal from IOMSTR2 UA1RTS = 0x5 - Configure as the UART1 RTS output signal M4nCE4 = 0x6 - Configure as the SPI channel 4 nCE signal from the IOMSTR4 SLWIR3LB = 0x7 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO loopback signal from IOSLAVE
18	PAD10STRNG	0x0	RW	Pad 10 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD10INPEN	0x0	RW	Pad 10 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD10PULL	0x0	RW	Pad 10 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	PAD9RSEL	0x0	RW	Pad 9 pull-up resistor selection PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
13:11	PAD9FNCSEL	0x3	RW	Pad 9 function select M1SDA = 0x0 - Configure as the IOMSTR1 I^2C SDA signal M1MISO = 0x1 - Configure as the IOMSTR1 SPI MISO signal M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0 GPIO9 = 0x3 - Configure as GPIO9 M4nCE5 = 0x4 - Configure as the SPI channel 5 nCE signal from IOMSTR4 SLMISOLB = 0x5 - Configure as the IOMSTR1 SPI MISO loopback signal from IOSLAVE UART1RX = 0x6 - Configure as UART1 RX input signal SLSDALB = 0x7 - Configure as the IOMSTR1 I^2C SDA loopback signal from IOSLAVE
10	PAD9STRNG	0x0	RW	Pad 9 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD9INPEN	0x0	RW	Pad 9 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD9PULL	0x0	RW	Pad 9 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

Table 8-60: PADREGC Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
7:6	PAD8RSEL	0x0	RW	Pad 8 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
5:3	PAD8FNCSEL	0x3	RW	Pad 8 function select M1SCL = 0x0 - Configure as the IOMSTR1 I^2C SCL signal M1SCK = 0x1 - Configure as the IOMSTR1 SPI SCK signal M0nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR0 GPIO8 = 0x3 - Configure as GPIO8 M2nCE4 = 0x4 - Configure as the SPI channel 4 nCE signal from IOMSTR2 M1SCKLB = 0x5 - Configure as the IOMSTR1 SPI SCK loopback signal from IOSLAVE UART1TX = 0x6 - Configure as the UART1 TX output signal M1SCLLB = 0x7 - Configure as the IOMSTR1 I^2C SCL loopback signal from IOSLAVE
2	PAD8STRNG	0x0	RW	Pad 8 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD8INPEN	0x0	RW	Pad 8 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD8PULL	0x0	RW	Pad 8 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.4 PADREGD Register

Pad Configuration Register D
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4001000C
This register controls the pad configuration controls for PAD15 through PAD12.
Writes to this register must be unlocked by the PADKEY register.

Table 8-61: PADREGD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD15FNCSEL			PAD15STRNG	PAD15INPEN	PAD15PULL	RSVD		PAD14FNCSEL			PAD14STRNG	PAD14INPEN	PAD14PULL	RSVD		PAD13FNCSEL			PAD13STRNG	PAD13INPEN	PAD13PULL	RSVD		PAD12FNCSEL			PAD12STRNG	PAD12INPEN	PAD12PULL

Table 8-62: PADREGD Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD15FNCSEL	0x3	RW	Pad 15 function select ADCD1N = 0x0 - Configure as the analog ADC differential pair 1 N input signal M1nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR1 UART1RX = 0x2 - Configure as the UART1 RX signal GPIO15 = 0x3 - Configure as GPIO15 M2nCE2 = 0x4 - Configure as the SPI Channel 2 nCE signal from IOMSTR2 RSV = 0x5 - Function reserved for future use SWDIO = 0x6 - Configure as an alternate port for the SWDIO I/O signal SWO = 0x7 - Configure as an SWO (Serial Wire Trace output)
26	PAD15STRNG	0x0	RW	Pad 15 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD15INPEN	0x0	RW	Pad 15 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD15PULL	0x0	RW	Pad 15 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD14FNCSEL	0x3	RW	Pad 14 function select ADCD1P = 0x0 - Configure as the analog ADC differential pair 1 P input signal M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1 UART1TX = 0x2 - Configure as the UART1 TX output signal GPIO14 = 0x3 - Configure as GPIO14 M2nCE1 = 0x4 - Configure as the SPI channel 1 nCE signal from IOMSTR2 RSV = 0x5 - Function reserved for future use SWDCK = 0x6 - Configure as the alternate input for the SWDCK input signal 32khz_XT = 0x7 - Configure as the 32kHz crystal output signal
18	PAD14STRNG	0x0	RW	Pad 14 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD14INPEN	0x0	RW	Pad 14 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD14PULL	0x0	RW	Pad 14 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED

Table 8-62: PADREGD Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
13:11	PAD13FNCSEL	0x3	RW	Pad 13 function select ADCD0PSE8 = 0x0 - Configure as the ADC Differential pair 0 P, or Single Ended input 8 analog input signal. Determination of the D0P vs SE8 usage is done when the particular channel is selected within the ADC module M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1 TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0 GPIO13 = 0x3 - Configure as GPIO13 M2nCE3 = 0x4 - Configure as the SPI channel 3 nCE signal from IOMSTR2 RSV = 0x5 - Function reserved for future use UA0RTS = 0x6 - Configure as the UART0 RTS signal output UART1RX = 0x7 - Configure as the UART1 RX input signal
10	PAD13STRNG	0x0	RW	Pad 13 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD13INPEN	0x0	RW	Pad 13 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD13PULL	0x0	RW	Pad 13 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD12FNCSEL	0x3	RW	Pad 12 function select ADCD0NSE9 = 0x0 - Configure as the ADC Differential pair 0 N, or Single Ended input 9 analog input signal. Determination of the D0N vs SE9 usage is done when the particular channel is selected within the ADC module M1nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR1 TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0 GPIO12 = 0x3 - Configure as GPIO12 CLKOUT = 0x4 - Configure as CLKOUT signal PDM_CLK = 0x5 - Configure as the PDM CLK output signal UA0CTS = 0x6 - Configure as the UART0 CTS input signal UART1TX = 0x7 - Configure as the UART1 TX output signal
2	PAD12STRNG	0x0	RW	Pad 12 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD12INPEN	0x0	RW	Pad 12 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD12PULL	0x0	RW	Pad 12 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.5 PADREGE Register

Pad Configuration Register E
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40010010
This register controls the pad configuration controls for PAD19 through PAD16. Writes to this register must be unlocked by the PADKEY register.

Table 8-63: PADREGE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD19FNCSEL			PAD19STRNG	PAD19INPEN	PAD19PULL	RSVD		PAD18FNCSEL			PAD18STRNG	PAD18INPEN	PAD18PULL	RSVD		PAD17FNCSEL			PAD17STRNG	PAD17INPEN	PAD17PULL	RSVD		PAD16FNCSEL			PAD16STRNG	PAD16INPEN	PAD16PULL

Table 8-64: PADREGE Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD19FNCSEL	0x3	RW	Pad 19 function select CMPRF0 = 0x0 - Configure as the analog comparator reference 0 signal M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO19 = 0x3 - Configure as GPIO19 TCTA1 = 0x4 - Configure as the input/output signal from CTIMER A1 ANATEST1 = 0x5 - Configure as the ANATEST1 I/O signal UART1RX = 0x6 - Configure as the UART1 RX input signal I2S_BCLK = 0x7 - Configure as the PDM I^2S Byte clock input signal
26	PAD19STRNG	0x0	RW	Pad 19 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD19INPEN	0x0	RW	Pad 19 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD19PULL	0x0	RW	Pad 19 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD18FNCSEL	0x3	RW	Pad 18 function select CMPIN1 = 0x0 - Configure as the analog comparator input 1 signal M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1 GPIO18 = 0x3 - Configure as GPIO18 M4nCE1 = 0x4 - Configure as the SPI nCE channel 1 from IOMSTR4 ANATEST2 = 0x5 - Configure as ANATEST2 I/O signal UART1TX = 0x6 - Configure as UART1 TX output signal 32khz_XT = 0x7 - Configure as the 32kHz output clock from the crystal

Table 8-64: PADREGE Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
18	PAD18STRNG	0x0	RW	Pad 18 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD18INPEN	0x0	RW	Pad 18 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD18PULL	0x0	RW	Pad 18 pull- up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD17FNCSEL	0x3	RW	Pad 17 function select CMPRF1 = 0x0 - Configure as the analog comparator reference signal 1 input signal M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOM-STR0 TRIG1 = 0x2 - Configure as the ADC Trigger 1 signal GPIO17 = 0x3 - Configure as GPIO17 M4nCE3 = 0x4 - Configure as the SPI channel 3 nCE signal from IOM-STR4 RSV = 0x5 - Function reserved for future use UART0RX = 0x6 - Configure as UART0 RX input signal UA1CTS = 0x7 - Configure as UART1 CTS input signal
10	PAD17STRNG	0x0	RW	Pad 17 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD17INPEN	0x0	RW	Pad 17 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD17PULL	0x0	RW	Pad 17 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD16FNCSEL	0x3	RW	Pad 16 function select ADCSE0 = 0x0 - Configure as the analog ADC single ended port 0 input signal M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOM-STR0 TRIG0 = 0x2 - Configure as the ADC Trigger 0 signal GPIO16 = 0x3 - Configure as GPIO16 M2nCE3 = 0x4 - Configure as SPI channel 3 nCE for IOMSTR2 CMPIN0 = 0x5 - Configure as comparator input 0 signal UART0TX = 0x6 - Configure as UART0 TX output signal UA1RTS = 0x7 - Configure as UART1 RTS output signal
2	PAD16STRNG	0x0	RW	Pad 16 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 8-64: PADREGE Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	PAD16INPEN	0x0	RW	Pad 16 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD16PULL	0x0	RW	Pad 16 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.6 PADREGF Register

Pad Configuration Register F
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x40010014
This register controls the pad configuration controls for PAD23 through PAD20. Writes to this register must be unlocked by the PADKEY register.

Table 8-65: PADREGF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD23FNCSEL			PAD23STRNG	PAD23INPEN	PAD23PULL	PAD22PWRUP	RSVD		PAD22FNCSEL		PAD22STRNG	PAD22INPEN	PAD22PULL	RSVD		PAD21FNCSEL			PAD21STRNG	PAD21INPEN	PAD21PULL	RSVD			PAD20FNCSEL			PAD20STRNG	PAD20INPEN	

Table 8-66: PADREGF Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD23FNCSEL	0x3	RW	Pad 23 function select UART0RX = 0x0 - Configure as the UART0 RX signal M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOM-STR0 TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3 GPIO23 = 0x3 - Configure as GPIO23 PDM_DATA = 0x4 - Configure as PDM Data input to the PDM module CMPOUT = 0x5 - Configure as voltage comparator output TCTB1 = 0x6 - Configure as the input/output signal from CTIMER B1 UNDEF7 = 0x7 - Undefined/should not be used
26	PAD23STRNG	0x0	RW	Pad 23 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD23INPEN	0x0	RW	Pad 23 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD23PULL	0x0	RW	Pad 23 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

Table 8-66: PADREGF Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
23	PAD22PWRUP	0x0	RW	Pad 22 upper power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled
22	RSVD	0x0	RO	RESERVED
21:19	PAD22FNCSEL	0x3	RW	Pad 22 function select UART0TX = 0x0 - Configure as the UART0 TX signal M1nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOM-STR1 TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3 GPIO22 = 0x3 - Configure as GPIO22 PDM_CLK = 0x4 - Configure as the PDM CLK output UNDEF5 = 0x5 - Undefined/should not be used TCTB1 = 0x6 - Configure as the input/output signal from CTIMER B1 SWO = 0x7 - Configure as the serial trace data output signal
18	PAD22STRNG	0x0	RW	Pad 22 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD22INPEN	0x0	RW	Pad 22 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD22PULL	0x0	RW	Pad 22 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD21FNCSEL	0x0	RW	Pad 21 function select SWDIO = 0x0 - Configure as the serial wire debug data signal M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOM-STR1 TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2 GPIO21 = 0x3 - Configure as GPIO21 UART0RX = 0x4 - Configure as UART0 RX input signal UART1RX = 0x5 - Configure as UART1 RX input signal UNDEF6 = 0x6 - Undefined/should not be used UNDEF7 = 0x7 - Undefined/should not be used
10	PAD21STRNG	0x0	RW	Pad 21 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD21INPEN	0x1	RW	Pad 21 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD21PULL	0x0	RW	Pad 21 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED

Table 8-66: PADREGF Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
5:3	PAD20FNCSEL	0x0	RW	Pad 20 function select SWDCK = 0x0 - Configure as the serial wire debug clock signal M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOM-STR1 TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2 GPIO20 = 0x3 - Configure as GPIO20 UART0TX = 0x4 - Configure as UART0 TX output signal UART1TX = 0x5 - Configure as UART1 TX output signal UNDEF6 = 0x6 - Undefined/should not be used UNDEF7 = 0x7 - Undefined/should not be used
2	PAD20STRNG	0x0	RW	Pad 20 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD20INPEN	0x1	RW	Pad 20 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD20PULL	0x0	RW	Pad 20 pull-down enable DIS = 0x0 - Pull-down disabled EN = 0x1 - Pull-down enabled

8.6.2.7 PADREGG Register

Pad Configuration Register G
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x40010018
This register controls the pad configuration controls for PAD27 through PAD24.
Writes to this register must be unlocked by the PADKEY register.

Table 8-67: PADREGG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PAD27RSEL		PAD27FNCSEL			PAD27STRNG	PAD27INPEN	PAD27PULL	RSVD		PAD26FNCSEL			PAD26STRNG	PAD26INPEN	PAD26PULL	PAD25RSEL		PAD25FNCSEL			PAD25STRNG	PAD25INPEN	PAD25PULL	RSVD		PAD24FNCSEL			PAD24STRNG	PAD24INPEN	PAD24PULL

Table 8-68: PADREGG Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD27RSEL	0x0	RW	Pad 27 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
29:27	PAD27FNCSEL	0x3	RW	Pad 27 function select EXTHF = 0x0 - Configure as the external HFRC oscillator inputRSV = 0x0 - Function reserved for future use M1nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR1 TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1 GPIO27 = 0x3 - Configure as GPIO27 M2SCL = 0x4 - Configure as I^2C clock I/O signal from IOMSTR2 M2SCK = 0x5 - Configure as SPI clock output signal from IOMSTR2 M2SCKLB = 0x6 - Configure as IOMSTR2 SPI SCK loopback signal from IOSLAVE M2SCLLB = 0x7 - Configure as IOMSTR2 I^2C SCL loopback signal from IOSLAVE
26	PAD27STRNG	0x0	RW	Pad 27 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD27INPEN	0x0	RW	Pad 27 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD27PULL	0x0	RW	Pad 27 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD26FNCSEL	0x3	RW	Pad 26 function select EXTLF = 0x0 - Configure as the external LFRC oscillator inputRSV = 0x0 - Function reserved for future use M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0 TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0 GPIO26 = 0x3 - Configure as GPIO26 M2nCE0 = 0x4 - Configure as the SPI channel 0 nCE signal from IOMSTR2 TCTA1 = 0x5 - Configure as the input/output signal from CTIMER A1 M5nCE1 = 0x6 - Configure as the SPI channel 1 nCE signal from IOMSTR5 M3nCE0 = 0x7 - Configure as the SPI channel 0 nCE signal from IOMSTR3
18	PAD26STRNG	0x0	RW	Pad 26 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD26INPEN	0x0	RW	Pad 26 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD26PULL	0x0	RW	Pad 26 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

Table 8-68: PADREGG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
15:14	PAD25RSEL	0x0	RW	Pad 25 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
13:11	PAD25FNCSEL	0x3	RW	Pad 25 function select EXTXT = 0x0 - Configure as the external XTAL oscillator inputRSV = 0x0 - Function reserved for future use M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0 GPIO25 = 0x3 - Configure as GPIO25 M2SDA = 0x4 - Configure as the IOMSTR2 I^2C Serial data I/O signal M2MISO = 0x5 - Configure as the IOMSTR2 SPI MISO input signal SLMISOLB = 0x6 - Configure as the IOMSTR0 SPI MISO loopback signal from IOSLAVE SLSDALB = 0x7 - Configure as the IOMSTR0 I^2C SDA loopback signal from IOSLAVE
10	PAD25STRNG	0x0	RW	Pad 25 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD25INPEN	0x0	RW	Pad 25 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD25PULL	0x0	RW	Pad 25 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD24FNCSEL	0x3	RW	Pad 24 function select M2nCE1 = 0x0 - Configure as the SPI channel 1 nCE signal from IOMSTR2 M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0 CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO24 = 0x3 - Configure as GPIO24 M5nCE0 = 0x4 - Configure as the SPI channel 0 nCE signal from IOMSTR5 TCTA1 = 0x5 - Configure as the input/output signal from CTIMER A1 I2S_BCLK = 0x6 - Configure as the PDM I^2S Byte clock input signal SWO = 0x7 - Configure as the serial trace data output signal
2	PAD24STRNG	0x0	RW	Pad 24 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD24INPEN	0x0	RW	Pad 24 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD24PULL	0x0	RW	Pad 24 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.8 PADREGH Register

Pad Configuration Register H
OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4001001C
This register controls the pad configuration controls for PAD31 through PAD28. Writes to this register must be unlocked by the PADKEY register.

Table 8-69: PADREGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	PAD31FNCSEL				PAD31STRNG	PAD31INPEN	PAD31PULL	RSVD		PAD30FNCSEL			PAD30STRNG	PAD30INPEN	PAD30PULL	RSVD		PAD29FNCSEL			PAD29STRNG	PAD29INPEN	PAD29PULL	RSVD		PAD28FNCSEL			PAD28STRNG	PAD28INPEN	PAD28PULL

Table 8-70: PADREGH Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD31FNCSEL	0x3	RW	Pad 31 function select ADCSE3 = 0x0 - Configure as the analog input for ADC single ended input 3 M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0 TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3 GPIO31 = 0x3 - Configure as GPIO31 UART0RX = 0x4 - Configure as the UART0 RX input signal TCTB1 = 0x5 - Configure as the input/output signal from CTIMER B1 UNDEF6 = 0x6 - Undefined/should not be used UNDEF7 = 0x7 - Undefined/should not be used
26	PAD31STRNG	0x0	RW	Pad 31 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD31INPEN	0x0	RW	Pad 31 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD31PULL	0x0	RW	Pad 31 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD30FNCSEL	0x3	RW	Pad 30 function select UNDEF0 = 0x0 - Undefined/should not be used M1nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR1 TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2 GPIO30 = 0x3 - Configure as GPIO30 UART0TX = 0x4 - Configure as UART0 TX output signal UA1RTS = 0x5 - Configure as UART1 RTS output signal UNDEF6 = 0x6 - Undefined/should not be used I2S_DAT = 0x7 - Configure as the PDM I^2S Data output signal
18	PAD30STRNG	0x0	RW	Pad 30 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 8-70: PADREGH Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
17	PAD30INPEN	0x0	RW	Pad 30 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD30PULL	0x0	RW	Pad 30 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD29FNCSEL	0x3	RW	Pad 29 function select ADCSE1 = 0x0 - Configure as the analog input for ADC single ended input 1 M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR1 TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2 GPIO29 = 0x3 - Configure as GPIO29 UA0CTS = 0x4 - Configure as the UART0 CTS signal UA1CTS = 0x5 - Configure as the UART1 CTS signal M4nCE0 = 0x6 - Configure as the SPI channel 0 nCE signal from IOMSTR4 PDM_DATA = 0x7 - Configure as PDM DATA input
10	PAD29STRNG	0x0	RW	Pad 29 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD29INPEN	0x0	RW	Pad 29 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD29PULL	0x0	RW	Pad 29 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD28FNCSEL	0x3	RW	Pad 28 function select I2S_WCLK = 0x0 - Configure as the PDM I^2S Word Clock input M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR1 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO28 = 0x3 - Configure as GPIO28 M2WIR3 = 0x4 - Configure as the IOMSTR2 SPI 3-wire MOSI/MISO signal M2MOSI = 0x5 - Configure as the IOMSTR2 SPI MOSI output signal M5nCE3 = 0x6 - Configure as the SPI channel 3 nCE signal from IOMSTR5 SLWIR3LB = 0x7 - Configure as the IOMSTR2 SPI 3-wire MOSI/MISO loopback signal from IOSLAVE
2	PAD28STRNG	0x0	RW	Pad 28 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD28INPEN	0x0	RW	Pad 28 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD28PULL	0x0	RW	Pad 28 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.9 PADREGI Register

Pad Configuration Register I
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x40010020
This register controls the pad configuration controls for PAD35 through PAD32. Writes to this register must be unlocked by the PADKEY register.

Table 8-71: PADREGI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD35FNCSEL			PAD35STRNG	PAD35INPEN	PAD35PULL	RSVD		PAD34FNCSEL			PAD34STRNG	PAD34INPEN	PAD34PULL	RSVD		PAD33FNCSEL			PAD33STRNG	PAD33INPEN	PAD33PULL	RSVD		PAD32FNCSEL			PAD32STRNG	PAD32INPEN	PAD32PULL

Table 8-72: PADREGI Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD35FNCSEL	0x3	RW	Pad 35 function select ADCSE7 = 0x0 - Configure as the analog input for ADC single ended input 7 M1nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR1 UART1TX = 0x2 - Configure as the UART1 TX signal GPIO35 = 0x3 - Configure as GPIO35 M4nCE6 = 0x4 - Configure as the SPI channel 6 nCE signal from IOMSTR4 TCTA1 = 0x5 - Configure as the input/output signal from CTIMER A1 UA0RTS = 0x6 - Configure as the UART0 RTS output M3nCE2 = 0x7 - Configure as the SPI channel 2 nCE signal from IOMSTR3
26	PAD35STRNG	0x0	RW	Pad 35 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD35INPEN	0x0	RW	Pad 35 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD35PULL	0x0	RW	Pad 35 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD34FNCSEL	0x3	RW	Pad 34 function select ADCSE6 = 0x0 - Configure as the analog input for ADC single ended input 6 M0nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR0 M2nCE3 = 0x2 - Configure as the SPI channel 3 nCE signal from IOMSTR2 GPIO34 = 0x3 - Configure as GPIO34 CMPRF2 = 0x4 - Configure as the analog comparator reference 2 signal M3nCE1 = 0x5 - Configure as the SPI channel 1 nCE signal from IOMSTR3 M4nCE0 = 0x6 - Configure as the SPI channel 0 nCE signal from IOMSTR4 M5nCE2 = 0x7 - Configure as the SPI channel 2 nCE signal from IOMSTR5

Table 8-72: PADREGI Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
18	PAD34STRNG	0x0	RW	Pad 34 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD34INPEN	0x0	RW	Pad 34 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD34PULL	0x0	RW	Pad 34 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD33FNCSEL	0x3	RW	Pad 33 function select ADCSE5 = 0x0 - Configure as the analog ADC single ended port 5 input signal M0nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR0 32khz_XT = 0x2 - Configure as the 32kHz crystal output signal GPIO33 = 0x3 - Configure as GPIO33 UNDEF4 = 0x4 - Undefined/should not be used M3nCE7 = 0x5 - Configure as the SPI channel 7 nCE signal from IOMSTR3 TCTB1 = 0x6 - Configure as the input/output signal from CTIMER B1 SWO = 0x7 - Configure as the serial trace data output signal
10	PAD33STRNG	0x0	RW	Pad 33 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD33INPEN	0x0	RW	Pad 33 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD33PULL	0x0	RW	Pad 33 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD32FNCSEL	0x3	RW	Pad 32 function select ADCSE4 = 0x0 - Configure as the analog input for ADC single ended input 4 M0nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR0 TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3 GPIO32 = 0x3 - Configure as GPIO32 UNDEF4 = 0x4 - Undefined/should not be used TCTB1 = 0x5 - Configure as the input/output signal from CTIMER B1 UNDEF6 = 0x6 - Undefined/should not be used UNDEF7 = 0x7 - Undefined/should not be used
2	PAD32STRNG	0x0	RW	Pad 32 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD32INPEN	0x0	RW	Pad 32 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD32PULL	0x0	RW	Pad 32 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

8.6.2.10 PADREGJ Register

Pad Configuration Register J
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x40010024
This register controls the pad configuration controls for PAD39 through PAD36. Writes to this register must be unlocked by the PADKEY register.

Table 8-73: PADREGJ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PAD39RSEL		PAD39FNCSEL			PAD39STRNG	PAD39INPEN	PAD39PULL	RSVD		PAD38FNCSEL			PAD38STRNG	PAD38INPEN	PAD38PULL	RSVD		PAD37FNCSEL			PAD37STRNG	PAD37INPEN	PAD37PULL	RSVD		PAD36FNCSEL			PAD36STRNG	PAD36INPEN	PAD36PULL

Table 8-74: PADREGJ Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD39RSEL	0x0	RW	Pad 39 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
29:27	PAD39FNCSEL	0x3	RW	Pad 39 function select UART0TX = 0x0 - Configure as the UART0 TX Signal UART1TX = 0x1 - Configure as the UART1 TX signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO39 = 0x3 - Configure as GPIO39 M4SCL = 0x4 - Configure as the IOMSTR4 I^2C SCL signal M4SCK = 0x5 - Configure as the IOMSTR4 SPI SCK signal M4SCKLB = 0x6 - Configure as the IOMSTR4 SPI SCK loopback signal from IOSLAVE M4SCLLB = 0x7 - Configure as the IOMSTR4 I^2C SCL loopback signal from IOSLAVE
26	PAD39STRNG	0x0	RW	Pad 39 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD39INPEN	0x0	RW	Pad 39 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD39PULL	0x0	RW	Pad 39 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED

Table 8-74: PADREGJ Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
21:19	PAD38FNCSEL	0x3	RW	Pad 38 function select TRIG3 = 0x0 - Configure as the ADC Trigger 3 signal M1nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR1 UA0CTS = 0x2 - Configure as the UART0 CTS signal GPIO38 = 0x3 - Configure as GPIO38 M3WIR3 = 0x4 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO signal M3MOSI = 0x5 - Configure as the IOMSTR3 SPI MOSI output signal M4nCE7 = 0x6 - Configure as the SPI channel 7 nCE signal from IOMSTR4 SLWIR3LB = 0x7 - Configure as the IOMSTR3 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE
18	PAD38STRNG	0x0	RW	Pad 38 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD38INPEN	0x0	RW	Pad 38 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD38PULL	0x0	RW	Pad 38 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD37FNCSEL	0x3	RW	Pad 37 function select TRIG2 = 0x0 - Configure as the ADC Trigger 2 signal M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1 UA0RTS = 0x2 - Configure as the UART0 RTS signal GPIO37 = 0x3 - Configure as GPIO37 M3nCE4 = 0x4 - Configure as the SPI channel 4 nCE signal from IOMSTR3 M4nCE1 = 0x5 - Configure as the SPI channel 1 nCE signal from IOMSTR4 PDM_CLK = 0x6 - Configure as the PDM CLK output signal TCTA1 = 0x7 - Configure as the input/output signal from CTIMER A1
10	PAD37STRNG	0x0	RW	Pad 37 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD37INPEN	0x0	RW	Pad 37 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD37PULL	0x0	RW	Pad 37 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD36FNCSEL	0x3	RW	Pad 36 function select TRIG1 = 0x0 - Configure as the ADC Trigger 1 signal M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1 UART1RX = 0x2 - Configure as the UART1 RX signal GPIO36 = 0x3 - Configure as GPIO36 32khz_XT = 0x4 - Configure as the 32kHz output clock from the crystal M2nCE0 = 0x5 - Configure as the SPI channel 0 nCE signal from IOMSTR2 UA0CTS = 0x6 - Configure as the UART0 CTS signal M3nCE3 = 0x7 - Configure as the SPI channel 3 nCE signal from IOMSTR3
2	PAD36STRNG	0x0	RW	Pad 36 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 8-74: PADREGJ Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	PAD36INPEN	0x0	RW	Pad 36 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD36PULL	0x0	RW	Pad 36 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.11 PADREGK Register

Pad Configuration Register K
OFFSET: 0x00000028
INSTANCE 0 ADDRESS: 0x40010028
This register controls the pad configuration controls for PAD43 through PAD40.
Writes to this register must be unlocked by the PADKEY register.

Table 8-75: PADREGK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PAD43RSEL			PAD43FNCSEL			PAD43STRNG	PAD43INPEN	PAD43PULL		PAD42RSEL			PAD42FNCSEL			PAD42STRNG	PAD42INPEN	PAD42PULL	PAD41PWRUP	RSVD		PAD41FNCSEL			PAD41STRNG	PAD41INPEN	PAD41PULL		PAD40RSEL		PAD40FNCSEL			PAD40STRNG	PAD40INPEN	PAD40PULL

Table 8-76: PADREGK Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD43RSEL	0x0	RW	Pad 43 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
29:27	PAD43FNCSEL	0x3	RW	Pad 43 function select M2nCE4 = 0x0 - Configure as the SPI channel 4 nCE signal from IOMSTR2 M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0 TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0 GPIO43 = 0x3 - Configure as GPIO43 M3SDA = 0x4 - Configure as the IOMSTR3 I^2C SDA signal M3MISO = 0x5 - Configure as the IOMSTR3 SPI MISO signal SLMISOLB = 0x6 - Configure as the IOMSTR3 SPI MISO loopback signal from IOSLAVE SLSDALB = 0x7 - Configure as the IOMSTR3 I^2C SDA loopback signal from IOSLAVE
26	PAD43STRNG	0x0	RW	Pad 43 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD43INPEN	0x0	RW	Pad 43 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled

Table 8-76: PADREGK Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
24	PAD43PULL	0x0	RW	Pad 43 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	PAD42RSEL	0x0	RW	Pad 42 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
21:19	PAD42FNCSEL	0x3	RW	Pad 42 function select M2nCE2 = 0x0 - Configure as the SPI channel 2 nCE signal from IOMSTR2 M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0 TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0 GPIO42 = 0x3 - Configure as GPIO42 M3SCL = 0x4 - Configure as the IOMSTR3 I^2C SCL clock I/O signal M3SCK = 0x5 - Configure as the IOMSTR3 SPI SCK output M3SCKLB = 0x6 - Configure as the IOMSTR3 SPI clock loopback to the IOSLAVE device M3SCLLB = 0x7 - Configure as the IOMSTR3 I^2C clock loopback to the IOSLAVE device
18	PAD42STRNG	0x0	RW	Pad 42 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD42INPEN	0x0	RW	Pad 42 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD42PULL	0x0	RW	Pad 42 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15	PAD41PWRUP	0x0	RW	Pad 41 upper power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (VDD switch)
14	RSVD	0x0	RO	RESERVED
13:11	PAD41FNCSEL	0x3	RW	Pad 41 function select M2nCE1 = 0x0 - Configure as the SPI channel 1 nCE signal from IOMSTR2 CLKOUT = 0x1 - Configure as the CLKOUT signal SWO = 0x2 - Configure as the serial wire debug SWO signal GPIO41 = 0x3 - Configure as GPIO41 M3nCE5 = 0x4 - Configure as the SPI channel 5 nCE signal from IOMSTR3 M5nCE7 = 0x5 - Configure as the SPI channel 7 nCE signal from IOMSTR5 M4nCE2 = 0x6 - Configure as the SPI channel 2 nCE signal from IOMSTR4 UA0RTS = 0x7 - Configure as the UART0 RTS output
10	PAD41STRNG	0x0	RW	Pad 41 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD41INPEN	0x0	RW	Pad 41 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD41PULL	0x0	RW	Pad 41 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

Table 8-76: PADREGK Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
7:6	PAD40RSEL	0x0	RW	Pad 40 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
5:3	PAD40FNCSEL	0x3	RW	Pad 40 function select UART0RX = 0x0 - Configure as the UART0 RX input signal UART1RX = 0x1 - Configure as the UART1 RX input signal TRIG0 = 0x2 - Configure as the ADC Trigger 0 signal GPIO40 = 0x3 - Configure as GPIO40 M4SDA = 0x4 - Configure as the IOMSTR4 I^2C serial data I/O signal M4MISO = 0x5 - Configure as the IOMSTR4 SPI MISO input signal SLMISOLB = 0x6 - Configure as the IOMSTR4 SPI MISO loopback signal from IOSLAVE SLSDALB = 0x7 - Configure as the IOMSTR4 I^2C SDA loopback signal from IOSLAVE
2	PAD40STRNG	0x0	RW	Pad 40 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD40INPEN	0x0	RW	Pad 40 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD40PULL	0x0	RW	Pad 40 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.12 PADREGL Register

Pad Configuration Register L
OFFSET: 0x0000002C
INSTANCE 0 ADDRESS: 0x4001002C
This register controls the pad configuration controls for PAD47 through PAD44.
Writes to this register must be unlocked by the PADKEY register.

Table 8-77: PADREGL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD		PAD47FNCSEL			PAD47STRNG	PAD47INPEN	PAD47PULL	RSVD		PAD46FNCSEL			PAD46STRNG	PAD46INPEN	PAD46PULL	RSVD		PAD45FNCSEL			PAD45STRNG	PAD45INPEN	PAD45PULL	RSVD		PAD44FNCSEL			PAD44STRNG	PAD44INPEN	PAD44PULL

Table 8-78: PADREGL Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD47FNCSEL	0x3	RW	Pad 47 function select M2nCE5 = 0x0 - Configure as the SPI channel 5 nCE signal from IOMSTR2 M0nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR0 TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2 GPIO47 = 0x3 - Configure as GPIO47 M5WIR3 = 0x4 - Configure as the IOMSTR5 SPI 3-wire MOSI/MISO signal M5MOSI = 0x5 - Configure as the IOMSTR5 SPI MOSI output signal M4nCE5 = 0x6 - Configure as the SPI channel 5 nCE signal from IOMSTR4 SLWIR3LB = 0x7 - Configure as the IOMSTR5 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE
26	PAD47STRNG	0x0	RW	Pad 47 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD47INPEN	0x0	RW	Pad 47 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD47PULL	0x0	RW	Pad 47 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD46FNCSEL	0x3	RW	Pad 46 function select 32khz_XT = 0x0 - Configure as the 32kHz output clock from the crystal M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0 TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2 GPIO46 = 0x3 - Configure as GPIO46 TCTA1 = 0x4 - Configure as the input/output signal from CTIMER A1 M5nCE4 = 0x5 - Configure as the SPI channel 4 nCE signal from IOMSTR5 M4nCE4 = 0x6 - Configure as the SPI channel 4 nCE signal from IOMSTR4 SWO = 0x7 - Configure as the serial wire debug SWO signal
18	PAD46STRNG	0x0	RW	Pad 46 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD46INPEN	0x0	RW	Pad 46 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD46PULL	0x0	RW	Pad 46 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD45FNCSEL	0x3	RW	Pad 45 function select UA1CTS = 0x0 - Configure as the UART1 CTS input signal M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO45 = 0x3 - Configure as GPIO45 M4nCE3 = 0x4 - Configure as the SPI channel 3 nCE signal from IOMSTR4 M3nCE6 = 0x5 - Configure as the SPI channel 6 nCE signal from IOMSTR3 M5nCE5 = 0x6 - Configure as the SPI channel 5 nCE signal from IOMSTR5 TCTA1 = 0x7 - Configure as the input/output signal from CTIMER A1

Table 8-78: PADREGL Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
10	PAD45STRNG	0x0	RW	Pad 45 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD45INPEN	0x0	RW	Pad 45 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD45PULL	0x0	RW	Pad 45 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD44FNCSEL	0x3	RW	Pad 44 function select UA1RTS = 0x0 - Configure as the UART1 RTS output signal M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1 GPIO44 = 0x3 - Configure as GPIO44 M4WIR3 = 0x4 - Configure as the IOMSTR4 SPI 3-wire MOSI/MISO signal M4MOSI = 0x5 - Configure as the IOMSTR4 SPI MOSI signal M5nCE6 = 0x6 - Configure as the SPI channel 6 nCE signal from IOMSTR5 SLWIR3LB = 0x7 - Configure as the IOMSTR4 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE
2	PAD44STRNG	0x0	RW	Pad 44 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD44INPEN	0x0	RW	Pad 44 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD44PULL	0x0	RW	Pad 44 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.13 PADREGM Register

Pad Configuration Register M
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x40010030
This register controls the pad configuration controls for PAD49 through PAD48.
Writes to this register must be unlocked by the PADKEY register.

Table 8-79: PADREGM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																PAD49RSEL		PAD49FNCSEL			PAD49STRNG	PAD49INPEN	PAD49PULL	PAD48RSEL		PAD48FNCSEL			PAD48STRNG	PAD48INPEN	PAD48PULL

Table 8-80: PADREGM Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:14	PAD49RSEL	0x0	RW	Pad 49 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
13:11	PAD49FNCSEL	0x3	RW	Pad 49 function select M2nCE7 = 0x0 - Configure as the SPI channel 7 nCE signal from IOMSTR2 M0nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR0 TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3 GPIO49 = 0x3 - Configure as GPIO49 M5SDA = 0x4 - Configure as the IOMSTR5 I^2C serial data I/O signal M5MISO = 0x5 - Configure as the IOMSTR5 SPI MISO input signal SLMISOLB = 0x6 - Configure as the IOMSTR5 SPI MISO loopback signal from IOSLAVE SLSDALB = 0x7 - Configure as the IOMSTR5 I^2C SDA loopback signal from IOSLAVE
10	PAD49STRNG	0x0	RW	Pad 49 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD49INPEN	0x0	RW	Pad 49 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD49PULL	0x0	RW	Pad 49 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled
7:6	PAD48RSEL	0x0	RW	Pad 48 pull-up resistor selection. PULL1_5K = 0x0 - Pull-up is ~1.5 KOhms PULL6K = 0x1 - Pull-up is ~6 KOhms PULL12K = 0x2 - Pull-up is ~12 KOhms PULL24K = 0x3 - Pull-up is ~24 KOhms
5:3	PAD48FNCSEL	0x3	RW	Pad 48 function select M2nCE6 = 0x0 - Configure as the SPI channel 6 nCE signal from IOMSTR2 M0nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR0 TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3 GPIO48 = 0x3 - Configure as GPIO48 M5SCL = 0x4 - Configure as the IOMSTR5 I^2C SCL clock I/O signal M5SCK = 0x5 - Configure as the IOMSTR5 SPI SCK output M5SCKLB = 0x6 - Configure as the IOMSTR5 SPI clock loopback to the IOSLAVE device M5SCLLB = 0x7 - Configure as the IOMSTR5 I^2C clock loopback to the IOSLAVE device
2	PAD48STRNG	0x0	RW	Pad 48 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 8-80: PADREGM Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	PAD48INPEN	0x0	RW	Pad 48 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD48PULL	0x0	RW	Pad 48 pull-up enable DIS = 0x0 - Pull-up disabled EN = 0x1 - Pull-up enabled

8.6.2.14 CFGA Register

GPIO Configuration Register A
OFFSET: 0x00000040
INSTANCE 0 ADDRESS: 0x40010040
GPIO configuration controls for GPIO[7:0]. Writes to this register must be unlocked by the PADKEY register.

Table 8-81: CFGA Register

31	30 29	28	27	26 25	24	23	22 21	20	19	18 17	16	15	14 13	12 11	10 09	08	07	06 05	04	03 02	01 00
GPIO7INTD	GPIO7OUTCFG	GPIO7INCFG	GPIO6INTD	GPIO6OUTCFG	GPIO6INCFG	GPIO5INTD	GPIO5OUTCFG	GPIO5INCFG	GPIO4INTD	GPIO4OUTCFG	GPIO4INCFG	GPIO3INTD	GPIO3OUTCFG	GPIO3INCFG	GPIO2INTD	GPIO2OUTCFG	GPIO2INCFG	GPIO1INTD	GPIO1OUTCFG	GPIO1INCFG	GPIO0INTD	GPIO0OUTCFG	GPIO0INCFG

Table 8-82: CFGA Register Bits

Bit	Name	Reset	RW	Description
31	GPIO7INTD	0x0	RW	GPIO7 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO7OUTCFG	0x0	RW	GPIO7 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO7INCFG	0x0	RW	GPIO7 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO6INTD	0x0	RW	GPIO6 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO6OUTCFG	0x0	RW	GPIO6 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 8-82: CFGA Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
24	GPIO6INCFG	0x0	RW	GPIO6 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO5INTD	0x0	RW	GPIO5 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO5OUTCFG	0x0	RW	GPIO5 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO5INCFG	0x0	RW	GPIO5 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO4INTD	0x0	RW	GPIO4 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO4OUTCFG	0x0	RW	GPIO4 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO4INCFG	0x0	RW	GPIO4 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO3INTD	0x0	RW	GPIO3 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO3OUTCFG	0x0	RW	GPIO3 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO3INCFG	0x0	RW	GPIO3 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO2INTD	0x0	RW	GPIO2 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO2OUTCFG	0x0	RW	GPIO2 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO2INCFG	0x0	RW	GPIO2 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO1INTD	0x0	RW	GPIO1 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-82: CFGA Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
6:5	GPIO1OUTCFG	0x0	RW	GPIO1 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO1INCFG	0x0	RW	GPIO1 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO0INTD	0x0	RW	GPIO0 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO0OUTCFG	0x0	RW	GPIO0 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO0INCFG	0x0	RW	GPIO0 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.15 CFGB Register

GPIO Configuration Register B
OFFSET: 0x00000044
INSTANCE 0 ADDRESS: 0x40010044
GPIO configuration controls for GPIO[15:8]. Writes to this register must be unlocked by the PADKEY register.

Table 8-83: CFGB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO15INTD	GPIO15OUTCFG		GPIO15INCFG	GPIO14INTD	GPIO14OUTCFG		GPIO14INCFG	GPIO13INTD	GPIO13OUTCFG		GPIO13INCFG	GPIO12INTD	GPIO12OUTCFG		GPIO12INCFG	GPIO11INTD	GPIO11OUTCFG		GPIO11INCFG	GPIO10INTD	GPIO10OUTCFG		GPIO10INCFG	GPIO9INTD	GPIO9OUTCFG		GPIO9INCFG	GPIO8INTD	GPIO8OUTCFG		GPIO8INCFG

Table 8-84: Register Bits

Bit	Name	Reset	RW	Description
31	GPIO15INTD	0x0	RW	GPIO15 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO15OUTCFG	0x0	RW	GPIO15 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 8-84: Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
28	GPIO15INCFG	0x0	RW	GPIO15 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO14INTD	0x0	RW	GPIO14 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO14OUTCFG	0x0	RW	GPIO14 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO14INCFG	0x0	RW	GPIO14 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO13INTD	0x0	RW	GPIO13 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO13OUTCFG	0x0	RW	GPIO13 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO13INCFG	0x0	RW	GPIO13 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO12INTD	0x0	RW	GPIO12 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO12OUTCFG	0x0	RW	GPIO12 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO12INCFG	0x0	RW	GPIO12 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO11INTD	0x0	RW	GPIO11 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO11OUTCFG	0x0	RW	GPIO11 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO11INCFG	0x0	RW	GPIO11 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO10INTD	0x0	RW	GPIO10 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-84: Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
10:9	GPIO10OUTCFG	0x0	RW	GPIO10 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO10INCFG	0x0	RW	GPIO10 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO9INTD	0x0	RW	GPIO9 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO9OUTCFG	0x0	RW	GPIO9 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO9INCFG	0x0	RW	GPIO9 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO8INTD	0x0	RW	GPIO8 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO8OUTCFG	0x0	RW	GPIO8 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO8INCFG	0x0	RW	GPIO8 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.16 CFGC Register

GPIO Configuration Register C
OFFSET: 0x00000048
INSTANCE 0 ADDRESS: 0x40010048
GPIO configuration controls for GPIO[23:16]. Writes to this register must be unlocked by the PADKEY register.

Table 8-85: CFGC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO23INTD	GPIO23OUTCFG		GPIO23INCFG	GPIO22INTD	GPIO22OUTCFG		GPIO22INCFG	GPIO21INTD	GPIO21OUTCFG		GPIO21INCFG	GPIO20INTD	GPIO20OUTCFG		GPIO20INCFG	GPIO19INTD	GPIO19OUTCFG		GPIO19INCFG	GPIO18INTD	GPIO18OUTCFG		GPIO18INCFG	GPIO17INTD	GPIO17OUTCFG		GPIO17INCFG	GPIO16INTD	GPIO16OUTCFG		GPIO16INCFG

Table 8-86: CFGC Register Bits

Bit	Name	Reset	RW	Description
31	GPIO23INTD	0x0	RW	GPIO23 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO23OUTCFG	0x0	RW	GPIO23 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO23INCFG	0x0	RW	GPIO23 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO22INTD	0x0	RW	GPIO22 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO22OUTCFG	0x0	RW	GPIO22 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO22INCFG	0x0	RW	GPIO22 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO21INTD	0x0	RW	GPIO21 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO21OUTCFG	0x0	RW	GPIO21 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO21INCFG	0x1	RW	GPIO21 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO20INTD	0x0	RW	GPIO20 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO20OUTCFG	0x0	RW	GPIO20 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO20INCFG	0x1	RW	GPIO20 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO19INTD	0x0	RW	GPIO19 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-86: CFGC Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
14:13	GPIO19OUTCFG	0x0	RW	GPIO19 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO19INCFG	0x0	RW	GPIO19 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO18INTD	0x0	RW	GPIO18 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO18OUTCFG	0x0	RW	GPIO18 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO18INCFG	0x0	RW	GPIO18 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO17INTD	0x0	RW	GPIO17 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO17OUTCFG	0x0	RW	GPIO17 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO17INCFG	0x0	RW	GPIO17 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO16INTD	0x0	RW	GPIO16 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO16OUTCFG	0x0	RW	GPIO16 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO16INCFG	0x0	RW	GPIO16 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

8.6.2.17 CFGD Register

GPIO Configuration Register D
OFFSET: 0x0000004C
INSTANCE 0 ADDRESS: 0x4001004C
GPIO configuration controls for GPIO[31:24]. Writes to this register must be unlocked by the PADKEY register.

Table 8-87: CFGD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO31INTD	GPIO31OUTCFG		GPIO31INCFG	GPIO30INTD	GPIO30OUTCFG		GPIO30INCFG	GPIO29INTD	GPIO29OUTCFG		GPIO29INCFG	GPIO28INTD	GPIO28OUTCFG		GPIO28INCFG	GPIO27INTD	GPIO27OUTCFG		GPIO27INCFG	GPIO26INTD	GPIO26OUTCFG		GPIO26INCFG	GPIO25INTD	GPIO25OUTCFG		GPIO25INCFG	GPIO24INTD	GPIO24OUTCFG		GPIO24INCFG

Table 8-88: CFGD Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31INTD	0x0	RW	GPIO31 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO31OUTCFG	0x0	RW	GPIO31 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO31INCFG	0x0	RW	GPIO31 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO30INTD	0x0	RW	GPIO30 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO30OUTCFG	0x0	RW	GPIO30 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO30INCFG	0x0	RW	GPIO30 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO29INTD	0x0	RW	GPIO29 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO29OUTCFG	0x0	RW	GPIO29 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO29INCFG	0x0	RW	GPIO29 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO28INTD	0x0	RW	GPIO28 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-88: CFGD Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
18:17	GPIO28OUTCFG	0x0	RW	GPIO28 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO28INCFG	0x0	RW	GPIO28 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO27INTD	0x0	RW	GPIO27 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO27OUTCFG	0x0	RW	GPIO27 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO27INCFG	0x0	RW	GPIO27 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO26INTD	0x0	RW	GPIO26 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO26OUTCFG	0x0	RW	GPIO26 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO26INCFG	0x0	RW	GPIO26 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO25INTD	0x0	RW	GPIO25 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO25OUTCFG	0x0	RW	GPIO25 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO25INCFG	0x0	RW	GPIO25 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO24INTD	0x0	RW	GPIO24 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-88: CFGD Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
2:1	GPIO24OUTCFG	0x0	RW	GPIO24 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO24INCFG	0x0	RW	GPIO24 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.18 CFGE Register

GPIO Configuration Register E
OFFSET: 0x00000050
INSTANCE 0 ADDRESS: 0x40010050
GPIO configuration controls for GPIO[39:32]. Writes to this register must be unlocked by the PADKEY register.

Table 8-89: CFGE Register

31	30 29	28	27	26 25	24	23	22 21	20	19	18 17	16	15	14 13	12 11	10	09 08	07	06 05	04	03 02	01 00
GPIO39INTD	GPIO39OUTCFG	GPIO39INCFG	GPIO38INTD	GPIO38OUTCFG	GPIO38INCFG	GPIO37INTD	GPIO37OUTCFG	GPIO37INCFG	GPIO36INTD	GPIO36OUTCFG	GPIO36INCFG	GPIO35INTD	GPIO35OUTCFG	GPIO35INCFG	GPIO34INTD	GPIO34OUTCFG	GPIO34INCFG	GPIO33INTD	GPIO33OUTCFG	GPIO33INCFG	GPIO32INTD	GPIO32OUTCFG	GPIO32INCFG

Table 8-90: CFGE Register Bits

Bit	Name	Reset	RW	Description
31	GPIO39INTD	0x0	RW	GPIO39 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO39OUTCFG	0x0	RW	GPIO39 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO39INCFG	0x0	RW	GPIO39 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO38INTD	0x0	RW	GPIO38 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO38OUTCFG	0x0	RW	GPIO38 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 8-90: CFGE Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
24	GPIO38INCFG	0x0	RW	GPIO38 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO37INTD	0x0	RW	GPIO37 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO37OUTCFG	0x0	RW	GPIO37 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO37INCFG	0x0	RW	GPIO37 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO36INTD	0x0	RW	GPIO36 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO36OUTCFG	0x0	RW	GPIO36 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO36INCFG	0x0	RW	GPIO36 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO35INTD	0x0	RW	GPIO35 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO35OUTCFG	0x0	RW	GPIO35 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO35INCFG	0x0	RW	GPIO35 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO34INTD	0x0	RW	GPIO34 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO34OUTCFG	0x0	RW	GPIO34 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO34INCFG	0x0	RW	GPIO34 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO33INTD	0x0	RW	GPIO33 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-90: CFGE Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
6:5	GPIO33OUTCFG	0x0	RW	GPIO33 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO33INCFG	0x0	RW	GPIO33 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO32INTD	0x0	RW	GPIO32 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO32OUTCFG	0x0	RW	GPIO32 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO32INCFG	0x0	RW	GPIO32 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.19 CFGF Register

GPIO Configuration Register F
OFFSET: 0x00000054
INSTANCE 0 ADDRESS: 0x40010054
GPIO configuration controls for GPIO[47:40]. Writes to this register must be unlocked by the PADKEY register.

Table 8-91: CFGF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO47INTD	GPIO47OUTCFG		GPIO47INCFG	GPIO46INTD	GPIO46OUTCFG		GPIO46INCFG	GPIO45INTD	GPIO45OUTCFG		GPIO45INCFG	GPIO44INTD	GPIO44OUTCFG		GPIO44INCFG	GPIO43INTD	GPIO43OUTCFG		GPIO43INCFG	GPIO42INTD	GPIO42OUTCFG		GPIO42INCFG	GPIO41INTD	GPIO41OUTCFG		GPIO41INCFG	GPIO40INTD	GPIO40OUTCFG		GPIO40INCFG

Table 8-92: CFGF Register Bits

Bit	Name	Reset	RW	Description
31	GPIO47INTD	0x0	RW	GPIO47 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO47OUTCFG	0x0	RW	GPIO47 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 8-92: CFGF Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
28	GPIO47INCFG	0x0	RW	GPIO47 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
27	GPIO46INTD	0x0	RW	GPIO46 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO46OUTCFG	0x0	RW	GPIO46 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO46INCFG	0x0	RW	GPIO46 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
23	GPIO45INTD	0x0	RW	GPIO45 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO45OUTCFG	0x0	RW	GPIO45 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO45INCFG	0x0	RW	GPIO45 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
19	GPIO44INTD	0x0	RW	GPIO44 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO44OUTCFG	0x0	RW	GPIO44 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO44INCFG	0x0	RW	GPIO44 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
15	GPIO43INTD	0x0	RW	GPIO43 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO43OUTCFG	0x0	RW	GPIO43 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO43INCFG	0x0	RW	GPIO43 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
11	GPIO42INTD	0x0	RW	GPIO42 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 8-92: CFGF Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
10:9	GPIO42OUTCFG	0x0	RW	GPIO42 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO42INCFG	0x0	RW	GPIO42 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
7	GPIO41INTD	0x0	RW	GPIO41 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO41OUTCFG	0x0	RW	GPIO41 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO41INCFG	0x0	RW	GPIO41 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO40INTD	0x0	RW	GPIO40 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO40OUTCFG	0x0	RW	GPIO40 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO40INCFG	0x0	RW	GPIO40 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.20 CFGG Register

GPIO Configuration Register G
OFFSET: 0x00000058
INSTANCE 0 ADDRESS: 0x40010058
GPIO configuration controls for GPIO[49:48]. Writes to this register must be unlocked by the PADKEY register.

Table 8-93: CFGG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								GPIO49INTD	GPIO49OUTCFG		GPIO49INCFG	GPIO48INTD	GPIO48OUTCFG		GPIO48INCFG

Table 8-94: CFGG Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	GPIO49INTD	0x0	RW	GPIO49 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO49OUTCFG	0x0	RW	GPIO49 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO49INCFG	0x0	RW	GPIO49 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero
3	GPIO48INTD	0x0	RW	GPIO48 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO48OUTCFG	0x0	RW	GPIO48 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO48INCFG	0x0	RW	GPIO48 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Read back will always be zero

8.6.2.21 PADKEY Register

Key Register for all pad configuration registers
OFFSET: 0x00000060
INSTANCE 0 ADDRESS: 0x40010060
Key Register for all pad configuration registers

Table 8-95: PADKEY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PADKEY																															

Table 8-96: PADKEY Register Bits

Bit	Name	Reset	RW	Description
31:0	PADKEY	0x0	RW	Key register value. Key = 0x73 - Key

8.6.2.22 RDA Register

GPIO Input Register A

OFFSET: 0x00000080
INSTANCE 0 ADDRESS: 0x40010080
GPIO Input Register A

Table 8-97: RDA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RDA																															

Table 8-98: RDA Register Bits

Bit	Name	Reset	RW	Description
31:0	RDA	0x0	RO	GPIO31-0 read data.

8.6.2.23 RDB Register

GPIO Input Register B
OFFSET: 0x00000084
INSTANCE 0 ADDRESS: 0x40010084
GPIO Input Register B

Table 8-99: RDB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																RDB															

Table 8-100: RDB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	RDB	0x0	RO	GPIO49-32 read data.

8.6.2.24 WTA Register

GPIO Output Register A
OFFSET: 0x00000088
INSTANCE 0 ADDRESS: 0x40010088
GPIO Output Register A

Table 8-101: WTA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WTA																															

Table 8-102: WTA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTA	0x0	RW	GPIO31-0 write data.

8.6.2.25 WTB Register

GPIO Output Register B
OFFSET: 0x0000008C
INSTANCE 0 ADDRESS: 0x4001008C
GPIO Output Register B

Table 8-103: WTB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																WTB															

Table 8-104: WTB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	WTB	0x0	RW	GPIO49-32 write data.

8.6.2.26 WTSA Register

GPIO Output Register A Set
OFFSET: 0x00000090
INSTANCE 0 ADDRESS: 0x40010090
GPIO Output Register A Set

Table 8-105: WTSA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WTSA																															

Table 8-106: WTSA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTSA	0x0	WO	Set the GPIO31-0 write data.

8.6.2.27 WTSB Register

GPIO Output Register B Set
OFFSET: 0x00000094
INSTANCE 0 ADDRESS: 0x40010094

GPIO Output Register B Set

Table 8-107: WTSB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																WTSB															

Table 8-108: WTSB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	WTSB	0x0	WO	Set the GPIO49-32 write data.

8.6.2.28 WTCA Register

GPIO Output Register A Clear
OFFSET: 0x00000098
INSTANCE 0 ADDRESS: 0x40010098
GPIO Output Register A Clear

Table 8-109: WTCA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WTCA																															

Table 8-110: WTCA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTCA	0x0	WO	Clear the GPIO31-0 write data.

8.6.2.29 WTCB Register

GPIO Output Register B Clear
OFFSET: 0x0000009C
INSTANCE 0 ADDRESS: 0x4001009C
GPIO Output Register B Clear

Table 8-111: WTCB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																WTCB															

Table 8-112: WTCB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	WTCB	0x0	WO	Clear the GPIO49-32 write data.

8.6.2.30 ENA Register

GPIO Enable Register A
OFFSET: 0x000000A0
INSTANCE 0 ADDRESS: 0x400100A0
GPIO Enable Register A

Table 8-113: ENA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ENA																															

Table 8-114: ENA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENA	0x0	RW	GPIO31-0 output enables

8.6.2.31 ENB Register

GPIO Enable Register B
OFFSET: 0x000000A4
INSTANCE 0 ADDRESS: 0x400100A4
GPIO Enable Register B

Table 8-115: ENB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																ENB															

Table 8-116: ENB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	ENB	0x0	RW	GPIO49-32 output enables

8.6.2.32 ENSA Register

GPIO Enable Register A Set
OFFSET: 0x000000A8

INSTANCE 0 ADDRESS: 0x400100A8
GPIO Enable Register A Set

Table 8-117: ENSA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ENSA																															

Table 8-118: ENSA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENSA	0x0	RW	Set the GPIO31-0 output enables

8.6.2.33 ENSB Register

GPIO Enable Register B Set
OFFSET: 0x000000AC
INSTANCE 0 ADDRESS: 0x400100AC
GPIO Enable Register B Set

Table 8-119: ENSB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																ENSB															

Table 8-120: ENSB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	ENSB	0x0	RW	Set the GPIO49-32 output enables

8.6.2.34 ENCA Register

GPIO Enable Register A Clear
OFFSET: 0x000000B4
INSTANCE 0 ADDRESS: 0x400100B4
GPIO Enable Register A Clear

Table 8-121: ENCA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ENCA																															

Table 8-122: ENCA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENCA	0x0	RW	Clear the GPIO31-0 output enables

8.6.2.35 ENCB Register

GPIO Enable Register B Clear
OFFSET: 0x000000B8
INSTANCE 0 ADDRESS: 0x400100B8
GPIO Enable Register B Clear

Table 8-123: ENCB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																ENCB															

Table 8-124: ENCB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	ENCB	0x0	RW	Clear the GPIO49-32 output enables

8.6.2.36 STMRCAP Register

STIMER Capture Control
OFFSET: 0x000000BC
INSTANCE 0 ADDRESS: 0x400100BC
STIMER Capture trigger select and enable.

Table 8-125: STMRCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	STPOL3	STSEL3						RSVD	STPOL2	STSEL2						RSVD	STPOL1	STSEL1						RSVD	STPOL0	STSEL0					

Table 8-126: STMRCAP Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	STPOL3	0x0	RW	STIMER Capture 3 Polarity. CAPLH = 0x0 - Capture on low to high GPIO transition CAPHL = 0x1 - Capture on high to low GPIO transition
29:24	STSEL3	0x3f	RW	STIMER Capture 3 Select.
23	RSVD	0x0	RO	RESERVED

Table 8-126: STMRCAP Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
22	STPOL2	0x0	RW	STIMER Capture 2 Polarity. CAPLH = 0x0 - Capture on low to high GPIO transition CAPHL = 0x1 - Capture on high to low GPIO transition
21:16	STSEL2	0x3f	RW	STIMER Capture 2 Select.
15	RSVD	0x0	RO	RESERVED
14	STPOL1	0x0	RW	STIMER Capture 1 Polarity. CAPLH = 0x0 - Capture on low to high GPIO transition CAPHL = 0x1 - Capture on high to low GPIO transition
13:8	STSEL1	0x3f	RW	STIMER Capture 1 Select.
7	RSVD	0x0	RO	RESERVED
6	STPOL0	0x0	RW	STIMER Capture 0 Polarity. CAPLH = 0x0 - Capture on low to high GPIO transition CAPHL = 0x1 - Capture on high to low GPIO transition
5:0	STSEL0	0x3f	RW	STIMER Capture 0 Select.

8.6.2.37 IOM0IRQ Register

IOM0 Flow Control IRQ Select
OFFSET: 0x000000C0
INSTANCE 0 ADDRESS: 0x400100C0
IOMSTR0 IRQ select for flow control.

Table 8-127: IOM0IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM0IRQ					

Table 8-128: IOM0IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM0IRQ	0x3f	RW	IOMSTR0 IRQ pad select.

8.6.2.38 IOM1IRQ Register

IOM1 Flow Control IRQ Select
OFFSET: 0x000000C4
INSTANCE 0 ADDRESS: 0x400100C4
IOMSTR1 IRQ select for flow control.

Table 8-129: IOM1IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM1IRQ					

Table 8-130: IOM1IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM1IRQ	0x3f	RW	IOMSTR1 IRQ pad select.

8.6.2.39 IOM2IRQ Register

IOM2 Flow Control IRQ Select
OFFSET: 0x000000C8
INSTANCE 0 ADDRESS: 0x400100C8
IOMSTR2 IRQ select for flow control.

Table 8-131: IOM2IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM2IRQ					

Table 8-132: IOM2IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM2IRQ	0x3f	RW	IOMSTR2 IRQ pad select.

8.6.2.40 IOM3IRQ Register

IOM3 Flow Control IRQ Select
OFFSET: 0x000000CC
INSTANCE 0 ADDRESS: 0x400100CC
IOMSTR3 IRQ select for flow control.

Table 8-133: IOM3IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM3IRQ					

Table 8-134: IOM3IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM3IRQ	0x3f	RW	IOMSTR3 IRQ pad select.

8.6.2.41 IOM4IRQ Register

IOM4 Flow Control IRQ Select

OFFSET: 0x000000D0
INSTANCE 0 ADDRESS: 0x400100D0
IOMSTR4 IRQ select for flow control.

Table 8-135: IOM4IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM4IRQ					

Table 8-136: IOM4IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM4IRQ	0x3f	RW	IOMSTR4 IRQ pad select.

8.6.2.42 IOM5IRQ Register

IOM5 Flow Control IRQ Select
OFFSET: 0x000000D4
INSTANCE 0 ADDRESS: 0x400100D4
IOMSTR5 IRQ select for flow control.

Table 8-137: IOM5IRQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										IOM5IRQ					

Table 8-138: IOM5IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM5IRQ	0x3f	RW	IOMSTR5 IRQ pad select.

8.6.2.43 LOOPBACK Register

IOM to IOS Loopback Control
OFFSET: 0x000000D8
INSTANCE 0 ADDRESS: 0x400100D8
IOM to IOS loopback control.

Table 8-139: LOOPBACK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															LOOPBACK

Table 8-140: LOOPBACK Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2:0	LOOPBACK	0x0	RW	IOM to IOS loopback control. LOOP0 = 0x0 - Loop IOM0 to IOS LOOP1 = 0x1 - Loop IOM1 to IOS LOOP2 = 0x2 - Loop IOM2 to IOS LOOP3 = 0x3 - Loop IOM3 to IOS LOOP4 = 0x4 - Loop IOM4 to IOS LOOP5 = 0x5 - Loop IOM5 to IOS LOOPNONE = 0x6 - No loopback connections

8.6.2.44 GPIOOBS Register

GPIO Observation Mode Sample register
OFFSET: 0x000000DC
INSTANCE 0 ADDRESS: 0x400100DC
GPIO Observation mode sample register

Table 8-141: GPIOOBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																OBS_DATA															

Table 8-142: GPIOOBS Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:0	OBS_DATA	0x0	RW	Sample of the data output on the GPIO observation port. May have async sampling issues, as the data is not synchronized to the read operation. Intended for debug purposes only

8.6.2.45 ALTPADCFGA Register

Alternate Pad Configuration reg0 (Pads 3,2,1,0)
OFFSET: 0x000000E0
INSTANCE 0 ADDRESS: 0x400100E0
This register has additional configuration control for pads 3, 2, 1, 0

Table 8-143: ALTPADCFGA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD3_SR	RSVD			PAD3_DS1	RSVD			PAD2_SR	RSVD			PAD2_DS1	RSVD			PAD1_SR	RSVD			PAD1_DS1	RSVD			PAD0_SR	RSVD			PAD0_DS1

Table 8-144: ALTPADCFGA Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD3_SR	0x0	RW	Pad 3 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD3_DS1	0x0	RW	Pad 3 high order drive strength selection. Used in conjunction with PAD3STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD2_SR	0x0	RW	Pad 2 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD2_DS1	0x0	RW	Pad 2 high order drive strength selection. Used in conjunction with PAD2STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD1_SR	0x0	RW	Pad 1 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD1_DS1	0x0	RW	Pad 1 high order drive strength selection. Used in conjunction with PAD1STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD0_SR	0x0	RW	Pad 0 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD0_DS1	0x0	RW	Pad 0 high order drive strength selection. Used in conjunction with PAD0STRNG field to set the pad drive strength.

8.6.2.46 ALTPADCFGB Register

Alternate Pad Configuration reg1 (Pads 7,6,5,4)
OFFSET: 0x000000E4
INSTANCE 0 ADDRESS: 0x400100E4
This register has additional configuration control for pads 7, 6, 5, 4

Table 8-145: ALTPADCFGB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD7_SR	RSVD			PAD7_DS1	RSVD			PAD6_SR	RSVD			PAD6_DS1	RSVD			PAD5_SR	RSVD			PAD5_DS1	RSVD			PAD4_SR	RSVD			PAD4_DS1

Table 8-146: ALTPADCFGB Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD7_SR	0x0	RW	Pad 7 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD7_DS1	0x0	RW	Pad 7 high order drive strength selection. Used in conjunction with PAD7STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD6_SR	0x0	RW	Pad 6 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD6_DS1	0x0	RW	Pad 6 high order drive strength selection. Used in conjunction with PAD6STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD5_SR	0x0	RW	Pad 5 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD5_DS1	0x0	RW	Pad 5 high order drive strength selection. Used in conjunction with PAD5STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD4_SR	0x0	RW	Pad 4 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD4_DS1	0x0	RW	Pad 4 high order drive strength selection. Used in conjunction with PAD4STRNG field to set the pad drive strength.

8.6.2.47 ALTPADCFGC Register

Alternate Pad Configuration reg2 (Pads 11,10,9,8)
OFFSET: 0x000000E8
INSTANCE 0 ADDRESS: 0x400100E8
This register has additional configuration control for pads 11, 10, 9, 8

Table 8-147: ALTPADCFGC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD11_SR	RSVD			PAD11_DS1	RSVD			PAD10_SR	RSVD			PAD10_DS1	RSVD			PAD9_SR	RSVD			PAD9_DS1	RSVD			PAD8_SR	RSVD			PAD8_DS1

Table 8-148: ALTPADCFGC Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD11_SR	0x0	RW	Pad 11 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD11_DS1	0x0	RW	Pad 11 high order drive strength selection. Used in conjunction with PAD11STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD10_SR	0x0	RW	Pad 10 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD10_DS1	0x0	RW	Pad 10 high order drive strength selection. Used in conjunction with PAD10STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD9_SR	0x0	RW	Pad 9 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD9_DS1	0x0	RW	Pad 9 high order drive strength selection. Used in conjunction with PAD9STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD8_SR	0x0	RW	Pad 8 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD8_DS1	0x0	RW	Pad 8 high order drive strength selection. Used in conjunction with PAD8STRNG field to set the pad drive strength.

8.6.2.48 ALTPADCFGD Register

Alternate Pad Configuration reg3 (Pads 15,14,13,12)
OFFSET: 0x000000EC
INSTANCE 0 ADDRESS: 0x400100EC
This register has additional configuration control for pads 15, 14, 13, 12

Table 8-149: ALTPADCFGD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD15_SR	RSVD			PAD15_DS1	RSVD			PAD14_SR	RSVD			PAD14_DS1	RSVD			PAD13_SR	RSVD			PAD13_DS1	RSVD			PAD12_SR	RSVD			PAD12_DS1

Table 8-150: ALTPADCFGD Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD15_SR	0x0	RW	Pad 15 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD15_DS1	0x0	RW	Pad 15 high order drive strength selection. Used in conjunction with PAD15STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD14_SR	0x0	RW	Pad 14 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD14_DS1	0x0	RW	Pad 14 high order drive strength selection. Used in conjunction with PAD14STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD13_SR	0x0	RW	Pad 13 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD13_DS1	0x0	RW	Pad 13 high order drive strength selection. Used in conjunction with PAD13STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD12_SR	0x0	RW	Pad 12 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD12_DS1	0x0	RW	Pad 12 high order drive strength selection. Used in conjunction with PAD12STRNG field to set the pad drive strength.

8.6.2.49 ALTPADCFGE Register

Alternate Pad Configuration reg4 (Pads 19,18,17,16)
OFFSET: 0x000000F0
INSTANCE 0 ADDRESS: 0x400100F0
This register has additional configuration control for pads 19, 18, 17, 16

Table 8-151: ALTPADCFGE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD19_SR	RSVD			PAD19_DS1	RSVD			PAD18_SR	RSVD			PAD18_DS1	RSVD			PAD17_SR	RSVD			PAD17_DS1	RSVD			PAD16_SR	RSVD			PAD16_DS1

Table 8-152: ALTPADCFGE Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD19_SR	0x0	RW	Pad 19 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD19_DS1	0x0	RW	Pad 19 high order drive strength selection. Used in conjunction with PAD19STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD18_SR	0x0	RW	Pad 18 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD18_DS1	0x0	RW	Pad 18 high order drive strength selection. Used in conjunction with PAD18STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD17_SR	0x0	RW	Pad 17 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD17_DS1	0x0	RW	Pad 17 high order drive strength selection. Used in conjunction with PAD17STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD16_SR	0x0	RW	Pad 16 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD16_DS1	0x0	RW	Pad 16 high order drive strength selection. Used in conjunction with PAD16STRNG field to set the pad drive strength.

8.6.2.50 ALTPADCFGF Register

Alternate Pad Configuration reg5 (Pads 23,22,21,20)
OFFSET: 0x000000F4
INSTANCE 0 ADDRESS: 0x400100F4
This register has additional configuration control for pads 23, 22, 21, 20

Table 8-153: ALTPADCFGF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD23_SR	RSVD			PAD23_DS1	RSVD			PAD22_SR	RSVD			PAD22_DS1	RSVD			PAD21_SR	RSVD			PAD21_DS1	RSVD			PAD20_SR	RSVD			PAD20_DS1

Table 8-154: ALTPADCFGF Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD23_SR	0x0	RW	Pad 23 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD23_DS1	0x0	RW	Pad 23 high order drive strength selection. Used in conjunction with PAD23STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD22_SR	0x0	RW	Pad 22 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD22_DS1	0x0	RW	Pad 22 high order drive strength selection. Used in conjunction with PAD22STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD21_SR	0x0	RW	Pad 21 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD21_DS1	0x0	RW	Pad 21 high order drive strength selection. Used in conjunction with PAD21STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD20_SR	0x0	RW	Pad 20 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD20_DS1	0x0	RW	Pad 20 high order drive strength selection. Used in conjunction with PAD20STRNG field to set the pad drive strength.

8.6.2.51 ALTPADCFGG Register

Alternate Pad Configuration reg6 (Pads 27,26,25,24)
OFFSET: 0x000000F8
INSTANCE 0 ADDRESS: 0x400100F8
This register has additional configuration control for pads 27, 26, 25, 24

Table 8-155: ALTPADCFGG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD27_SR	RSVD			PAD27_DS1	RSVD			PAD26_SR	RSVD			PAD26_DS1	RSVD			PAD25_SR	RSVD			PAD25_DS1	RSVD			PAD24_SR	RSVD			PAD24_DS1

Table 8-156: ALTPADCFGG Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD27_SR	0x0	RW	Pad 27 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD27_DS1	0x0	RW	Pad 27 high order drive strength selection. Used in conjunction with PAD27STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD26_SR	0x0	RW	Pad 26 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD26_DS1	0x0	RW	Pad 26 high order drive strength selection. Used in conjunction with PAD26STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD25_SR	0x0	RW	Pad 25 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD25_DS1	0x0	RW	Pad 25 high order drive strength selection. Used in conjunction with PAD25STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD24_SR	0x0	RW	Pad 24 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD24_DS1	0x0	RW	Pad 24 high order drive strength selection. Used in conjunction with PAD24STRNG field to set the pad drive strength.

8.6.2.52 ALTPADCFGH Register

Alternate Pad Configuration reg7 (Pads 31,30,29,28)
OFFSET: 0x000000FC
INSTANCE 0 ADDRESS: 0x400100FC
This register has additional configuration control for pads 31, 30, 29, 28

Table 8-157: ALTPADCFGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD31_SR	RSVD			PAD31_DS1	RSVD			PAD30_SR	RSVD			PAD30_DS1	RSVD			PAD29_SR	RSVD			PAD29_DS1	RSVD			PAD28_SR	RSVD			PAD28_DS1

Table 8-158: ALTPADCFGH Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD31_SR	0x0	RW	Pad 31 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD31_DS1	0x0	RW	Pad 31 high order drive strength selection. Used in conjunction with PAD31STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD30_SR	0x0	RW	Pad 30 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD30_DS1	0x0	RW	Pad 30 high order drive strength selection. Used in conjunction with PAD30STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD29_SR	0x0	RW	Pad 29 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD29_DS1	0x0	RW	Pad 29 high order drive strength selection. Used in conjunction with PAD29STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD28_SR	0x0	RW	Pad 28 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD28_DS1	0x0	RW	Pad 28 high order drive strength selection. Used in conjunction with PAD28STRNG field to set the pad drive strength.

8.6.2.53 ALTPADCFGI Register

Alternate Pad Configuration reg8 (Pads 35,34,33,32)
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x40010100
This register has additional configuration control for pads 35, 34, 33, 32

Table 8-159: ALTPADCFGI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD35_SR	RSVD			PAD35_DS1	RSVD			PAD34_SR	RSVD			PAD34_DS1	RSVD			PAD33_SR	RSVD			PAD33_DS1	RSVD			PAD32_SR	RSVD			PAD32_DS1

Table 8-160: ALTPADCFGI Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD35_SR	0x0	RW	Pad 35 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD35_DS1	0x0	RW	Pad 35 high order drive strength selection. Used in conjunction with PAD35STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD34_SR	0x0	RW	Pad 34 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD34_DS1	0x0	RW	Pad 34 high order drive strength selection. Used in conjunction with PAD34STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD33_SR	0x0	RW	Pad 33 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD33_DS1	0x0	RW	Pad 33 high order drive strength selection. Used in conjunction with PAD33STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD32_SR	0x0	RW	Pad 32 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD32_DS1	0x0	RW	Pad 32 high order drive strength selection. Used in conjunction with PAD32STRNG field to set the pad drive strength.

8.6.2.54 ALTPADCFGJ Register

Alternate Pad Configuration reg9 (Pads 39,38,37,36)
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x40010104
This register has additional configuration control for pads 39, 38, 37, 36

Table 8-161: ALTPADCFGJ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD39_SR	RSVD			PAD39_DS1	RSVD			PAD38_SR	RSVD			PAD38_DS1	RSVD			PAD37_SR	RSVD			PAD37_DS1	RSVD			PAD36_SR	RSVD			PAD36_DS1

Table 8-162: ALTPADCFGJ Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD39_SR	0x0	RW	Pad 39 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD39_DS1	0x0	RW	Pad 39 high order drive strength selection. Used in conjunction with PAD39STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD38_SR	0x0	RW	Pad 38 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD38_DS1	0x0	RW	Pad 38 high order drive strength selection. Used in conjunction with PAD38STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD37_SR	0x0	RW	Pad 37 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD37_DS1	0x0	RW	Pad 37 high order drive strength selection. Used in conjunction with PAD37STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD36_SR	0x0	RW	Pad 36 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD36_DS1	0x0	RW	Pad 36 high order drive strength selection. Used in conjunction with PAD36STRNG field to set the pad drive strength.

8.6.2.55 ALTPADCFGK Register

Alternate Pad Configuration reg10 (Pads 43,42,41,40)
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x40010108
This register has additional configuration control for pads 43, 42, 41, 40

Table 8-163: ALTPADCFGK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD43_SR	RSVD			PAD43_DS1	RSVD			PAD42_SR	RSVD			PAD42_DS1	RSVD			PAD41_SR	RSVD			PAD41_DS1	RSVD			PAD40_SR	RSVD			PAD40_DS1

Table 8-164: ALTPADCFGK Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD43_SR	0x0	RW	Pad 43 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD43_DS1	0x0	RW	Pad 43 high order drive strength selection. Used in conjunction with PAD43STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD42_SR	0x0	RW	Pad 42 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD42_DS1	0x0	RW	Pad 42 high order drive strength selection. Used in conjunction with PAD42STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD41_SR	0x0	RW	Pad 41 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD41_DS1	0x0	RW	Pad 41 high order drive strength selection. Used in conjunction with PAD41STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD40_SR	0x0	RW	Pad 40 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD40_DS1	0x0	RW	Pad 40 high order drive strength selection. Used in conjunction with PAD40STRNG field to set the pad drive strength.

8.6.2.56 ALTPADCFGL Register

Alternate Pad Configuration reg11 (Pads 47,46,45,44)
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x4001010C
This register has additional configuration control for pads 47, 46, 45, 44

Table 8-165: ALTPADCFGL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			PAD47_SR	RSVD			PAD47_DS1	RSVD			PAD46_SR	RSVD			PAD46_DS1	RSVD			PAD45_SR	RSVD			PAD45_DS1	RSVD			PAD44_SR	RSVD			PAD44_DS1

Table 8-166: ALTPADCFGL Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28	PAD47_SR	0x0	RW	Pad 47 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
27:25	RSVD	0x0	RO	RESERVED
24	PAD47_DS1	0x0	RW	Pad 47 high order drive strength selection. Used in conjunction with PAD47STRNG field to set the pad drive strength.
23:21	RSVD	0x0	RO	RESERVED
20	PAD46_SR	0x0	RW	Pad 46 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
19:17	RSVD	0x0	RO	RESERVED
16	PAD46_DS1	0x0	RW	Pad 46 high order drive strength selection. Used in conjunction with PAD46STRNG field to set the pad drive strength.
15:13	RSVD	0x0	RO	RESERVED
12	PAD45_SR	0x0	RW	Pad 45 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD45_DS1	0x0	RW	Pad 45 high order drive strength selection. Used in conjunction with PAD45STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD44_SR	0x0	RW	Pad 44 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD44_DS1	0x0	RW	Pad 44 high order drive strength selection. Used in conjunction with PAD44STRNG field to set the pad drive strength.

8.6.2.57 ALTPADCFGM Register

IAlternate Pad Configuration reg12 (Pads 49,48)
OFFSET: 0x00000110
INSTANCE 0 ADDRESS: 0x40010110
This register has additional configuration control for pads 49, 48

Table 8-167: ALTPADCFGM Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13	12	11 10 09	08	07 06 05	04	03 02 01	00
RSVD	PAD49_SR	RSVD	PAD49_DS1	RSVD	PAD48_SR	RSVD	PAD48_DS1

Table 8-168: ALTPADCFGM Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	PAD49_SR	0x0	RW	Pad 49 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
11:9	RSVD	0x0	RO	RESERVED
8	PAD49_DS1	0x0	RW	Pad 49 high order drive strength selection. Used in conjunction with PAD49STRNG field to set the pad drive strength.
7:5	RSVD	0x0	RO	RESERVED
4	PAD48_SR	0x0	RW	Pad 48 slew rate selection. SR_EN = 0x1 - Enables Slew rate control on pad
3:1	RSVD	0x0	RO	RESERVED
0	PAD48_DS1	0x0	RW	Pad 48 high order drive strength selection. Used in conjunction with PAD48STRNG field to set the pad drive strength.

8.6.2.58 INT0EN Register

GPIO Interrupt Registers 31-0: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x40010200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 8-169: INT0EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

Table 8-170: INT0EN Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31	0x0	RW	GPIO31 interrupt.
30	GPIO30	0x0	RW	GPIO30 interrupt.
29	GPIO29	0x0	RW	GPIO29 interrupt.
28	GPIO28	0x0	RW	GPIO28 interrupt.
27	GPIO27	0x0	RW	GPIO27 interrupt.
26	GPIO26	0x0	RW	GPIO26 interrupt.
25	GPIO25	0x0	RW	GPIO25 interrupt.
24	GPIO24	0x0	RW	GPIO24 interrupt.
23	GPIO23	0x0	RW	GPIO23 interrupt.
22	GPIO22	0x0	RW	GPIO22 interrupt.
21	GPIO21	0x0	RW	GPIO21 interrupt.
20	GPIO20	0x0	RW	GPIO20 interrupt.

Table 8-170: INT0EN Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
19	GPIO19	0x0	RW	GPIO19 interrupt.
18	GPIO18	0x0	RW	GPIO18interrupt.
17	GPIO17	0x0	RW	GPIO17 interrupt.
16	GPIO16	0x0	RW	GPIO16 interrupt.
15	GPIO15	0x0	RW	GPIO15 interrupt.
14	GPIO14	0x0	RW	GPIO14 interrupt.
13	GPIO13	0x0	RW	GPIO13 interrupt.
12	GPIO12	0x0	RW	GPIO12 interrupt.
11	GPIO11	0x0	RW	GPIO11 interrupt.
10	GPIO10	0x0	RW	GPIO10 interrupt.
9	GPIO9	0x0	RW	GPIO9 interrupt.
8	GPIO8	0x0	RW	GPIO8 interrupt.
7	GPIO7	0x0	RW	GPIO7 interrupt.
6	GPIO6	0x0	RW	GPIO6 interrupt.
5	GPIO5	0x0	RW	GPIO5 interrupt.
4	GPIO4	0x0	RW	GPIO4 interrupt.
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

8.6.2.59 INT0STAT Register

GPIO Interrupt Registers 31-0: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x40010204
Read bits from this register to discover the cause of a recent interrupt.

Table 8-171: INT0STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

Table 8-172: INT0STAT Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31	0x0	RW	GPIO31 interrupt.
30	GPIO30	0x0	RW	GPIO30 interrupt.
29	GPIO29	0x0	RW	GPIO29 interrupt.
28	GPIO28	0x0	RW	GPIO28 interrupt.
27	GPIO27	0x0	RW	GPIO27 interrupt.

Table 8-172: INT0STAT Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
26	GPIO26	0x0	RW	GPIO26 interrupt.
25	GPIO25	0x0	RW	GPIO25 interrupt.
24	GPIO24	0x0	RW	GPIO24 interrupt.
23	GPIO23	0x0	RW	GPIO23 interrupt.
22	GPIO22	0x0	RW	GPIO22 interrupt.
21	GPIO21	0x0	RW	GPIO21 interrupt.
20	GPIO20	0x0	RW	GPIO20 interrupt.
19	GPIO19	0x0	RW	GPIO19 interrupt.
18	GPIO18	0x0	RW	GPIO18interrupt.
17	GPIO17	0x0	RW	GPIO17 interrupt.
16	GPIO16	0x0	RW	GPIO16 interrupt.
15	GPIO15	0x0	RW	GPIO15 interrupt.
14	GPIO14	0x0	RW	GPIO14 interrupt.
13	GPIO13	0x0	RW	GPIO13 interrupt.
12	GPIO12	0x0	RW	GPIO12 interrupt.
11	GPIO11	0x0	RW	GPIO11 interrupt.
10	GPIO10	0x0	RW	GPIO10 interrupt.
9	GPIO9	0x0	RW	GPIO9 interrupt.
8	GPIO8	0x0	RW	GPIO8 interrupt.
7	GPIO7	0x0	RW	GPIO7 interrupt.
6	GPIO6	0x0	RW	GPIO6 interrupt.
5	GPIO5	0x0	RW	GPIO5 interrupt.
4	GPIO4	0x0	RW	GPIO4 interrupt.
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

8.6.2.60 INT0CLR Register

GPIO Interrupt Registers 31-0: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x40010208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 8-173: INT0CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

Table 8-174: INT0CLR Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31	0x0	RW	GPIO31 interrupt.
30	GPIO30	0x0	RW	GPIO30 interrupt.
29	GPIO29	0x0	RW	GPIO29 interrupt.
28	GPIO28	0x0	RW	GPIO28 interrupt.
27	GPIO27	0x0	RW	GPIO27 interrupt.
26	GPIO26	0x0	RW	GPIO26 interrupt.
25	GPIO25	0x0	RW	GPIO25 interrupt.
24	GPIO24	0x0	RW	GPIO24 interrupt.
23	GPIO23	0x0	RW	GPIO23 interrupt.
22	GPIO22	0x0	RW	GPIO22 interrupt.
21	GPIO21	0x0	RW	GPIO21 interrupt.
20	GPIO20	0x0	RW	GPIO20 interrupt.
19	GPIO19	0x0	RW	GPIO19 interrupt.
18	GPIO18	0x0	RW	GPIO18interrupt.
17	GPIO17	0x0	RW	GPIO17 interrupt.
16	GPIO16	0x0	RW	GPIO16 interrupt.
15	GPIO15	0x0	RW	GPIO15 interrupt.
14	GPIO14	0x0	RW	GPIO14 interrupt.
13	GPIO13	0x0	RW	GPIO13 interrupt.
12	GPIO12	0x0	RW	GPIO12 interrupt.
11	GPIO11	0x0	RW	GPIO11 interrupt.
10	GPIO10	0x0	RW	GPIO10 interrupt.
9	GPIO9	0x0	RW	GPIO9 interrupt.
8	GPIO8	0x0	RW	GPIO8 interrupt.
7	GPIO7	0x0	RW	GPIO7 interrupt.
6	GPIO6	0x0	RW	GPIO6 interrupt.
5	GPIO5	0x0	RW	GPIO5 interrupt.
4	GPIO4	0x0	RW	GPIO4 interrupt.
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

8.6.2.61 INT0SET Register

GPIO Interrupt Registers 31-0: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x4001020C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 8-175: INT0SET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

Table 8-176: INT0SET Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31	0x0	RW	GPIO31 interrupt.
30	GPIO30	0x0	RW	GPIO30 interrupt.
29	GPIO29	0x0	RW	GPIO29 interrupt.
28	GPIO28	0x0	RW	GPIO28 interrupt.
27	GPIO27	0x0	RW	GPIO27 interrupt.
26	GPIO26	0x0	RW	GPIO26 interrupt.
25	GPIO25	0x0	RW	GPIO25 interrupt.
24	GPIO24	0x0	RW	GPIO24 interrupt.
23	GPIO23	0x0	RW	GPIO23 interrupt.
22	GPIO22	0x0	RW	GPIO22 interrupt.
21	GPIO21	0x0	RW	GPIO21 interrupt.
20	GPIO20	0x0	RW	GPIO20 interrupt.
19	GPIO19	0x0	RW	GPIO19 interrupt.
18	GPIO18	0x0	RW	GPIO18interrupt.
17	GPIO17	0x0	RW	GPIO17 interrupt.
16	GPIO16	0x0	RW	GPIO16 interrupt.
15	GPIO15	0x0	RW	GPIO15 interrupt.
14	GPIO14	0x0	RW	GPIO14 interrupt.
13	GPIO13	0x0	RW	GPIO13 interrupt.
12	GPIO12	0x0	RW	GPIO12 interrupt.
11	GPIO11	0x0	RW	GPIO11 interrupt.
10	GPIO10	0x0	RW	GPIO10 interrupt.
9	GPIO9	0x0	RW	GPIO9 interrupt.
8	GPIO8	0x0	RW	GPIO8 interrupt.
7	GPIO7	0x0	RW	GPIO7 interrupt.
6	GPIO6	0x0	RW	GPIO6 interrupt.
5	GPIO5	0x0	RW	GPIO5 interrupt.
4	GPIO4	0x0	RW	GPIO4 interrupt.
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

8.6.2.62 INT1EN Register

GPIO Interrupt Registers 49-32: Enable
OFFSET: 0x00000210
INSTANCE 0 ADDRESS: 0x40010210
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 8-177: INT1EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 8-178: INT1EN Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	GPIO49	0x0	RW	GPIO49 interrupt.
16	GPIO48	0x0	RW	GPIO48 interrupt.
15	GPIO47	0x0	RW	GPIO47 interrupt.
14	GPIO46	0x0	RW	GPIO46 interrupt.
13	GPIO45	0x0	RW	GPIO45 interrupt.
12	GPIO44	0x0	RW	GPIO44 interrupt.
11	GPIO43	0x0	RW	GPIO43 interrupt.
10	GPIO42	0x0	RW	GPIO42 interrupt.
9	GPIO41	0x0	RW	GPIO41 interrupt.
8	GPIO40	0x0	RW	GPIO40 interrupt.
7	GPIO39	0x0	RW	GPIO39 interrupt.
6	GPIO38	0x0	RW	GPIO38 interrupt.
5	GPIO37	0x0	RW	GPIO37 interrupt.
4	GPIO36	0x0	RW	GPIO36 interrupt.
3	GPIO35	0x0	RW	GPIO35 interrupt.
2	GPIO34	0x0	RW	GPIO34 interrupt.
1	GPIO33	0x0	RW	GPIO33 interrupt.
0	GPIO32	0x0	RW	GPIO32 interrupt.

8.6.2.63 INT1STAT Register

GPIO Interrupt Registers 49-32: Status
OFFSET: 0x00000214
INSTANCE 0 ADDRESS: 0x40010214
Read bits from this register to discover the cause of a recent interrupt.

Table 8-179: INT1STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 8-180: INT1STAT Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	GPIO49	0x0	RW	GPIO49 interrupt.
16	GPIO48	0x0	RW	GPIO48 interrupt.
15	GPIO47	0x0	RW	GPIO47 interrupt.
14	GPIO46	0x0	RW	GPIO46 interrupt.
13	GPIO45	0x0	RW	GPIO45 interrupt.
12	GPIO44	0x0	RW	GPIO44 interrupt.
11	GPIO43	0x0	RW	GPIO43 interrupt.
10	GPIO42	0x0	RW	GPIO42 interrupt.
9	GPIO41	0x0	RW	GPIO41 interrupt.
8	GPIO40	0x0	RW	GPIO40 interrupt.
7	GPIO39	0x0	RW	GPIO39 interrupt.
6	GPIO38	0x0	RW	GPIO38 interrupt.
5	GPIO37	0x0	RW	GPIO37 interrupt.
4	GPIO36	0x0	RW	GPIO36 interrupt.
3	GPIO35	0x0	RW	GPIO35 interrupt.
2	GPIO34	0x0	RW	GPIO34 interrupt.
1	GPIO33	0x0	RW	GPIO33 interrupt.
0	GPIO32	0x0	RW	GPIO32 interrupt.

8.6.2.64 INT1CLR Register

GPIO Interrupt Registers 49-32: Clear
OFFSET: 0x00000218
INSTANCE 0 ADDRESS: 0x40010218
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 8-181: INT1CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 8-182: INT1CLR Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	GPIO49	0x0	RW	GPIO49 interrupt.
16	GPIO48	0x0	RW	GPIO48 interrupt.
15	GPIO47	0x0	RW	GPIO47 interrupt.
14	GPIO46	0x0	RW	GPIO46 interrupt.
13	GPIO45	0x0	RW	GPIO45 interrupt.
12	GPIO44	0x0	RW	GPIO44 interrupt.
11	GPIO43	0x0	RW	GPIO43 interrupt.
10	GPIO42	0x0	RW	GPIO42 interrupt.
9	GPIO41	0x0	RW	GPIO41 interrupt.
8	GPIO40	0x0	RW	GPIO40 interrupt.
7	GPIO39	0x0	RW	GPIO39 interrupt.
6	GPIO38	0x0	RW	GPIO38 interrupt.
5	GPIO37	0x0	RW	GPIO37 interrupt.
4	GPIO36	0x0	RW	GPIO36 interrupt.
3	GPIO35	0x0	RW	GPIO35 interrupt.
2	GPIO34	0x0	RW	GPIO34 interrupt.
1	GPIO33	0x0	RW	GPIO33 interrupt.
0	GPIO32	0x0	RW	GPIO32 interrupt.

8.6.2.65 INT1SET Register

GPIO Interrupt Registers 49-32: Set
OFFSET: 0x0000021C
INSTANCE 0 ADDRESS: 0x4001021C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 8-183: INT1SET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 8-184: INT1SET Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	GPIO49	0x0	RW	GPIO49 interrupt.
16	GPIO48	0x0	RW	GPIO48 interrupt.
15	GPIO47	0x0	RW	GPIO47 interrupt.

Table 8-184: INT1SET Register Bits *(Continued)*

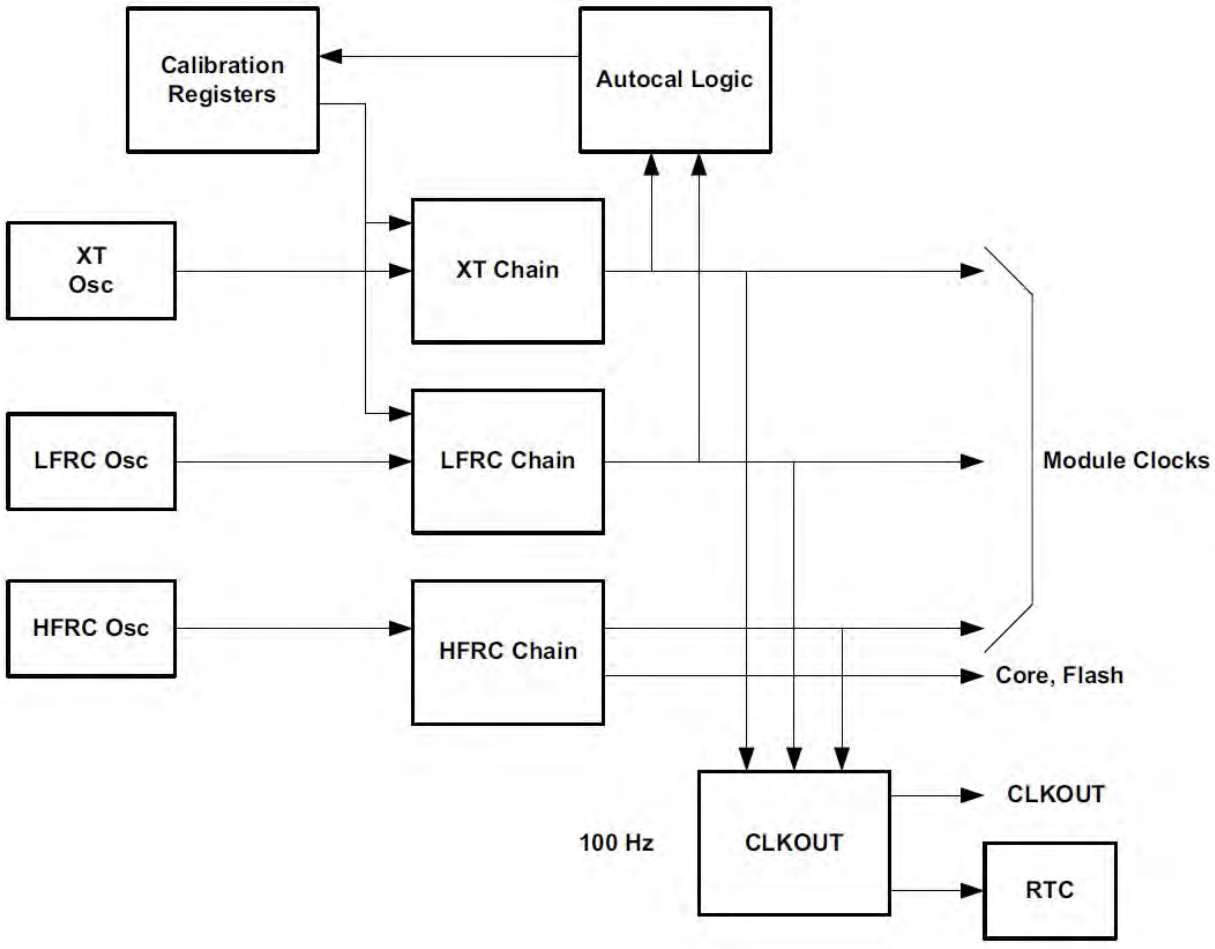
Bit	Name	Reset	RW	Description
14	GPIO46	0x0	RW	GPIO46 interrupt.
13	GPIO45	0x0	RW	GPIO45 interrupt.
12	GPIO44	0x0	RW	GPIO44 interrupt.
11	GPIO43	0x0	RW	GPIO43 interrupt.
10	GPIO42	0x0	RW	GPIO42 interrupt.
9	GPIO41	0x0	RW	GPIO41 interrupt.
8	GPIO40	0x0	RW	GPIO40 interrupt.
7	GPIO39	0x0	RW	GPIO39 interrupt.
6	GPIO38	0x0	RW	GPIO38 interrupt.
5	GPIO37	0x0	RW	GPIO37 interrupt.
4	GPIO36	0x0	RW	GPIO36 interrupt.
3	GPIO35	0x0	RW	GPIO35 interrupt.
2	GPIO34	0x0	RW	GPIO34 interrupt.
1	GPIO33	0x0	RW	GPIO33 interrupt.
0	GPIO32	0x0	RW	GPIO32 interrupt.

SECTION

9

Clock Generation and Real-Time Clock Module

Figure 9-1: Block Diagram for the Clock Generator and Real Time Clock Module



9.1 Clock Generator

9.1.1 Functional Overview

A high-level view of the Clock Generator Module, which supplies all clocks required by the Apollo2 SoC, is shown in Figure 9-1 on page 326. These clocks are derived from one of three fundamental clock sources: a high precision crystal controlled oscillator (XT), a low power 1 KHz RC oscillator (LFRC) and a high frequency 48 MHz oscillator (HFRC).

A clock, CLKOUT, generated from any of the oscillators, may be configured and driven onto an external pin. Each oscillator may be replaced with an external clock input for test or measurement purposes (See *Section 8 GPIO and Pad Configuration Module on page 218*). CLKOUT also drives the Real Time Clock (RTC) Module and other internal clock nodes.

The Clock Generator automatically controls the enabling of the oscillators, so that they are only powered up and used when requested by another module. This allows minimal power consumption without complex software intervention, so that software does not need to manage any enabling or disabling of the oscillators. As an example, an I^2C/SPI Master requires the HFRC in order to generate the serial interface clock. If a transfer is initiated and the processor is put into Deep Sleep mode, the HFRC will remain active until the I/O transfer is completed. At that point the HFRC will be powered down without requiring any software intervention.

9.1.2 Apollo2 SoC Low Frequency RC Oscillator (LFRC)

The low power LFRC, with a nominal frequency of 1024 Hz, is used when short term frequency accuracy is not important. It also supplies clocks for some basic state machines and is always enabled. Calibration logic is included.

9.1.2.1 LFRC Oscillator Digital Calibration

The LFRC Oscillator includes a patented Distributed Digital Calibration function similar to that of the XT Oscillator (*Section 9.1.3.1 XT Oscillator Digital Calibration on page 329*). Because the LFRC Oscillator has a greater fundamental variability, the required range of calibration is much larger. When the 1024 Hz RC oscillator is selected, the clock at the 512 Hz level of the divider chain is modified on a selectable interval using the calibration value CALRC in the REG_CLKGEN_CALRC Register. Clock pulses are either added or subtracted to ensure accuracy of the LFRC. CALRC cycles of the 512 Hz clock are gated (negative calibration) or replaced by 1024 Hz pulses (positive calibration) within every 1024 second calibration period. Each step in CALRC modifies the clock frequency by 1.907 ppm, with a maximum adjustment of +249,954/-249,955 ppm (±25%).

The pulses which are added to or subtracted from the 512 Hz clock are spread evenly over each 1024 second period using the Ambiq patented Distributed Calibration algorithm. This ensures that in LFRC mode the maximum cycle-to-cycle jitter in any clock of a frequency 512 Hz or lower caused by calibration will be no more than one 512 Hz period (~2 ms). This maximum jitter applies to all clocks in the Apollo2 SoC which use the LFRC.

NOTE: Since the 512 Hz LFRC clock is calibrated, the original 1024 Hz LFRC is an uncalibrated clock. This may be a useful selection in some cases.

9.1.2.2 LFRC Calibration Process

The LFRC oscillator calibration value is determined by the following process:

1. Write "0x47" to the CLKKEY register to enable access to CLK_GEN registers.
2. Set the CALRC field to 0 to insure calibration is not occurring.
3. Select the LFRC oscillator by setting the REG_CLKGEN_OCTRL_OSEL bit to 1.
4. Select the LFRC or a division of it on a CLKOUT pad.
5. Measure the frequency Fmeas at the CLKOUT pad.
6. Compute the adjustment value required in ppm as ((Fnom – Fmeas)*1000000)/ Fmeas = PAdj.
7. Compute the adjustment value in steps as PAdj/(1000000/2^19) = PAdj/ (1.90735) = Adj.
8. Compare Adj value with min/max range of -249,955 to 249,954.
9. If target Adj is within min and max, set CALRC = Adj.
10. Otherwise, the LFRC frequency is too low or too high to be calibrated.

9.1.3 High Precision XT Oscillator (XT)

The high accuracy XT Oscillator is tuned to an external 32.768 KHz crystal, and has a nominal frequency of 32.768 KHz. It is used when frequency accuracy is critically important. Because a crystal oscillator uses a significant amount of power, the XT is only enabled when an internal module is using it. Digital calibration logic is included. The output of the XT oscillator may be digitally calibrated to ±1 ppm (part per million).

It should be noted that the XT oscillator is also optional if the requirements of the design can tolerate the internal LFRC/HFRC oscillator specifications. It should also be noted that external capacitors are not required to tune an internal divided clock of the crystal input to achieve a precise scaling of 32.768 KHz. This is handled within the Apollo2 SoC.

9.1.3.1 XT Oscillator Digital Calibration

The XT Oscillator includes a Distributed Digital Calibration function. When the 32 KHz XT oscillator is selected, the clock at the 16 KHz level of the divider chain is modified on a selectable interval using the calibration value CALXT in the REG_-CLKGEN_CALXT Register. Clock pulses are either added or subtracted to ensure accuracy of the XT. CALXT cycles of the 16 KHz clock are gated (negative calibration) or replaced by 32 KHz pulses (positive calibration) within every 64 second calibration period. Each step in CALXT modifies the clock frequency by 0.9535 ppm, with a maximum adjustment of +975/-976 ppm (±0.1%).

The pulses which are added to or subtracted from the 16 KHz clock are spread evenly over each 64 second period using the Ambiq Micro patented Distributed Calibration algorithm. This insures that in XT mode the maximum cycle-to-cycle jitter in any clock of a frequency 16 KHz or lower caused by calibration will be no more than one 16 KHz period (~60 us). This maximum jitter applies to all clocks in the Apollo2 SoC which use the XT.

NOTE: Since the 16 KHz XT clock is calibrated, the 32 KHz XT is an uncalibrated clock. This may be a useful selection in some cases.

9.1.3.2 XT Calibration Process

The XT Oscillator calibration value is determined by the following process:

1. Write "0x47" to the CLKKEY register to enable access to CLK_GEN registers
2. Set the CALXT register field to 0 to insure calibration is not occurring.
3. Select the XT oscillator by setting the REG_CLKGEN_OCTRL_OSEL bit to 0.
4. Select the XT or a division of it on a CLKOUT pad.
5. Measure the frequency Fmeas at the CLKOUT pad.
6. Compute the adjustment value required in ppm as ((Fnom – Fmeas)*1000000)/ Fmeas = PAdj.
7. Compute the adjustment value in steps as PAdj/(1000000/219) = PAdj/(0.9535) = Adj.
8. Compare Adj value with min/max range of -976 to 975.
9. If target Adj is within min and max, set CALXT = Adj.
10. Otherwise, the XT frequency is too low to be calibrated.

9.1.3.3 XT Oscillator Failure Detection

If the 32 KHz XT Oscillator generates clocks at less than 8 KHz for a period of more than 32 ms, the Apollo2 SoC detects an Oscillator Failure. The Oscillator Fail (OF)

flag is set when an Oscillator Failure occurs, and is also set when the Apollo2 SoC initially powers up. If the Oscillator Fail interrupt enable (OFIE) bit is set, the OF flag will generate an interrupt. The current status of the XT Oscillator can be read in the REG_CLKGEN_STATUS_OSCF bit, which will be a 1 if the XT Oscillator is not running at least 8 KHz. Note that OSCF will always be set if the LFRC Oscillator is currently selected by the REG_CLKGEN_OCTRL_OSEL bit.

If the FOS bit in REG_CLK_GEN_OCTRL is set and the Apollo2 SoC RTC is currently using the XT Oscillator, it will automatically switch to the LFRC Oscillator on an Oscillator Failure. This guarantees that the RTC clock will not stop in any case. If the XT Oscillator experiences a temporary failure and subsequently restarts, the Apollo2 SoC will switch back to the XT Oscillator. The REG_CLKGEN_STATUS_OMODE bit indicates the currently selected oscillator, which may not match the oscillator requested by the REG_CLKGEN_OCTRL_OSEL bit if the XT Oscillator is not running.

9.1.4 High Frequency RC Oscillator (HFRC)

The high frequency HFRC Oscillator, with a nominal frequency of 48 MHz, is used to supply all high frequency clocks in the Apollo2 SoC such as the processor clock for the Arm core, memories and many peripheral modules. Digital calibration is not supported for the HFRC, but its frequency may be automatically adjusted by the Auto-adjustment function which is a combination of analog and digital operations.

The HFRC is enabled only when it is required by an internal module. When the Arm core goes into a sleep mode, the HFRC will be disabled unless another module is using it. If the Arm core goes into deep sleep mode, the HFRC will be powered down when it is not needed. When the HFRC is powered up, it will take a few microseconds for it to begin oscillating, and a few more microseconds before the output is completely stable. In order to prevent erroneous internal clocks from occurring, the internal clocks are gated until the HFRC is stable.

9.1.5 HFRC Auto-adjustment

In some applications it is important that the HFRC frequency be more accurate than the ±2% variation typically seen, particularly in cases where the temperature may vary widely. A good example of this is in cases where the Apollo2 SoC communicates with another device via the UART. The frequency matching with the other device in the connection is an important factor in the reliability of the connection. In order to support a highly accurate HFRC, a function called Auto-adjustment is provided.

It should be noted that Auto-adjustment is dependent on an accurate clock source such as the crystal. The min/max variation of the HFRC frequency with and without adjustment is different. See *Section 18.5 Clocks/Oscillators on page 477*.

During auto-adjustment, the number of HFRC cycles which occur in one 32.768 KHz XT Oscillator cycle is compared to a target value. If the count is different from

the target, an HFRC tuning value is modified to change the HFRC frequency. The target count is held in the REG_CLKGEN_HFADJ_HFXTADJ field. If the target HFRC frequency is 48 MHz, the optimal HFXTADJ value is 48,000/32.768 or 1464. A different value will result in a different nominal HFRC frequency.

Auto-adjustment works by periodically enabling the HFRC and the XT, counting the HFRC cycles in a single XT cycle, subtracting that value from HFXTADJ and adding the resulting difference to the actual HFRC tuning value. The current tuning value may be read back in the HFTUNERB field of the REG_CLKGEN_HFVAL Register. Auto-adjustment is enabled in the REG_CLKGEN_HFADJ Register by loading the repeat frequency value into the HFADJCK field and then setting the HFADJEN bit.

Auto-adjustment cycles will occur continuously if both the XT and the HFRC are currently requested by other modules. If either oscillator is disabled, Auto-adjustment cycles will then occur at intervals determined by the REG_CLKGEN_HFADJ_HFADJCK field, as shown in the register description. Shorter repeat intervals will result in more accurate HFRC frequencies, especially if the temperature is changing rapidly, but will result in higher power consumption. When an Auto-adjustment cycle occurs, if the XT was disabled it is enabled and then a delay occurs to allow the XT to stabilize. This delay is defined by the REG_CLKGEN_HFADJ_HFWARMUP field as defined in the Register document. Once the HFRC is stable, the HFRC is enabled and several Auto-adjustments occur, each of which results in a refinement of the tuning value. Once those adjustments are complete, the HFRC and XT are powered down unless they are in use by other functions.

9.1.6 Frequency Measurement

The Autocalibration logic may be used to measure the frequency of an internal clock signal relative to the XT Oscillator frequency. The following steps are required to perform this measurement:

1. Write "0x47" to the CLKKEY register to enable access to CLK_GEN registers
2. Set the REG_CLKGEN_OCTRL_ACAL field to 000.
3. Clear the ACC interrupt flag.
4. Select the clock to be measured with the CKSEL REG_CLKGEN_CLKOUT_CKSEL field.
5. Set ACAL to 110.
6. Wait for the ACC interrupt flag to be set.
7. Read the REG_CLKGEN_ACALCTR_ACALCTR field. This will contain the number of reference clocks which occurred during one cycle of the XT Oscillator.
8. Calculate the frequency of the measured clock.

The measured frequency is:

$F_{MEAS} = F_{REF} \div ACALCTR$

where F_{REF} is the frequency of the reference clock and ACALCTR is the value read from ACALCTR when the measurement is complete. Note that the longer the measurement period is, the more time the measurement takes, but the resulting F_{MEAS} will be more accurate.

9.1.7 Generating 100 Hz

The Real Time Clock (RTC) module requires a 100 Hz clock which is provided by the Clock Generator. This clock may come either from the LFRC or the XT Oscillators, as determined by the REG_CLKGEN_OCTRL_OSEL bit. Since 100 Hz is not a simple power of two division of either of these oscillators, special functions are used to create it.

If the XT Oscillator is selected, 100 Hz is generated by dividing the 2048 Hz division of the XT by 21 for 12 iterations and by 20 for 13 iterations out of every 25 clock periods. This produces an effective division of **(21 * 12 + 20 * 13)/25 = 20.48** producing an exact average frequency of 100 Hz with a maximum jitter of less than 1 ms.

If the LFRC Oscillator is selected, 100 Hz is generated by dividing the 256 Hz division of the LFRC by 2 for 11 iterations and by 3 for 14 iterations out of every 25 clock periods. This produces an effective division of **(2 * 11 + 3 * 14)/25 = 2.56** producing an exact average frequency of 100 Hz with a maximum jitter of less than 8 ms.

9.2 CLKGEN Registers

Clock Generator
INSTANCE 0 BASE ADDRESS:0x40004000

9.2.1 Register Memory Map

Table 9-1: CLKGEN Register Map

Address(es)	Registered Name	Description
0x40004000	CALXT	XT Oscillator Control
0x40004004	CALRC	RC Oscillator Control
0x40004008	ACALCTR	Autocalibration Counter
0x4000400C	OCTRL	Oscillator Control
0x40004010	CLKOUT	CLKOUT Frequency Select
0x40004018	CCTRL	HFRC Clock Control

Table 9-1: CLKGEN Register Map *(Continued)*

Address(es)	Registered Name	Description
0x4000401C	STATUS	Clock Generator Status
0x40004020	HFADJ	HFRC Adjustment
0x40004024	HFVAL	HFADJ readback
0x40004028	CLOCKEN	Clock Enable Status
0x4000402C	CLOCKEN2	Clock Enable Status
0x40004030	CLOCKEN3	Clock Enable Status
0x40004034	UARTEN	UART Enable
0x40004100	INTEN	CLKGEN Interrupt Register: Enable
0x40004104	INTSTAT	CLKGEN Interrupt Register: Status
0x40004108	INTCLR	CLKGEN Interrupt Register: Clear
0x4000410C	INTSET	CLKGEN Interrupt Register: Set

9.2.2 CLKGEN Registers

9.2.2.1 CALXT Register

XT Oscillator Control
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40004000
XT Oscillator Control

Table 9-2: CALXT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					CALXT										

Table 9-3: CALXT Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:0	CALXT	0x0	RW	XT Oscillator calibration value

9.2.2.2 CALRC Register

RC Oscillator Control
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40004004
RC Oscillator Control

Table 9-4: CALRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CALRC															

Table 9-5: CALRC Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	CALRC	0x0	RW	LFRC Oscillator calibration value

9.2.2.3 ACALCTR Register

Autocalibration Counter
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40004008
Autocalibration Counter

Table 9-6: ACALCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD								ACALCTR																							

Table 9-7: ACALCTR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:0	ACALCTR	0x0	RO	Autocalibration Counter result.

9.2.2.4 OCTRL Register

Oscillator Control
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4000400C
Oscillator Control

Table 9-8: OCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					ACAL		OSEL	FOS	RSVD					STOPRC	STOPXT

Table 9-9: OCTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:8	ACAL	0x0	RW	Autocalibration control DIS = 0x0 - Disable Autocalibration 1024SEC = 0x2 - Autocalibrate every 1024 seconds 512SEC = 0x3 - Autocalibrate every 512 seconds XTFREQ = 0x6 - Frequency measurement using XT
7	OSEL	0x0	RW	Selects the RTC oscillator (1 => LFRC, 0 => XT) RTC_XT = 0x0 - RTC uses the XT RTC_LFRC = 0x1 - RTC uses the LFRC
6	FOS	0x0	RW	Oscillator switch on failure function DIS = 0x0 - Disable the oscillator switch on failure function EN = 0x1 - Enable the oscillator switch on failure function
5:2	RSVD	0x0	RO	RESERVED
1	STOPRC	0x0	RW	Stop the LFRC Oscillator to the RTC EN = 0x0 - Enable the LFRC Oscillator to drive the RTC STOP = 0x1 - Stop the LFRC Oscillator when driving the RTC
0	STOPXT	0x0	RW	Stop the XT Oscillator to the RTC EN = 0x0 - Enable the XT Oscillator to drive the RTC STOP = 0x1 - Stop the XT Oscillator when driving the RTC

9.2.2.5 CLKOUT Register

CLKOUT Frequency Select
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40004010
CLKOUT Frequency Select

Table 9-10: CLKOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								CKEN	RSVD	CKSEL					

Table 9-11: CLKOUT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	CKEN	0x0	RW	Enable the CLKOUT signal DIS = 0x0 - Disable CLKOUT EN = 0x1 - Enable CLKOUT

Table 9-11: CLKOUT Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
6	RSVD	0x0	RO	RESERVED
5:0	CKSEL	0x0	RW	CLKOUT signal select. Note that HIGH_DRIVE should be selected if any high frequencies (such as from HFRC) are selected for CLKOUT. LFRC = 0x0 - LFRC XT_DIV2 = 0x1 - XT / 2 XT_DIV4 = 0x2 - XT / 4 XT_DIV8 = 0x3 - XT / 8 XT_DIV16 = 0x4 - XT / 16 XT_DIV32 = 0x5 - XT / 32 RTC_1Hz = 0x10 - 1 Hz as selected in RTC XT_DIV2M = 0x16 - XT / 2^21 XT = 0x17 - XT CG_100Hz = 0x18 - 100 Hz as selected in CLKGEN RSV = 0x19-0x22 - Reserved for future use LFRC_DIV2 = 0x23 - LFRC / 2 LFRC_DIV32 = 0x24 - LFRC / 32 LFRC_DIV512 = 0x25 - LFRC / 512 LFRC_DIV32K = 0x26 - LFRC / 32768 XT_DIV256 = 0x27 - XT / 256 XT_DIV8K = 0x28 - XT / 8192 XT_DIV64K = 0x29 - XT / 2^16 ULFRC_DIV16 = 0x2A - Uncal LFRC / 16 ULFRC_DIV128 = 0x2B - Uncal LFRC / 128 ULFRC_1Hz = 0x2C - Uncal LFRC / 1024 ULFRC_DIV4K = 0x2D - Uncal LFRC / 4096 ULFRC_DIV1M = 0x2E - Uncal LFRC / 2^20 RSV - 0x2F-0x30 - Reserved for future use LFRC_DIV2M = 0x31 - LFRC / 2^20 RSV - 0x32-0x33 - Reserved for future use XTNE = 0x35 - XT (not auto enabled) XTNE_DIV16 = 0x36 - XT / 16 (not auto enabled) LFRCNE_DIV32 = 0x37 - LFRC / 32 (not auto enabled) LFRCNE = 0x39 - LFRC (not auto enabled) - Default for undefined values

9.2.2.6 CCTRL Register

HFRC Clock Control
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x40004018
HFRC Clock Control

Table 9-12: CCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															CORESEL

Table 9-13: CCTRL Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	CORESEL	0x1	RW	Core Clock divisor HFRC = 0x0 - Core Clock is HFRC HFRC_DIV2 = 0x1 - Core Clock is HFRC / 2 Note: HFRC_DIV2 is not supported. CORESEL must be set to HFRC (0x0) during device initialization for proper operation of SoC and modules.

9.2.2.7 STATUS Register

Clock Generator Status
OFFSET: 0x0000001C
INSTANCE 0 ADDRESS: 0x4000401C
Clock Generator Status

Table 9-14: STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														OSCF	OMODE

Table 9-15: STATUS Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	OSCF	0x0	RO	XT Oscillator is enabled but not oscillating
0	OMODE	0x0	RO	Current RTC oscillator (1 => LFRC, 0 => XT)

9.2.2.8 HFADJ Register

HFRC Adjustment
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x40004020
HFRC Adjustment

Table 9-16: HFADJ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD									HFADJ_GAIN		HFWARMUP	HFXTADJ												RSVD					HFADJCK	HFADJEN

Table 9-17: HFADJ Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:21	HFADJ_GAIN	0x1	RW	Gain control for HFRC adjustment Gain_of_1 = 0x0 - HF Adjust with Gain of 1 Gain_of_1_in_2 = 0x1 - HF Adjust with Gain of 0.5 Gain_of_1_in_4 = 0x2 - HF Adjust with Gain of 0.25 Gain_of_1_in_8 = 0x3 - HF Adjust with Gain of 0.125 Gain_of_1_in_16 = 0x4 - HF Adjust with Gain of 0.0625 Gain_of_1_in_32 = 0x5 - HF Adjust with Gain of 0.03125
20	HFWARMUP	0x0	RW	XT warm up period for HFRC adjustment 1SEC = 0x0 - Auto adjust XT warm up period = 1-2 seconds 2SEC = 0x1 - Auto adjust XT warm up period = 2-4 seconds
19:8	HFXTADJ	0x5b8	RW	Target HFRC adjustment value.
7:4	RSVD	0x0	RO	RESERVED
3:1	HFADJCK	0x0	RW	Repeat period for HFRC adjustment 4SEC = 0x0 - Auto adjust repeat period = 4 seconds 16SEC = 0x1 - Auto adjust repeat period = 16 seconds 32SEC = 0x2 - Auto adjust repeat period = 32 seconds 64SEC = 0x3 - Auto adjust repeat period = 64 seconds 128SEC = 0x4 - Auto adjust repeat period = 128 seconds 256SEC = 0x5 - Auto adjust repeat period = 256 seconds 512SEC = 0x6 - Auto adjust repeat period = 512 seconds 1024SEC = 0x7 - Auto adjust repeat period = 1024 seconds
0	HFADJEN	0x0	RW	HFRC adjustment control DIS = 0x0 - Disable the HFRC adjustment EN = 0x1 - Enable the HFRC adjustment

9.2.2.9 HFVAL Register

HFADJ readback
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x40004024
HFADJ readback

Table 9-18: HFVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					HFTUNERB									

Table 9-19: HFVAL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:0	HFTUNERB	0x0	RO	Current HFTUNE value

9.2.2.10 CLOCKEN Register

Clock Enable Status
OFFSET: 0x00000028
INSTANCE 0 ADDRESS: 0x40004028
Clock Enable Status

Table 9-20: CLOCKEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CLOCKEN																															

Table 9-21: CLOCKEN Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCKEN	0x0	RO	Clock enable status ADC_CLKEN = 0x1 - Clock enable for the ADC. CTIMER_CLKEN = 0x2 - Clock enable for the CTIMER. CTIMER0A_CLKEN = 0x4 - Clock enable for the CTIMER0A. CTIMER0B_CLKEN = 0x8 - Clock enable for the CTIMER0B. CTIMER1A_CLKEN = 0x10 - Clock enable for the CTIMER1A. CTIMER1B_CLKEN = 0x20 - Clock enable for the CTIMER1B. CTIMER2A_CLKEN = 0x40 - Clock enable for the CTIMER2A. CTIMER2B_CLKEN = 0x80 - Clock enable for the CTIMER2B. CTIMER3A_CLKEN = 0x100 - Clock enable for the CTIMER3A. CTIMER3B_CLKEN = 0x200 - Clock enable for the CTIMER3B. IOMSTR0_CLKEN = 0x400 - Clock enable for the IO Master 0. IOMSTR1_CLKEN = 0x800 - Clock enable for the IO Master 1. IOMSTR2_CLKEN = 0x1000 - Clock enable for the IO Master 2. IOMSTR3_CLKEN = 0x2000 - Clock enable for the IO Master 3. IOMSTR4_CLKEN = 0x4000 - Clock enable for the IO Master 4. IOMSTR5_CLKEN = 0x8000 - Clock enable for the IO Master 5. IOMSTRIFC0_CLKEN = 0x10000 - Clock enable for the IO Master IFC0. IOMSTRIFC1_CLKEN = 0x20000 - Clock enable for the IO Master IFC1. IOMSTRIFC2_CLKEN = 0x40000 - Clock enable for the IO Master IFC2. IOMSTRIFC3_CLKEN = 0x80000 - Clock enable for the IO Master IFC3. IOMSTRIFC4_CLKEN = 0x100000 - Clock enable for the IO Master IFC4. IOMSTRIFC5_CLKEN = 0x200000 - Clock enable for the IO Master IFC5. IOSLAVE_CLKEN = 0x400000 - Clock enable for the IO Slave. PDM_CLKEN = 0x800000 - Clock enable for the PDM. PDMIFC_CLKEN = 0x1000000 - Clock enable for the PDM IFC. RSTGEN_CLKEN = 0x2000000 - Clock enable for the RSTGEN. SRAM_WIPE_CLKEN = 0x4000000 - Clock enable for the SRAM_WIPE. STIMER_CLKEN = 0x8000000 - Clock enable for the STIMER. STIMER_CNT_CLKEN = 0x10000000 - Clock enable for the STIMER_CNT. TPIU_CLKEN = 0x20000000 - Clock enable for the TPIU. UART0_HCLK_CLKEN = 0x40000000 - Clock enable for the UART0_HCLK. UART0HF_CLKEN = 0x80000000 - Clock enable for the UART0HF.

9.2.2.11 CLOCKEN2 Register

Clock Enable Status
OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4000402C
Clock Enable Status

Table 9-22: CLOCKEN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CLOCKEN2																															

Table 9-23: CLOCKEN2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCKEN2	0x0	RO	Clock enable status 2 UART1_HCLK_CLKEN = 0x1 - Clock enable for the UART1_HCLK. UART1HF_CLKEN = 0x2 - Clock enable for the UART1HF. WDT_CLKEN = 0x4 - Clock enable for the WDT. XT_32KHz_EN = 0x40000000 - Clock enable for the XT_32KHz. FRCHFRC = 0x80000000 - Force HFRC On Status

9.2.2.12 CLOCKEN3 Register

Clock Enable Status
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x40004030
Clock Enable Status

Table 9-24: CLOCKEN3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CLOCKEN3																															

Table 9-25: CLOCKEN3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCKEN3	0x0	RO	Clock enable status 3 periph_all_xtal_en = 0x1000000 - At least 1 peripherial is requesting for XTAL Clock periph_all_hfrc_en = 0x2000000 - At least 1 peripherial is requesting for HFRC Clock HFADJEN = 0x4000000 - HFRC Adjust Enable Status HFRC_en_out = 0x8000000 - HFRC is enabled during adjustment status RTC_SOURCE = 0x10000000 - Selects the RTC oscillator (0 => LFRC, 1 => XT) XTAL_EN = 0x20000000 - XT is enabled Status HFRC_EN = 0x40000000 - HFRC is enabled Status FLASHCLK_EN = 0x80000000 - Flash Clock is enabled Status

9.2.2.13 UARTEN Register

UART Enable
OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x40004034
UART Enable

Table 9-26: UARTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						UART1EN		RSVD							UART0EN

Table 9-27: UARTEN Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED
9:8	UART1EN	0x0	RW	UART1 system clock control DIS = 0x0 - Disable the UART1 system clock EN = 0x1 - Enable the UART1 system clock REDUCE_FREQ = 0x2 - Run UART_Hclk at the same frequency as UART_hfclk EN_POWER_SAV = 0x3 - Enable UART_hclk to reduce to UART_hfclk at low power mode
7:2	RSVD	0x0	RO	RESERVED
1:0	UART0EN	0x0	RW	UART0 system clock control DIS = 0x0 - Disable the UART0 system clock EN = 0x1 - Enable the UART0 system clock REDUCE_FREQ = 0x2 - Run UART_Hclk at the same frequency as UART_hfclk EN_POWER_SAV = 0x3 - Enable UART_hclk to reduce to UART_hfclk at low power mode

9.2.2.14 INTEN Register

CLKGEN Interrupt Register: Enable
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x40004100
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 9-28: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-29: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt

Table 9-29: INTEN Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.2.2.15 INTSTAT Register

CLKGEN Interrupt Register: Status
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x40004104
Read bits from this register to discover the cause of a recent interrupt.

Table 9-30: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-31: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.2.2.16 INTCLR Register

CLKGEN Interrupt Register: Clear
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x40004108
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 9-32: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-33: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt

Table 9-33: INTCLR Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.2.2.17 INTSET Register

CLKGEN Interrupt Register: Set
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x4000410C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 9-34: INTSET Register

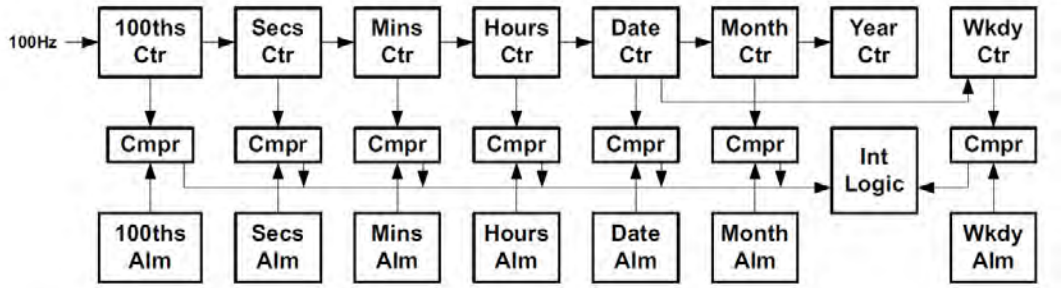
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-35: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.3 Real Time Clock

Figure 9-2: Block Diagram for the Real Time Clock Module



9.3.1 RTC Functional Overview

The Real Time Clock (RTC) Module, shown in Figure 9-2, provides an accurate real time measurement.

Key features are:

- 100^{th} of a second resolution
- Time is measured for the years between 1900 and 2199
- Automatic leap year calculation
- Hours may be specified in 12 or 24 hour mode
- Alarm precise to 1/100 second
- Alarm interval every 100^{th} second, 10^{th} second, second, minute, hour, day, week, month or year.
- 100 Hz input clock taken from either the high accuracy XT Oscillator or the low power LFRC Oscillator.

9.3.2 Calendar Counters

The real time is held in a set of eight Calendar Counters, which hold the current 1/100^{th} of a second (REG_CLK_GEN_CTRLOW_CTR100), the current second (REG_CLK_GEN_CTRLOW_CTRSEC), the minute (REG_CLK_GEN_CTRLOW_CTRMIN), the hour (REG_CLK_GEN_CTRLOW_CTRHR), the current day of the month (REG_CLK_GEN_CTRUP_CTRDATE), the current day of the week (REG_CLK_GEN_CTRUP_CTRWKDY), the current month (REG_CLK_GEN_CTRUP_CTRMO), the current year (REG_CLK_GEN_CTRUP_CTRYR) and the current century (REG_CLK_GEN_CTRUP_CB), all in BCD format. In order to insure that the RTC starts precisely, the timer chain which generates the 100 Hz clock is reset to 0 whenever any of the Calendar Counter Registers is written. Since unintentional modification of the Calendar Counters is a serious problem, the REG_CLK_GEN_RTCCTL_WRTC bit must be

set in order to write any of the counters, and should be reset by software after any load of the Calendar Counters.

Software may stop the clock to the Calendar Counters by setting the REG_CLK_GEN_RTCCTL_RSTOP bit. This may be used in modes like Stopwatch to precisely start and stop the Calendar Counters.

9.3.3 Calendar Counter Reads

The RTC includes special logic to help insure that the Calendar Counters may be read reliably, e.g., that no rollover has occurred. Because two 32-bit reads are required to read the complete set of counters, it is possible that a delay occurs between the two reads which causes a rollover to occur. An interrupt is the most likely reason this could occur. If two 100 Hz clocks occur between these two reads, the REG_CLK_GEN_CTRUP_CTRERR bit will be set. Software should check this bit after any Calendar Counter read, and perform the read again if it is set. Any read of the upper counter word will clear the CTRERR bit.

9.3.4 Alarms

There are seven Alarm Registers which may be used to generate an Alarm interrupt at a specific time.

These registers correspond to the 100th of a second (REG_CLK_GEN_ALMLOW_ALM100), second (REG_CLK_GEN_ALMLOW_ALMSEC), minute (REG_CLK_GEN_ALMLOW_ALMMIN), hour (REG_CLK_GEN_ALMLOW_ALMHR), day of the month (REG_CLK_GEN_ALMUP_ALMDATE), day of the week (REG_CLK_GEN_ALMUP_ALMWKDY) and month (REG_CLK_GEN_ALMUP_ALMMO) Calendar Counters. The comparison is controlled by the REG_CLK_GEN_RTCCTL_RPT field and the REG_CLK_GEN_ALMLOW_ALM100 Register as shown in 12/24 Hour Mode. In the ALM100 Register, n indicates any digit 0-9. When all selected Counters match their corresponding Alarm Register, the ALM interrupt flag is set (see the Clock Generator section for the ALM interrupt control).

Table 9-36: Alarm RPT Function

RPT Value	Interval	Comparison
000	Disabled	None
001	Every Year	100th, second, minute, hour, day, month
010	Every Month	100th, second, minute, hour, day
011	Every Week	100th, second, minute, hour, weekday
100	Every Day	100th, second, minute, hour
101	Every Hour	100th, second, minute
110	Every Minute	100th, second
111	Every Second	100th

All alarm interrupts are asserted on the next 100 Hz clock cycle after the counters match the alarm register, except for 100ths of a second. To get an interrupt that occurs precisely at a certain time, the comparison value in the corresponding alarm register should be set 10 ms (one 100 Hz count) earlier than the desired interrupt time.

For the 100ths of a second interrupt, the first 100 Hz clock sets the comparison with the alarm register and the next clock asserts the interrupt. Therefore, the first 100ths interrupt will be asserted after 20 ms, not 10 ms. This occurs each and every time the 100ths of a second counter with interrupts is enabled if the RTC is stopped. If the RTC is already running when configured, then the first interrupt will occur between 10 and 20 ms after configuration.

9.3.5 12/24 Hour Mode

If the REG_CLK_GEN_RTCCTL_HR1224 bit is 0, the RTC is in 24-hour mode, and the Hours and Hours Alarm Registers hold a 6-bit BCD value which is the 24-hour time (values 0 to 23). If the HR1224 bit is 1, the RTC is in 12-hour mode, and the Hours and Hours Alarm Registers hold a 5-bit BCD value which is the 12-hour time (values 1 to 12) , and bit 5 is the AP bit which is 0 for an AM time and 1 for a PM time. If the HR1224 bit is modified the Hours and Hours Alarm fields must be updated.

9.3.6 Century Control and Leap Year Management

The REG_CLK_GEN_CTRUP_CB bit indicates the current century. A value of 0 indicates the 20^{th} century, and a value of 1 indicates the 19^{th} or 21^{st} century. The CB value will toggle when the Years counter rolls over from 99 to 0 if the REG_CLK_GEN_CTRUP_CEB bit is set, and will remain constant if CEB is clear. The century value is used to control the Leap Year functions, which create the correct insertion of February 29 in years which are divisible by 4 and not divisible by 100, and also the year 2000.

9.3.7 Weekday Function

The Weekday Counter is simply a 3-bit counter which counts up to 6 and then resets to 0. It is the responsibility of software to assign particular days of the week to each counter value.

9.4 RTC Registers

Real Time Clock
INSTANCE 0 BASE ADDRESS:0x40004000

9.4.1 Register Memory Map

Table 9-37: RTC Register Map

Address(es)	Register Name	Description
0x40004040	CTRLOW	RTC Counters Lower
0x40004044	CTRUP	RTC Counters Upper
0x40004048	ALMLOW	RTC Alarms Lower
0x4000404C	ALMUP	RTC Alarms Upper
0x40004050	RTCCTL	RTC Control Register
0x40004100	INTEN	RTC Interrupt Register: Enable
0x40004104	INTSTAT	RTC Interrupt Register: Status
0x40004108	INTCLR	RTC Interrupt Register: Clear
0x4000410C	INTSET	RTC Interrupt Register: Set

9.4.2 RTC Registers

9.4.2.1 CTRLOW Register

RTC Counters Lower
OFFSET: 0x00000040
INSTANCE 0 ADDRESS: 0x40004040
RTC Counters Lower

Table 9-38: CTRLOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			CTRHR					RSVD	CTRMIN							RSVD	CTRSEC							CTR100							

Table 9-39: CTRLOW Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:24	CTRHR	0x1	RW	Hours Counter
23	RSVD	0x0	RO	RESERVED
22:16	CTRMIN	0x0	RW	Minutes Counter
15	RSVD	0x0	RO	RESERVED

Table 9-39: CTRLOW Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
14:8	CTRSEC	0x0	RW	Seconds Counter
7:0	CTR100	0x0	RW	100ths of a second Counter

9.4.2.2 CTRUP Register

RTC Counters Upper
OFFSET: 0x00000044
INSTANCE 0 ADDRESS: 0x40004044
RTC Counters Upper

Table 9-40: CTRUP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTERR	RSVD		CEB	CB	CTRWKDY			CTRYR								RSVD			CTRMO					RSVD			CTRDATE				

Table 9-41: CTRUP Register Bits

Bit	Name	Reset	RW	Description
31	CTERR	0x0	RO	Counter read error status NOERR = 0x0 - No read error occurred RDERR = 0x1 - Read error occurred
30:29	RSVD	0x0	RO	RESERVED
28	CEB	0x0	RW	Century enable DIS = 0x0 - Disable the Century bit from changing EN = 0x1 - Enable the Century bit to change
27	CB	0x0	RW	Century 2000 = 0x0 - Century is 2000s 1900_2100 = 0x1 - Century is 1900s/2100s
26:24	CTRWKDY	0x0	RW	Weekdays Counter
23:16	CTRYR	0x0	RW	Years Counter
15:13	RSVD	0x0	RO	RESERVED
12:8	CTRMO	0x0	RW	Months Counter
7:6	RSVD	0x0	RO	RESERVED
5:0	CTRDATE	0x0	RW	Date Counter

9.4.2.3 ALMLOW Register

RTC Alarms Lower
OFFSET: 0x00000048
INSTANCE 0 ADDRESS: 0x40004048
RTC Alarms Lower

Table 9-42: ALMLOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD			ALMHR					RSVD	ALMMIN							RSVD	ALMSEC							ALM100							

Table 9-43: ALMLOW Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:24	ALMHR	0x0	RW	Hours Alarm
23	RSVD	0x0	RO	RESERVED
22:16	ALMMIN	0x0	RW	Minutes Alarm
15	RSVD	0x0	RO	RESERVED
14:8	ALMSEC	0x0	RW	Seconds Alarm
7:0	ALM100	0x0	RW	100ths of a second Alarm

9.4.2.4 ALMUP Register

RTC Alarms Upper
OFFSET: 0x0000004C
INSTANCE 0 ADDRESS: 0x4000404C
RTC Alarms Upper

Table 9-44: ALMUP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD														ALMWKDY		RSVD			ALMMO					RSVD			ALMDATE			

Table 9-45: ALMUP Register Bits

Bit	Name	Reset	RW	Description
31:19	RSVD	0x0	RO	RESERVED
18:16	ALMWKDY	0x0	RW	Weekdays Alarm
15:13	RSVD	0x0	RO	RESERVED
12:8	ALMMO	0x0	RW	Months Alarm
7:6	RSVD	0x0	RO	RESERVED
5:0	ALMDATE	0x0	RW	Date Alarm

9.4.2.5 RTCCTL Register

RTC Control Register
OFFSET: 0x00000050
INSTANCE 0 ADDRESS: 0x40004050
RTC Control Register

Table 9-46: RTCCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										HR1224	RSTOP	RPT			WRTC

Table 9-47: RTCCTL Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5	HR1224	0x0	RW	Hours Counter mode 24HR = 0x0 - Hours in 24 hour mode 12HR = 0x1 - Hours in 12 hour mode
4	RSTOP	0x0	RW	RTC input clock control RUN = 0x0 - Allow the RTC input clock to run STOP = 0x1 - Stop the RTC input clock
3:1	RPT	0x0	RW	Alarm repeat interval DIS = 0x0 - Alarm interrupt disabled YEAR = 0x1 - Interrupt every year MONTH = 0x2 - Interrupt every month WEEK = 0x3 - Interrupt every week DAY = 0x4 - Interrupt every day HR = 0x5 - Interrupt every hour MIN = 0x6 - Interrupt every minute SEC = 0x7 - Interrupt every second/10th/100th
0	WRTC	0x0	RW	Counter write control DIS = 0x0 - Counter writes are disabled EN = 0x1 - Counter writes are enabled

9.4.2.6 INTEN Register

RTC Interrupt Register: Enable
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x40004100
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 9-48: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-49: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.4.2.7 INTSTAT Register

RTC Interrupt Register: Status
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x40004104
Read bits from this register to discover the cause of a recent interrupt.

Table 9-50: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-51: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.4.2.8 INTCLR Register

RTC Interrupt Register: Clear
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x40004108
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 9-52: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-53: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

9.4.2.9 INTSET Register

RTC Interrupt Register: Set
OFFSET: 0x0000010C
INSTANCE 0 ADDRESS: 0x4000410C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 9-54: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												ALM	OF	ACC	ACF

Table 9-55: INTSET Register Bits

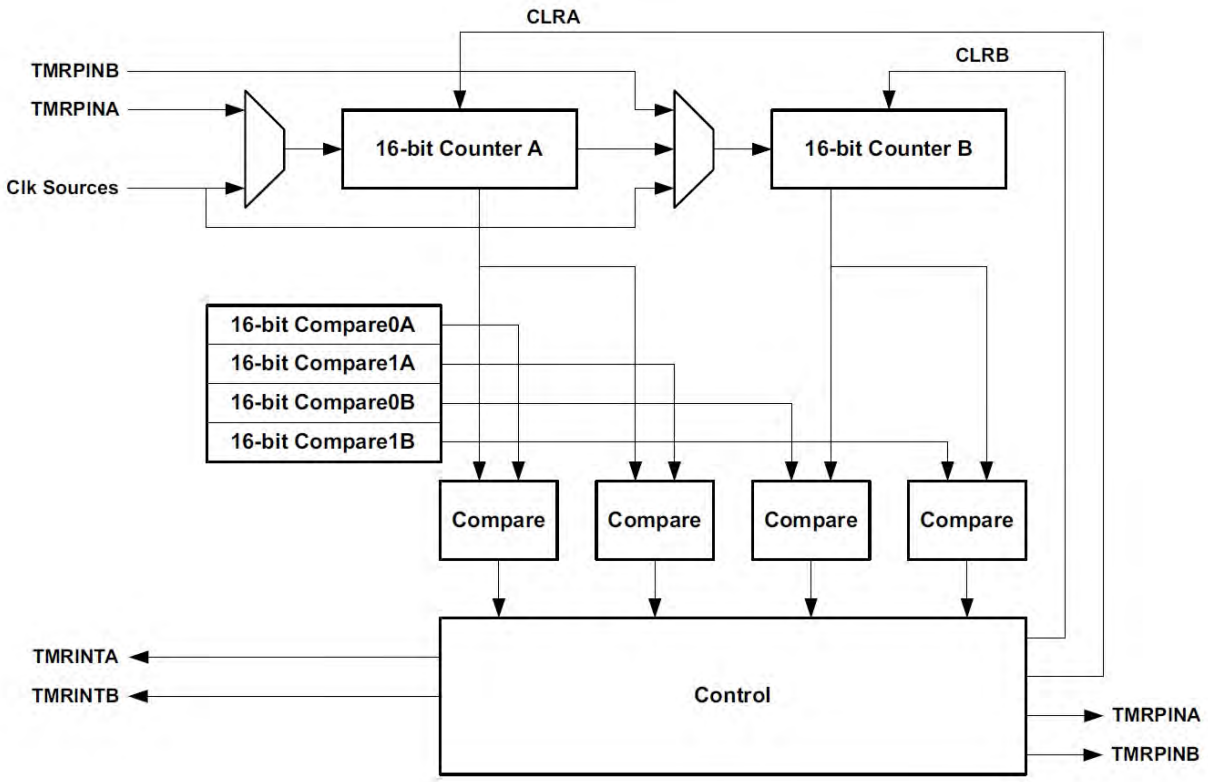
Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3	ALM	0x0	RW	RTC Alarm interrupt
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

SECTION

10

Counter/Timer Module

Figure 10-1: Block Diagram for One General Purpose Counter/Timer Pair



10.1 Functional Overview

The Apollo2 SoC Timer/Counter module includes four general purpose Timer/Counter pairs, one of which is shown in Figure 10-1. This is in addition to a system timer as described in the System Timer chapter. Each general purpose Timer/

Counter pair includes two very low power asynchronous 16-bit counters, which may be combined to provide a 32-bit counter. Four registers contain reset values for the counters and/or comparison values to allow the generation of complex signals. Each Timer/Counter has an external pin connection, which can be configured to provide a variety of outputs.

The features of the Timer Module are as follows:

- Interrupt after a specified delay
- Interrupt periodically with a specified period
- Determine the time between events
- Generate an external pulse of a specified width, configurable after a specified delay
- Generate an external PWM signal with a specified period and duty cycle
- Count edges on an external input

10.2 Counter/Timer Functions

Each Counter/Timer operates in a mode controlled by the REG_CTIMER_CTCTRLx-_TMRxyFN bit field (x=0 to 3, y=A or B). The mode affects both the generation of interrupts and the control of an external pin. Each mode is described in the following sections. Note that for all functions, a REG_CTIMER_CMPR0/1 value of zero (a count of 1) is invalid, and that the first measured period will be between the REG_CTIMER_CMPR0 value plus 2 and the specified value plus 3. Subsequent repeated cycles will be correctly of length (CMPR value + 1). There are five modes:

0 => Single Count: Timer counts from 0 to CMPR0 and stops, with an optional interrupt.

1 => Repeated Count: Periodic 1-clock-cycle wide pulses with optional interrupts.

2 => Single Pulse (One Shot): Timer counts from 0 to CMPR0, generates a pin transition and an optional interrupt, continues counting from CMPR0 to CMPR1, generates another pin transition (with no interrupt), and then stops.

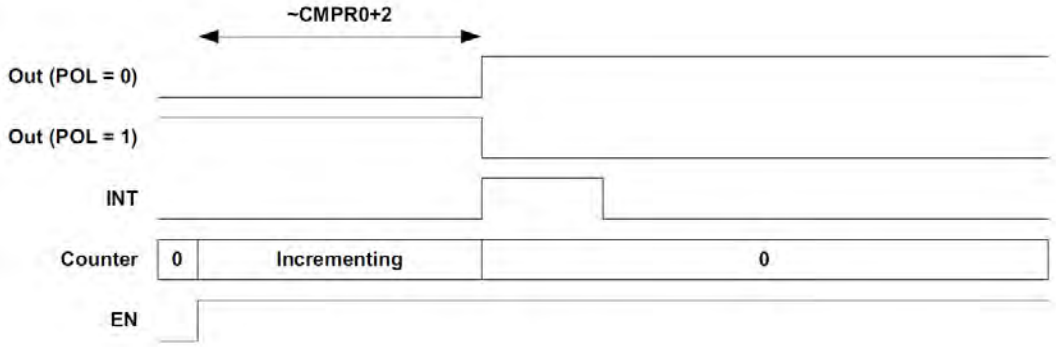
3 => Repeated Pulse: Same as single pulse, but the timer rolls over to 0 and restarts immediately after reaching CMPR1. Often used to generate PWM signals.

6 => Continuous: The timer repeatedly counts from 0 to 2^{16} - 1 forever, regardless of what the CMPR values are. The timer can optionally generate an interrupt or pin transition the first time it reaches CMPR0, but it won't generate an interrupt or pin transition on subsequent counter cycles.

10.2.1 Single Count (FN = 0)

Operation in this mode is shown in Figure 10-2. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding REG_CTIMER_CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer resets to 0 and the output pin is maintained at the selected level until the Timer is cleared with CLR. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register.

Figure 10-2: Counter/Timer Operation, FN = 0

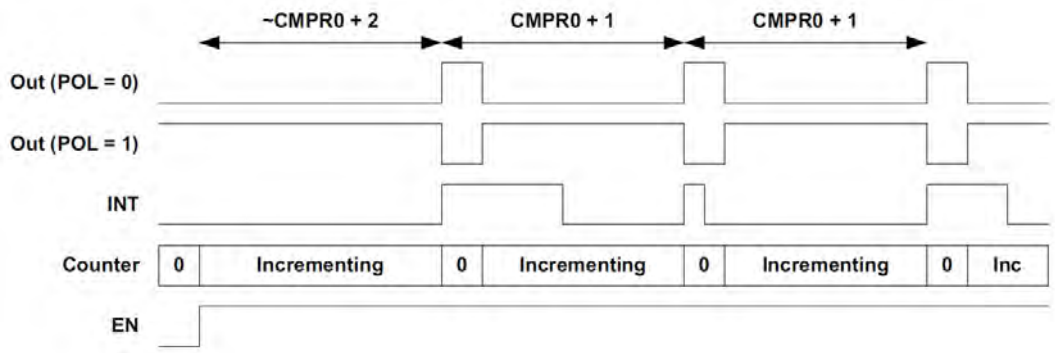


10.2.2 Repeated Count (FN = 1)

Operation in this mode is shown in Figure 10-3 on page 356. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because REG_CTIMER_TMRxyCLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the REG_CTIMER_TMRxyPE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer resets to 0 and the output pin is maintained at the selected level for one clock cycle, after which it returns to the original value. The Timer continues to count up and the process is repeated, creating a stream of pulses or interrupts at a fixed interval. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register at any point prior to the next setting pulse.

If the REG_CTIMER_TMRxyEN bit is cleared, the Timer will stop counting but will not be cleared, so the sequence may be paused and then resumed. Setting CLR will reset the Timer to zero. Note that CMPR0 must be at least 1 so that the repeat interval is two clock cycles.

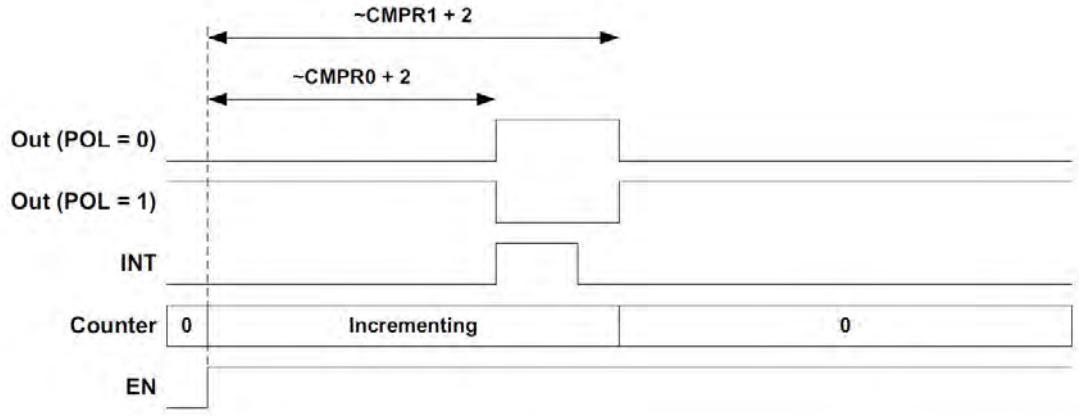
Figure 10-3: Counter/Timer Operation, FN = 1



10.2.3 Single Pulse (FN = 2)

Operation in this mode is shown in Figure 10-4. When the Timer is enabled, the pin output is at the level selected by the REG_CTIMER_TMRxyPOL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the REG_CTIMER_TMRxyPE bit is set) and an interrupt is generated (if the REG_CTIMER_TMRxyIE bit is set). At this point the Timer continues to increment and the output pin is maintained at the selected level until the Timer reaches the value in the CMPR1 Register, at which point it switches back to the original level. This allows the creation of a pulse of a specified width. The Timer is reset to 0 so that a single pulse is created. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register.

Figure 10-4: Counter/Timer Operation, FN = 2



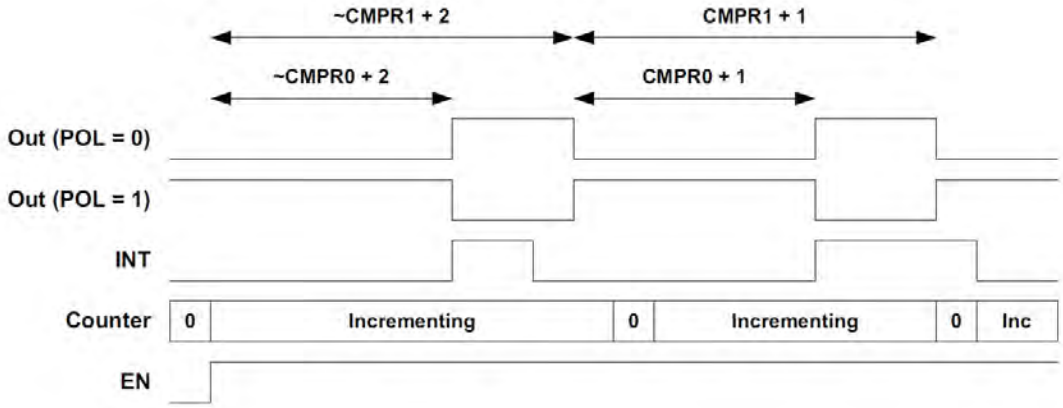
10.2.4 Repeated Pulse (FN = 3)

Operation in this mode is shown in Figure 10-5 on page 357. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each

selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer continues to increment and the output pin is maintained at the selected level until the Timer reaches the value in the CMPR1 Register, at which point it switches back to the original level. This allows the creation of a pulse of a specified width. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register. Note that CMPR1 must be at least 1 so that the repeat interval is two clock cycles.

The Timer is reset to 0 and continues to increment, so that a stream of pulses of the specified width and period is generated. If the EN bit is cleared, the Timer stops counting, but is not cleared, so the sequence may be paused and restarted. This mode is particularly valuable for creating a PWM (Pulse Width Modulation) output on the pin which may be used, for example, to vary the brightness of an LED.

Figure 10-5: Counter/Timer Operation, FN = 3

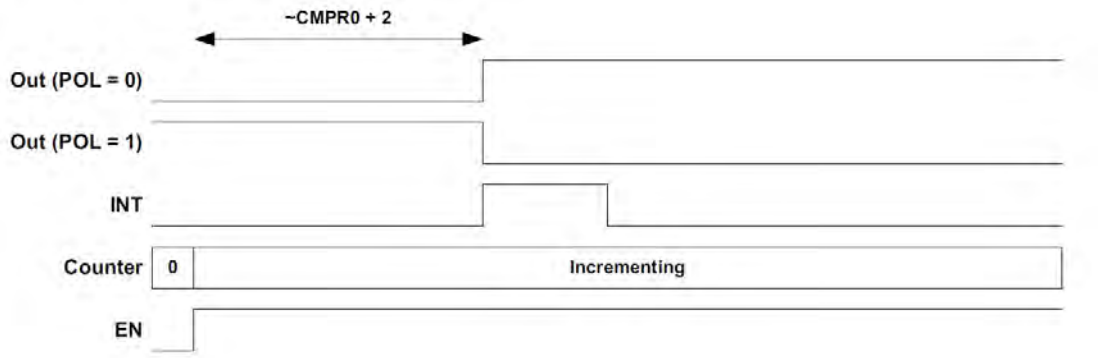


10.2.5 Continuous (FN = 4)

Operation in this mode is shown in Figure 10-6 on page 358. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). The Timer continues to count and is never automatically reset. If the Timer rolls over to zero and reaches the CMPR0 value again, an interrupt will not be generated and the output pin will not change.

This mode is primarily used for two functions. The first is counting transitions on the external input pin, and it may be valuable to generate an interrupt when a specified number of transitions have been detected. The second is as a general timer which software reads in order to measure time periods. In this second case an interrupt is often not used and will not be enabled.

Figure 10-6: Counter/Timer Operation, FN = 4



10.3 Creating 32-bit Counters

Each pair (A/B) of 16-bit counters may be combined to create a 32-bit counter. This configuration is created by setting the REG_CTIMER_CTCTRLx_CTRLINKx bit for the pair. The control bits for the A counter of the pair are used to control the 32-bit counter, and the B control bits are ignored. The CMPR0and CMPR1 registers for each 16-bit counter are concatenated to provide the 32-bit comparison values, and all timer modes are supported.

10.4 Power Optimization by Measuring HCLK_DIV4

Each timer has the capability to select the processor clock HCLK_DIV4 as the counter clock input. This allows a very straightforward measurement of how much of the time the processor is in a Sleep or Deep Sleep mode. Two counters are configured with FN = 4 so that they count continuously. One is supplied HCLK_DIV4 as its clock, and the other is supplied with a divided version of the HFRC clock. The two counters are enabled simultaneously, and after some period of system operation they are disabled and read. The HFRC count value defines how much real time has elapsed and how many HCLK_DIV4 counts could have occurred in that time, and the HCLK_DIV4 count value defines how many actual HCLK_DIV4 counts were received in that time. The scaled ratio is an accurate measurement of the percentage of time the CPU is asleep, and is an effective tool for power optimization.

10.5 Generating the Sample Rate for the ADC

Timer CTTMRA3 has a special function which allows it to function as the sample trigger generator for the ADC. If the CTIMER_ CTCTRL3 _ADCEN bit is set, the output of the timer is sent to the ADC which uses it as a trigger. Typically, Ctimer3 is configured in Repeated Count(FN =1) mode. TMRA3IE may be set to generate an

interrupt whenever the trigger occurs, but typically the ADC interrupt will be used for this purpose.

10.6 Measuring Buck Converter Charge Insertion

Each counter may be connected to a pulse stream from one of the two Buck Converters. One pulse is generated each time the Buck Converter delivers charge onto the capacitor, and therefore the number of pulses is a good indication of the amount of energy used by the corresponding power domain in a particular time period.

Following is a possible option to determine energy consumption. Two counters could be configured with FN = 4 so that they count continuously. One is supplied a Buck Converter pulse stream as its clock, and the other is supplied with a divided version of the LFRC clock to avoid creating extra power consumption due to the power measurement. Once configured such, the two counters should be enabled simultaneously, and after some period of system operation they should be disabled and read. The LFRC count value would now define how much real time has elapsed, and the Buck Converter count value would define how much energy was consumed in that time.

10.7 CTIMER Registers

Counter/Timer
INSTANCE 0 BASE ADDRESS:0x40008000
The Counter/Timer block contains 8 sixteen bit counter or timer functions. Each pair of these counters can be cascaded into 32 bit Counter/Timer functions.

10.7.1 Register Memory Map

Table 10-1: CTIMER Register Map

Address(es)	Registered Name	Description
0x40008000	TMR0	Counter/Timer Register
0x40008004	CMPRA0	Counter/Timer A0 Compare Registers
0x40008008	CMPRB0	Counter/Timer B0 Compare Registers
0x4000800C	CTRL0	Counter/Timer Control
0x40008010	TMR1	Counter/Timer Register
0x40008014	CMPRA1	Counter/Timer A1 Compare Registers
0x40008018	CMPRB1	Counter/Timer B1 Compare Registers
0x4000801C	CTRL1	Counter/Timer Control
0x40008020	TMR2	Counter/Timer Register

Table 10-1: CTIMER Register Map *(Continued)*

Address(es)	Registered Name	Description
0x40008024	CMPRA2	Counter/Timer A2 Compare Registers
0x40008028	CMPRB2	Counter/Timer B2 Compare Registers
0x4000802C	CTRL2	Counter/Timer Control
0x40008030	TMR3	Counter/Timer Register
0x40008034	CMPRA3	Counter/Timer A3 Compare Registers
0x40008038	CMPRB3	Counter/Timer B3 Compare Registers
0x4000803C	CTRL3	Counter/Timer Control
0x40008200	INTEN	Counter/Timer Interrupts: Enable
0x40008204	INTSTAT	Counter/Timer Interrupts: Status
0x40008208	INTCLR	Counter/Timer Interrupts: Clear
0x4000820C	INTSET	Counter/Timer Interrupts: Set

10.7.2 CTIMER Registers

10.7.2.1 TMR0 Register

Counter/Timer Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40008000
This register holds the running time or event count, either for each 16 bit half or for the whole 32 bit count when the pair is linked.

Table 10-2: TMR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB0																CTTMRA0															

Table 10-3: TMR0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB0	0x0	RO	Counter/Timer B0.
15:0	CTTMRA0	0x0	RO	Counter/Timer A0.

10.7.2.2 CMPRA0 Register

Counter/Timer A0 Compare Registers
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40008004
Compare limits for timer half A.

Table 10-4: CMPRA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A0																CMPR0A0															

Table 10-5: CMPRA0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A0	0x0	RW	Counter/Timer A0 Compare Register 1. Holds the upper limit for timer half A.
15:0	CMPR0A0	0x0	RW	Counter/Timer A0 Compare Register 0. Holds the lower limit for timer half A.

10.7.2.3 CMPRB0 Register

Counter/Timer B0 Compare Registers
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40008008
Compare limits for timer half B.

Table 10-6: CMPRB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B0																CMPR0B0															

Table 10-7: CMPRB0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B0	0x0	RW	Counter/Timer B0 Compare Register 1. Holds the upper limit for timer half B.
15:0	CMPR0B0	0x0	RW	Counter/Timer B0 Compare Register 0. Holds the lower limit for timer half B.

10.7.2.4 CTRL0 Register

Counter/Timer Control
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4000800C
Control bit fields for both halves of timer 0.

Table 10-8: CTRL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK0	RSVD	TMRB0PE	TMRB0POL	TMRB0CLR	TMRB0IE1	TMRB0IE0	TMRB0FN			TMRB0CLK					TMRB0EN	RSVD		TMRA0PE	TMRA0POL	TMRA0CLR	TMRA0IE1	TMRA0IE0	TMRA0FN			TMRA0CLK					TMRA0EN

Table 10-9: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK0	0x0	RW	Counter/Timer A0/B0 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A0/B0 timers into a single 32-bit timer.
30	RSVD	0x0	RO	RESERVED
29	TMRB0PE	0x0	RW	Counter/Timer B0 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB0POL. EN = 0x1 - Enable counter/timer B0 to generate a signal on TMRPINB.
28	TMRB0POL	0x0	RW	Counter/Timer B0 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB0 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB0 pin is the inverse of the timer output.
27	TMRB0CLR	0x0	RW	Counter/Timer B0 Clear bit. RUN = 0x0 - Allow counter/timer B0 to run CLEAR = 0x1 - Holds counter/timer B0 at 0x0000.
26	TMRB0IE1	0x0	RW	Counter/Timer B0 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B0 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B0 to generate an interrupt based on COMPR1.
25	TMRB0IE0	0x0	RW	Counter/Timer B0 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B0 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B0 to generate an interrupt based on COMPR0
24:22	TMRB0FN	0x0	RW	Counter/Timer B0 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B0, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B0, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B0, assert, count to CMPR1B0, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B0, assert, count to CMPR1B0, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.

Table 10-9: CTRL0 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
21:17	TMRB0CLK	0x0	RW	Counter/Timer B0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKB = 0x10 - Clock source is buck converter stream from CORE Buck.
16	TMRB0EN	0x0	RW	Counter/Timer B0 Enable bit. DIS = 0x0 - Counter/Timer B0 Disable. EN = 0x1 - Counter/Timer B0 Enable.
15:14	RSVD	0x0	RO	RESERVED
13	TMRA0PE	0x0	RW	Counter/Timer A0 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA0POL. EN = 0x1 - Enable counter/timer A0 to generate a signal on TMRPINA.
12	TMRA0POL	0x0	RW	Counter/Timer A0 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA0 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA0 pin is the inverse of the timer output.
11	TMRA0CLR	0x0	RW	Counter/Timer A0 Clear bit. RUN = 0x0 - Allow counter/timer A0 to run CLEAR = 0x1 - Holds counter/timer A0 at 0x0000.
10	TMRA0IE1	0x0	RW	Counter/Timer A0 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A0 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A0 to generate an interrupt based on COMPR1.
9	TMRA0IE0	0x0	RW	Counter/Timer A0 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A0 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A0 to generate an interrupt based on COMPR0.

Table 10-9: CTRL0 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
8:6	TMRA0FN	0x0	RW	Counter/Timer A0 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A0, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A0, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A0, assert, count to CMPR1A0, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A0, assert, count to CMPR1A0, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA0CLK	0x0	RW	Counter/Timer A0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKA = 0x10 - Clock source is buck converter stream from MEM Buck.
0	TMRA0EN	0x0	RW	Counter/Timer A0 Enable bit. DIS = 0x0 - Counter/Timer A0 Disable. EN = 0x1 - Counter/Timer A0 Enable.

10.7.2.5 TMR1 Register

Counter/Timer Register
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40008010
This register holds the running time or event count, either for each 16 bit half or for the whole 32 bit count when the pair is linked.

Table 10-10: TMR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB1																CTTMRA1															

Table 10-11: TMR1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB1	0x0	RO	Counter/Timer B1.
15:0	CTTMRA1	0x0	RO	Counter/Timer A1.

10.7.2.6 CMPRA1 Register

Counter/Timer A1 Compare Registers
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x40008014
This register holds the compare limits for timer half A.

Table 10-12: CMPRA1 Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
CMPR1A1	CMPR0A1

Table 10-13: CMPRA1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A1	0x0	RW	Counter/Timer A1 Compare Register 1.
15:0	CMPR0A1	0x0	RW	Counter/Timer A1 Compare Register 0.

10.7.2.7 CMPRB1 Register

Counter/Timer B1 Compare Registers
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x40008018
This register holds the compare limits for timer half B.

Table 10-14: CMPRB1 Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
CMPR1B1	CMPR0B1

Table 10-15: CMPRB1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B1	0x0	RW	Counter/Timer B1 Compare Register 1.
15:0	CMPR0B1	0x0	RW	Counter/Timer B1 Compare Register 0.

10.7.2.8 CTRL1 Register

Counter/Timer Control
OFFSET: 0x0000001C
INSTANCE 0 ADDRESS: 0x4000801C
Control bit fields for both halves of timer 0.

Table 10-16: CTRL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK1	RSVD	TMRB1PE	TMRB1POL	TMRB1CLR	TMRB1IE1	TMRB1IE0	TMRB1FN	TMRB1CLK							TMRB1EN	RSVD		TMRA1PE	TMRA1POL	TMRA1CLR	TMRA1IE1	TMRA1IE0	TMRA1FN	TMRA1CLK							TMRA1EN

Table 10-17: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK1	0x0	RW	Counter/Timer A1/B1 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A1/B1 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A1/B1 timers into a single 32-bit timer.
30	RSVD	0x0	RO	RESERVED
29	TMRB1PE	0x0	RW	Counter/Timer B1 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB1POL. EN = 0x1 - Enable counter/timer B1 to generate a signal on TMRPINB.
28	TMRB1POL	0x0	RW	Counter/Timer B1 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB1 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB1 pin is the inverse of the timer output.
27	TMRB1CLR	0x0	RW	Counter/Timer B1 Clear bit. RUN = 0x0 - Allow counter/timer B1 to run CLEAR = 0x1 - Holds counter/timer B1 at 0x0000.
26	TMRB1IE1	0x0	RW	Counter/Timer B1 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B1 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B1 to generate an interrupt based on COMPR1.
25	TMRB1IE0	0x0	RW	Counter/Timer B1 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B1 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B1 to generate an interrupt based on COMPR0

Table 10-17: CTRL1 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
24:22	TMRB1FN	0x0	RW	Counter/Timer B1 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B1, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B1, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B1, assert, count to CMPR1B1, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B1, assert, count to CMPR1B1, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
21:17	TMRB1CLK	0x0	RW	Counter/Timer B1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKB = 0x10 - Clock source is buck converter stream from CORE Buck.
16	TMRB1EN	0x0	RW	Counter/Timer B1 Enable bit. DIS = 0x0 - Counter/Timer B1 Disable. EN = 0x1 - Counter/Timer B1 Enable.
15:14	RSVD	0x0	RO	RESERVED
13	TMRA1PE	0x0	RW	Counter/Timer A1 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA1POL. EN = 0x1 - Enable counter/timer A1 to generate a signal on TMRPINA.
12	TMRA1POL	0x0	RW	Counter/Timer A1 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA1 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA1 pin is the inverse of the timer output.
11	TMRA1CLR	0x0	RW	Counter/Timer A1 Clear bit. RUN = 0x0 - Allow counter/timer A1 to run CLEAR = 0x1 - Holds counter/timer A1 at 0x0000.
10	TMRA1IE1	0x0	RW	Counter/Timer A1 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A1 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A1 to generate an interrupt based on COMPR1.

Table 10-17: CTRL1 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
9	TMRA1IE0	0x0	RW	Counter/Timer A1 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A1 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A1 to generate an interrupt based on COMPR0.
8:6	TMRA1FN	0x0	RW	Counter/Timer A1 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A1, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A1, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A1, assert, count to CMPR1A1, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A1, assert, count to CMPR1A1, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA1CLK	0x0	RW	Counter/Timer A1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKA = 0x10 - Clock source is buck converter stream from MEM Buck.
0	TMRA1EN	0x0	RW	Counter/Timer A1 Enable bit. DIS = 0x0 - Counter/Timer A1 Disable. EN = 0x1 - Counter/Timer A1 Enable.

10.7.2.9 TMR2 Register

Counter/Timer Register
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x40008020
Counter/Timer Register

Table 10-18: TMR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB2																CTTMRA2															

Table 10-19: TMR2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB2	0x0	RO	Counter/Timer B2.
15:0	CTTMRA2	0x0	RO	Counter/Timer A2.

10.7.2.10 CMPRA2 Register

Counter/Timer A2 Compare Registers
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x40008024
This register holds the compare limits for timer half A.

Table 10-20: CMPRA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A2																CMPR0A2															

Table 10-21: CMPRA2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A2	0x0	RW	Counter/Timer A2 Compare Register 1.
15:0	CMPR0A2	0x0	RW	Counter/Timer A2 Compare Register 0.

10.7.2.11 CMPRB2 Register

Counter/Timer B2 Compare Registers
OFFSET: 0x00000028
INSTANCE 0 ADDRESS: 0x40008028
This register holds the compare limits for timer half B.

Table 10-22: CMPRB2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B2																CMPR0B2															

Table 10-23: CMPRB2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B2	0x0	RW	Counter/Timer B2 Compare Register 1.
15:0	CMPR0B2	0x0	RW	Counter/Timer B2 Compare Register 0.

10.7.2.12 CTRL2 Register

Counter/Timer Control
OFFSET: 0x0000002C
INSTANCE 0 ADDRESS: 0x4000802C
This register holds the control bit fields for both halves of timer 2.

Table 10-24: CTRL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK2	RSVD	TMRB2PE	TMRB2POL	TMRB2CLR	TMRB2IE1	TMRB2IE0	TMRB2FN			TMRB2CLK					TMRB2EN	RSVD		TMRA2PE	TMRA2POL	TMRA2CLR	TMRA2IE1	TMRA2IE0	TMRA2FN			TMRA2CLK					TMRA2EN

Table 10-25: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK2	0x0	RW	Counter/Timer A2/B2 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A2/B2 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A2/B2 timers into a single 32-bit timer.
30	RSVD	0x0	RO	RESERVED
29	TMRB2PE	0x0	RW	Counter/Timer B2 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB2POL. EN = 0x1 - Enable counter/timer B2 to generate a signal on TMRPINB.
28	TMRB2POL	0x0	RW	Counter/Timer B2 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB2 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB2 pin is the inverse of the timer output.
27	TMRB2CLR	0x0	RW	Counter/Timer B2 Clear bit. RUN = 0x0 - Allow counter/timer B2 to run CLEAR = 0x1 - Holds counter/timer B2 at 0x0000.
26	TMRB2IE1	0x0	RW	Counter/Timer B2 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B2 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B2 to generate an interrupt based on COMPR1.

Table 10-25: CTRL2 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
25	TMRB2IE0	0x0	RW	Counter/Timer B2 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B2 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B2 to generate an interrupt based on COMPR0
24:22	TMRB2FN	0x0	RW	Counter/Timer B2 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B2, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B2, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B2, assert, count to CMPR1B2, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B2, assert, count to CMPR1B2, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
21:17	TMRB2CLK	0x0	RW	Counter/Timer B2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKA = 0x10 - Clock source is buck converter stream from MEM Buck.
16	TMRB2EN	0x0	RW	Counter/Timer B2 Enable bit. DIS = 0x0 - Counter/Timer B2 Disable. EN = 0x1 - Counter/Timer B2 Enable.
15:14	RSVD	0x0	RO	RESERVED
13	TMRA2PE	0x0	RW	Counter/Timer A2 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA2POL. EN = 0x1 - Enable counter/timer A2 to generate a signal on TMRPINA.
12	TMRA2POL	0x0	RW	Counter/Timer A2 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA2 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA2 pin is the inverse of the timer output.

Table 10-25: CTRL2 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
11	TMRA2CLR	0x0	RW	Counter/Timer A2 Clear bit. RUN = 0x0 - Allow counter/timer A2 to run CLEAR = 0x1 - Holds counter/timer A2 at 0x0000.
10	TMRA2IE1	0x0	RW	Counter/Timer A2 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A2 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A2 to generate an interrupt based on COMPR1.
9	TMRA2IE0	0x0	RW	Counter/Timer A2 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A2 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A2 to generate an interrupt based on COMPR0.
8:6	TMRA2FN	0x0	RW	Counter/Timer A2 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A2, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A2, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A2, assert, count to CMPR1A2, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A2, assert, count to CMPR1A2, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA2CLK	0x0	RW	Counter/Timer A2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKB = 0x10 - Clock source is buck converter stream from CORE Buck.
0	TMRA2EN	0x0	RW	Counter/Timer A2 Enable bit. DIS = 0x0 - Counter/Timer A2 Disable. EN = 0x1 - Counter/Timer A2 Enable.

10.7.2.13 TMR3 Register

Counter/Timer Register
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x40008030
Counter/Timer Register

Table 10-26: TMR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB3																CTTMRA3															

Table 10-27: TMR3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB3	0x0	RO	Counter/Timer B3.
15:0	CTTMRA3	0x0	RO	Counter/Timer A3.

10.7.2.14 CMPRA3 Register

Counter/Timer A3 Compare Registers
OFFSET: 0x00000034
INSTANCE 0 ADDRESS: 0x40008034
This register holds the compare limits for timer half A.

Table 10-28: CMPRA3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A3																CMPR0A3															

Table 10-29: CMPRA3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A3	0x0	RW	Counter/Timer A3 Compare Register 1.
15:0	CMPR0A3	0x0	RW	Counter/Timer A3 Compare Register 0.

10.7.2.15 CMPRB3 Register

Counter/Timer B3 Compare Registers
OFFSET: 0x00000038
INSTANCE 0 ADDRESS: 0x40008038
This register holds the compare limits for timer half B.

Table 10-30: CMPRB3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B3																CMPR0B3															

Table 10-31: CMPRB3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B3	0x0	RW	Counter/Timer B3 Compare Register 1.
15:0	CMPR0B3	0x0	RW	Counter/Timer B3 Compare Register 0.

10.7.2.16 CTRL3 Register

Counter/Timer Control
OFFSET: 0x0000003C
INSTANCE 0 ADDRESS: 0x4000803C
This register holds the control bit fields for both halves of timer 3.

Table 10-32: CTRL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK3	RSVD	TMRB3PE	TMRB3POL	TMRB3CLR	TMRB3IE1	TMRB3IE0	TMRB3FN			TMRB3CLK					TMRB3EN	ADCEN	RSVD	TMRA3PE	TMRA3POL	TMRA3CLR	TMRA3IE1	TMRA3IE0	TMRA3FN			TMRA3CLK				TMRA3EN

Table 10-33: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK3	0x0	RW	Counter/Timer A3/B3 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A3/B3 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A3/B3 timers into a single 32-bit timer.
30	RSVD	0x0	RO	RESERVED
29	TMRB3PE	0x0	RW	Counter/Timer B3 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB3POL. EN = 0x1 - Enable counter/timer B3 to generate a signal on TMRPINB.
28	TMRB3POL	0x0	RW	Counter/Timer B3 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB3 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB3 pin is the inverse of the timer output.
27	TMRB3CLR	0x0	RW	Counter/Timer B3 Clear bit. RUN = 0x0 - Allow counter/timer B3 to run CLEAR = 0x1 - Holds counter/timer B3 at 0x0000.

Table 10-33: CTRL3 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
26	TMRB3IE1	0x0	RW	Counter/Timer B3 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B3 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B3 to generate an interrupt based on COMPR1.
25	TMRB3IE0	0x0	RW	Counter/Timer B3 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B3 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B3 to generate an interrupt based on COMPR0
24:22	TMRB3FN	0x0	RW	Counter/Timer B3 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B3, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B3, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B3, assert, count to CMPR1B3, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B3, assert, count to CMPR1B3, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
21:17	TMRB3CLK	0x0	RW	Counter/Timer B3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKA = 0x10 - Clock source is buck converter stream from MEM Buck.
16	TMRB3EN	0x0	RW	Counter/Timer B3 Enable bit. DIS = 0x0 - Counter/Timer B3 Disable. EN = 0x1 - Counter/Timer B3 Enable.
15	ADCEN	0x0	RW	Special Timer A3 enable for ADC function.
14	RSVD	0x0	RO	RESERVED
13	TMRA3PE	0x0	RW	Counter/Timer A3 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA3POL. EN = 0x1 - Enable counter/timer A3 to generate a signal on TMRPINA.

Table 10-33: CTRL3 Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
12	TMRA3POL	0x0	RW	Counter/Timer A3 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA3 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA3 pin is the inverse of the timer output.
11	TMRA3CLR	0x0	RW	Counter/Timer A3 Clear bit. RUN = 0x0 - Allow counter/timer A3 to run CLEAR = 0x1 - Holds counter/timer A3 at 0x0000.
10	TMRA3IE1	0x0	RW	Counter/Timer A3 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A3 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A3 to generate an interrupt based on COMPR1.
9	TMRA3IE0	0x0	RW	Counter/Timer A3 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A3 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A3 to generate an interrupt based on COMPR0.
8:6	TMRA3FN	0x0	RW	Counter/Timer A3 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A3, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A3, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A3, assert, count to CMPR1A3, de-assert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A3, assert, count to CMPR1A3, de-assert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA3CLK	0x0	RW	Counter/Timer A3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is HFRC / 4 HFRC_DIV16 = 0x2 - Clock source is HFRC / 16 HFRC_DIV256 = 0x3 - Clock source is HFRC / 256 HFRC_DIV1024 = 0x4 - Clock source is HFRC / 1024 HFRC_DIV4K = 0x5 - Clock source is HFRC / 4096 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK_DIV4 = 0xF - Clock source is HCLK / 4 BUCKB = 0x10 - Clock source is buck converter stream from CORE Buck.
0	TMRA3EN	0x0	RW	Counter/Timer A3 Enable bit. DIS = 0x0 - Counter/Timer A3 Disable. EN = 0x1 - Counter/Timer A3 Enable.

10.7.2.17 INTEN Register

Counter/Timer Interrupts: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x40008200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 10-34: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 10-35: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
14	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
13	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
12	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
11	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
10	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
9	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
8	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
7	CTMRB3C0INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR0.
6	CTMRA3C0INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR0.
5	CTMRB2C0INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR0.
4	CTMRA2C0INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR0.
3	CTMRB1C0INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR0.
2	CTMRA1C0INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR0.
1	CTMRB0C0INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR0.
0	CTMRA0C0INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR0.

10.7.2.18 INTSTAT Register

Counter/Timer Interrupts: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x40008204
Read bits from this register to discover the cause of a recent interrupt.

Table 10-36: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 10-37: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
14	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
13	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
12	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
11	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
10	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
9	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
8	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
7	CTMRB3C0INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR0.
6	CTMRA3C0INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR0.
5	CTMRB2C0INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR0.
4	CTMRA2C0INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR0.
3	CTMRB1C0INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR0.
2	CTMRA1C0INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR0.
1	CTMRB0C0INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR0.
0	CTMRA0C0INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR0.

10.7.2.19 INTCLR Register

Counter/Timer Interrupts: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x40008208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 10-38: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 10-39: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
14	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
13	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
12	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
11	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
10	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
9	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
8	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
7	CTMRB3C0INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR0.
6	CTMRA3C0INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR0.
5	CTMRB2C0INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR0.
4	CTMRA2C0INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR0.
3	CTMRB1C0INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR0.
2	CTMRA1C0INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR0.
1	CTMRB0C0INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR0.
0	CTMRA0C0INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR0.

10.7.2.20 INTSET Register

Counter/Timer Interrupts: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x4000820C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 10-40: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 10-41: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
14	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
13	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.

Table 10-41: INTSET Register Bits *(Continued)*

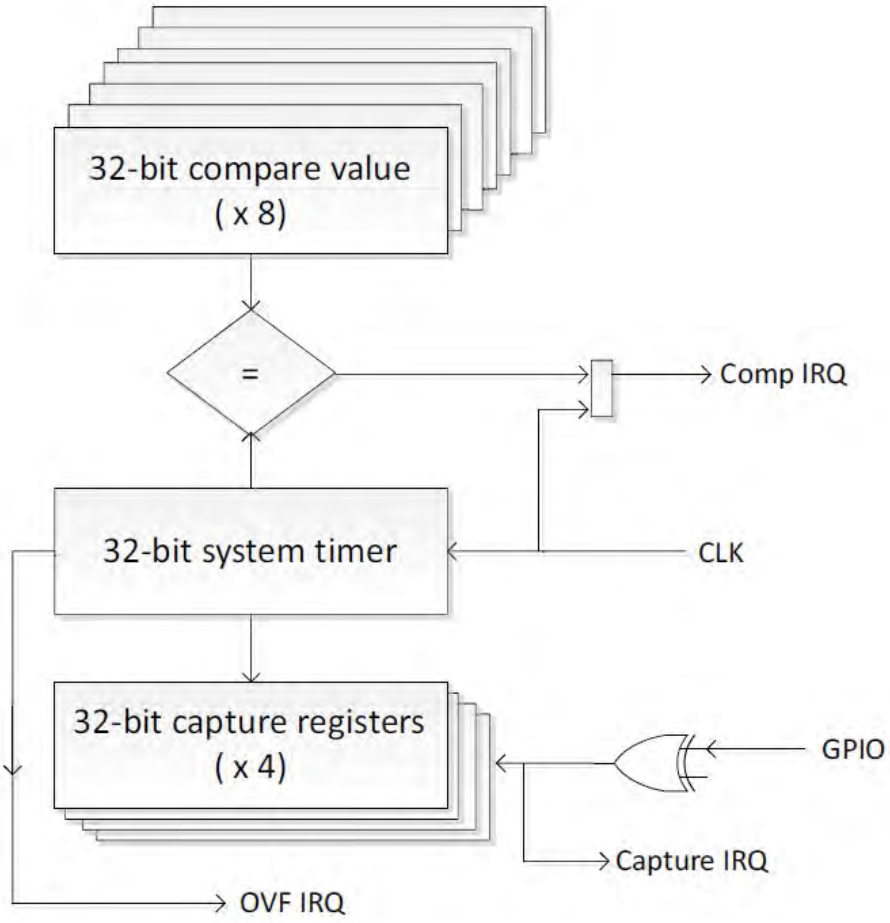
Bit	Name	Reset	RW	Description
12	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
11	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
10	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
9	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
8	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
7	CTMRB3C0INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR0.
6	CTMRA3C0INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR0.
5	CTMRB2C0INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR0.
4	CTMRA2C0INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR0.
3	CTMRB1C0INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR0.
2	CTMRA1C0INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR0.
1	CTMRB0C0INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR0.
0	CTMRA0C0INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR0.

SECTION

11

System Timer Module

Figure 11-1: Block Diagram for the System Timer



11.1 Functional Overview

The Apollo2 SoC System Timer (STIMER), shown above in Figure 11-1 on page 381, tracks the global synchronized counter. It can be used for RTOS scheduling and real-time system tracking. This timer is provided in addition to the other timer peripherals to enable software/firmware to have a simple, globally synchronized timer source.

The System Timer (STIMER) Module provides real time measurement for all task scheduling, sensor sample rate calibration, and tracking of real time and calendar maintenance. Key features are:

- 32-bit binary counter used for RTOS scheduling decisions.
- Eight 32-bit compare and interrupt registers to facilitate light weight scheduling (designs without RTOS).
- Accurate scheduling of comparator interrupts
- Only offsets from “NOW” are written to comparator registers.
- Maintains real time epoch for applications.
- Overflow interrupt to allow firmware to keep the extended part (more than 32-bits) of real time epoch.
- Time stamping hardware for multiple sensor streams (4 capture registers).
- Firmware handling of odd calculations such as Leap Second. It also handles things like surprise/legislated changes to the daylight savings time transition dates.
- Firmware handling of 1024 versus 1000 scaling of real time conversions.
- Only reset by POA (Power On Analog - system cold reset) so that it retains time across all POI and POR (system warm reset) events except full power cycles.
- Contains three 32-bit NVRAM registers that are only reset by POA to maintain real time offset from epoch.

The heart of the STIMER is a single 32-bit counter that keeps track of current time for the application running on the Apollo2 SoC. This counter is reset at the actual power cycle reset of the SoC. It is generally never reset or changed again. Up to eight 32-bit comparator registers can be loaded each of which can generate an interrupt signal to the NVIC. Comparators A through H generate interrupt A through H while capture registers A through D and the overflow event generate interrupt I, all the way to the NVIC. Thus the scheduler can run these 9 interrupts at different priorities in the NVIC.

The comparator interrupts are each used to schedule a function (task) to run for the application. Thus these tasks run on interrupt levels at priorities lower than the I/O interrupts. The overflow interrupt allows firmware to keep track of real time beyond that maintained in the 32-bit timer.

11.2 STIMER Registers

System Timer
INSTANCE 0 BASE ADDRESS:0x40008000
The System Timer block contains a 32-bit counter for system timer functions. This counter is the source for time-stamping events when performing capture or compare functions.

11.2.1 Register Memory Map

Table 11-1: STIMER Register Map

Address(es)	Registered Name	Description
0x40008100	STCFG	Configuration Register
0x40008104	STTMR	System Timer Count Register (Real Time Counter)
0x40008108	CAPTURE_CONTROL	Capture Control Register
0x40008110	SCMPR0	Compare Register A
0x40008114	SCMPR1	Compare Register B
0x40008118	SCMPR2	Compare Register C
0x4000811C	SCMPR3	Compare Register D
0x40008120	SCMPR4	Compare Register E
0x40008124	SCMPR5	Compare Register F
0x40008128	SCMPR6	Compare Register G
0x4000812C	SCMPR7	Compare Register H
0x400081E0	SCAPT0	Capture Register A
0x400081E4	SCAPT1	Capture Register B
0x400081E8	SCAPT2	Capture Register C
0x400081EC	SCAPT3	Capture Register D
0x400081F0	SNVR0	System Timer NVRAM_A Register
0x400081F4	SNVR1	System Timer NVRAM_B Register
0x400081F8	SNVR2	System Timer NVRAM_C Register
0x40008300	STMINTEN	STIMER Interrupt registers: Enable
0x40008304	STMINTSTAT	STIMER Interrupt registers: Status
0x40008308	STMINTCLR	STIMER Interrupt registers: Clear
0x4000830C	STMINTSET	STIMER Interrupt registers: Set

11.2.2 STIMER Registers

11.2.2.1 STCFG Register

Configuration Register
OFFSET: 0x00000100
INSTANCE 0 ADDRESS: 0x40008100

The STIMER Configuration Register contains the software control for selecting the clock divider and source feeding the system timer.

Table 11-2: STCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FREEZE	CLEAR	RSVD														COMPARE_H_EN	COMPARE_G_EN	COMPARE_F_EN	COMPARE_E_EN	COMPARE_D_EN	COMPARE_C_EN	COMPARE_B_EN	COMPARE_A_EN	RSVD				CLKSEL			

Table 11-3: STCFG Register Bits

Bit	Name	Reset	RW	Description
31	FREEZE	0x1	RW	Set this bit to one to freeze the clock input to the COUNTER register. Once frozen, the value can be safely written from the SoC. Unfreeze to resume. THAW = 0x0 - Let the COUNTER register run on its input clock. FREEZE = 0x1 - Stop the COUNTER register for loading.
30	CLEAR	0x0	RW	Set this bit to one to clear the System Timer register. If this bit is set to '1', the system timer register will stay cleared. It needs to be set to '0' for the system timer to start running. RUN = 0x0 - Let the COUNTER register run on its input clock. CLEAR = 0x1 - Stop the COUNTER register for loading.
29:16	RSVD	0x0	RO	RESERVED.
15	COMPARE_H_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare H disabled. ENABLE = 0x1 - Compare H enabled.
14	COMPARE_G_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare G disabled. ENABLE = 0x1 - Compare G enabled.
13	COMPARE_F_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare F disabled. ENABLE = 0x1 - Compare F enabled.

Table 11-3: STCFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
12	COMPARE_E_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare E disabled. ENABLE = 0x1 - Compare E enabled.
11	COMPARE_D_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare D disabled. ENABLE = 0x1 - Compare D enabled.
10	COMPARE_C_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare C disabled. ENABLE = 0x1 - Compare C enabled.
9	COMPARE_B_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare B disabled. ENABLE = 0x1 - Compare B enabled.
8	COMPARE_A_EN	0x0	RW	Selects whether compare is enabled for the corresponding SCMPR register. If compare is enabled, the interrupt status is set once the comparison is met. DISABLE = 0x0 - Compare A disabled. ENABLE = 0x1 - Compare A enabled.
7:4	RSVD	0x0	RO	RESERVED.
3:0	CLKSEL	0x0	RW	Selects an appropriate clock source and divider to use for the System Timer clock. NOCLK = 0x0 - No clock enabled. HFRC_DIV16 = 0x1 - 3MHz from the HFRC clock divider. HFRC_DIV256 = 0x2 - 187.5KHz from the HFRC clock divider. XTAL_DIV1 = 0x3 - 32768Hz from the crystal oscillator. XTAL_DIV2 = 0x4 - 16384Hz from the crystal oscillator. XTAL_DIV32 = 0x5 - 1024Hz from the crystal oscillator. LFRC_DIV1 = 0x6 - Approximately 1KHz from the LFRC oscillator (uncalibrated). CTIMER0A = 0x7 - Use CTIMER 0 section A as a prescaler for the clock source. CTIMER0B = 0x8 - Use CTIMER 0 section B (or A and B linked together) as a prescaler for the clock source.

11.2.2.2 STTMR Register

System Timer Count Register (Real Time Counter)
OFFSET: 0x00000104
INSTANCE 0 ADDRESS: 0x40008104

The System Timer Count Register (STTMR) stores a running count of time as determined by its count being incremented on every rising clock edge of the clock source selected in the CLKSEL field of the REG_STIMER_STCFG register. It is this

counter value that is captured in the capture registers and it is this counter value that is compared against the various compare registers. This register cannot be written, but can be cleared to 0 for a deterministic value, by setting the REG_STIMER_STCFG_FREEZE bit to stop the counter from incrementing, and then setting the REG_STIMER_STCFG_CLEAR bit. Clear the REG_STIMER_STCFG_FREEZE bit to restart the counter.

Table 11-4: STTMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-5: STTMR Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Value of the 32-bit counter as it ticks over.

11.2.2.3 CAPTURE_CONTROL Register

Capture Control Register
OFFSET: 0x00000108
INSTANCE 0 ADDRESS: 0x40008108

The STIMER Capture Control Register controls each of the 4 capture registers. It selects their GPIO pin number for a trigger source, enables a capture operation and sets the input polarity for the capture. Note that 8-bit writes can control individual capture registers atomically.

Table 11-6: CAPTURE_CONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												CAPTURE_D	CAPTURE_C	CAPTURE_B	CAPTURE_A

Table 11-7: CAPTURE_CONTROL Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED.
3	CAPTURE_D	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.
2	CAPTURE_C	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.

Table 11-7: CAPTURE_CONTROL Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
1	CAPTURE_B	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.
0	CAPTURE_A	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.

11.2.2.4 SCMPR0 Register

Compare Register A
OFFSET: 0x00000110
INSTANCE 0 ADDRESS: 0x40008110

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-8: SCMPR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-9: SCMPR0 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_A_EN bit in the REG_CTIMER_STCGF register.

11.2.2.5 SCMPR1 Register

Compare Register B
OFFSET: 0x00000114
INSTANCE 0 ADDRESS: 0x40008114

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The

hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-10: SCMPR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-11: SCMPR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_B_EN bit in the REG_CTIMER_STCGF register.

11.2.2.6 SCMPR2 Register

Compare Register C
OFFSET: 0x00000118
INSTANCE 0 ADDRESS: 0x40008118

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur..

Table 11-12: SCMPR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-13: SCMPR2 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_C_EN bit in the REG_CTIMER_STCGF register.

11.2.2.7 SCMPR3 Register

Compare Register D
OFFSET: 0x0000011C
INSTANCE 0 ADDRESS: 0x4000811C

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-14: SCMPR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-15: SCMPR3 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_D_EN bit in the REG_CTIMER_STCGF register.

11.2.2.8 SCMPR4 Register

Compare Register E
OFFSET: 0x00000120
INSTANCE 0 ADDRESS: 0x40008120

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-16: SCMPR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-17: SCMPR4 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_E_EN bit in the REG_CTIMER_STCGF register.

11.2.2.9 SCMPR5 Register

Compare Register F
OFFSET: 0x00000124
INSTANCE 0 ADDRESS: 0x40008124

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-18: SCMPR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-19: SCMPR5 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_F_EN bit in the REG_CTIMER_STCGF register.

11.2.2.10 SCMPR6 Register

Compare Register G
OFFSET: 0x00000128
INSTANCE 0 ADDRESS: 0x40008128

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so

that the math is precise. Reading this register shows the COUNTER value at which this interrupt will occur.

Table 11-20: SCMPR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-21: SCMPR6 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_G_EN bit in the REG_CTIMER_STCGF register.

11.2.2.11 SCMPR7 Register

Compare Register H
OFFSET: 0x0000012C
INSTANCE 0 ADDRESS: 0x4000812C

The VALUE in this bit field is used to compare against the VALUE in the COUNTER register. If the match criterion in the configuration register is met then a corresponding interrupt status bit is set. The match criterion is defined as COUNTER equal to COMPARE. To establish a desired value in this COMPARE register, write the number of ticks in the future to this register to indicate when to interrupt. The hardware does the addition to the COUNTER value in the STIMER clock domain so that the math is precise.

Table 11-22: SCMPR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-23: SCMPR7 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Compare this value to the value in the COUNTER register according to the match criterion, as selected in the COMPARE_H_EN bit in the REG_CTIMER_STCGF register.

11.2.2.12 SCAPT0 Register

Capture Register A
OFFSET: 0x000001E0
INSTANCE 0 ADDRESS: 0x400081E0
The STIMER capture Register A grabs the VALUE in the COUNTER register whenever capture condition (event) A is asserted. This register holds a time stamp for the event.

Table 11-24: SCAPT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-25: SCAPT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Whenever the event is detected, the value in the COUNTER is copied into this register and the corresponding interrupt status bit is set.

11.2.2.13 SCAPT1 Register

Capture Register B
OFFSET: 0x000001E4
INSTANCE 0 ADDRESS: 0x400081E4

The STIMER capture Register B grabs the VALUE in the COUNTER register whenever capture condition (event) B is asserted. This register holds a time stamp for the event.

Table 11-26: SCAPT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-27: SCAPT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Whenever the event is detected, the value in the COUNTER is copied into this register and the corresponding interrupt status bit is set.

11.2.2.14 SCAPT2 Register

Capture Register C
OFFSET: 0x000001E8
INSTANCE 0 ADDRESS: 0x400081E8

The STIMER capture Register C grabs the VALUE in the COUNTER register whenever capture condition (event) C is asserted. This register holds a time stamp for the event.

Table 11-28: SCAPT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-29: SCAPT2 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Whenever the event is detected, the value in the COUNTER is copied into this register and the corresponding interrupt status bit is set.

11.2.2.15 SCAPT3 Register

Capture Register D
OFFSET: 0x000001EC
INSTANCE 0 ADDRESS: 0x400081EC

The STIMER capture Register D grabs the VALUE in the COUNTER register whenever capture condition (event) D is asserted. This register holds a time stamp for the event.

Table 11-30: SCAPT3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-31: SCAPT3 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RO	Whenever the event is detected, the value in the COUNTER is copied into this register and the corresponding interrupt status bit is set.

11.2.2.16 SNVR0 Register

System Timer NVRAM_A Register
OFFSET: 0x000001F0
INSTANCE 0 ADDRESS: 0x400081F0

The NVRAM_A Register contains a portion of the stored epoch offset associated with the time in the COUNTER register. This register is only reset by POI not by HRESETn. Its contents are intended to survive all reset level except POI and full power cycles.

Table 11-32: SNVR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-33: SNVR0 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Value of the 32-bit counter as it ticks over.

11.2.2.17 SNVR1

System Timer NVRAM_B Register
OFFSET: 0x000001F4
INSTANCE 0 ADDRESS: 0x400081F4

The NVRAM_B Register contains a portion of the stored epoch offset associated with the time in the COUNTER register. This register is only reset by POI not by HRESETn. Its contents are intended to survive all reset level except POI and full power cycles.

Table 11-34: SNVR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-35: SNVR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Value of the 32-bit counter as it ticks over.

11.2.2.18 SNVR2

System Timer NVRAM_C Register
OFFSET: 0x000001F8
INSTANCE 0 ADDRESS: 0x400081F8

The NVRAM_C Register contains a portion of the stored epoch offset associated with the time in the COUNTER register. This register is only reset by POI not by HRESETn. Its contents are intended to survive all reset level except POI and full power cycles.

Table 11-36: SNVR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VALUE																															

Table 11-37: SNVR2 Register Bits

Bit	Name	Reset	RW	Description
31:0	VALUE	0x0	RW	Value of the 32-bit counter as it ticks over.

11.2.2.19 STMINTEN

STIMER Interrupt registers: Enable
OFFSET: 0x00000300
INSTANCE 0 ADDRESS: 0x40008300
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 11-38: STMINTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																			CAPTURED	CAPTUREC	CAPTUREB	CAPTUREA	OVERFLOW	COMPAREH	COMPAREG	COMPAREF	COMPAREE	COMPARED	COMPAREC	COMPAREB	COMPAREA

Table 11-39: STMINTEN Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED.
12	CAPTURED	0x0	RW	CAPTURE register D has grabbed the value in the counter CAPD_INT = 0x1 - Capture D interrupt status bit was set.
11	CAPTUREC	0x0	RW	CAPTURE register C has grabbed the value in the counter CAPC_INT = 0x1 - CAPTURE C interrupt status bit was set.
10	CAPTUREB	0x0	RW	CAPTURE register B has grabbed the value in the counter CAPB_INT = 0x1 - CAPTURE B interrupt status bit was set.

Table 11-39: STMINTEN Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
9	CAPTUREA	0x0	RW	CAPTURE register A has grabbed the value in the counter CAPA_INT = 0x1 - CAPTURE A interrupt status bit was set.
8	OVERFLOW	0x0	RW	COUNTER over flowed from 0xFFFFFFFF back to 0x00000000. OFLOW_INT = 0x1 - Overflow interrupt status bit was set.
7	COMPAREH	0x0	RW	COUNTER is greater than or equal to COMPARE register H. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
6	COMPAREG	0x0	RW	COUNTER is greater than or equal to COMPARE register G. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
5	COMPAREF	0x0	RW	COUNTER is greater than or equal to COMPARE register F. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
4	COMPAREE	0x0	RW	COUNTER is greater than or equal to COMPARE register E. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
3	COMPARED	0x0	RW	COUNTER is greater than or equal to COMPARE register D. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
2	COMPAREC	0x0	RW	COUNTER is greater than or equal to COMPARE register C. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
1	COMPAREB	0x0	RW	COUNTER is greater than or equal to COMPARE register B. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
0	COMPAREA	0x0	RW	COUNTER is greater than or equal to COMPARE register A. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.

11.2.2.20 STMINTSTAT

STIMER Interrupt registers: Status
OFFSET: 0x00000304
INSTANCE 0 ADDRESS: 0x40008304
Read bits from this register to discover the cause of a recent interrupt.

Table 11-40: STMINTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																			CAPTURED	CAPTUREC	CAPTUREB	CAPTUREA	OVERFLOW	COMPAREH	COMPAREG	COMPAREF	COMPAREE	COMPARED	COMPAREC	COMPAREB	COMPAREA

Table 11-41: STMINTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED.
12	CAPTURED	0x0	RW	CAPTURE register D has grabbed the value in the counter CAPD_INT = 0x1 - Capture D interrupt status bit was set.
11	CAPTUREC	0x0	RW	CAPTURE register C has grabbed the value in the counter CAPC_INT = 0x1 - CAPTURE C interrupt status bit was set.
10	CAPTUREB	0x0	RW	CAPTURE register B has grabbed the value in the counter CAPB_INT = 0x1 - CAPTURE B interrupt status bit was set.
9	CAPTUREA	0x0	RW	CAPTURE register A has grabbed the value in the counter CAPA_INT = 0x1 - CAPTURE A interrupt status bit was set.
8	OVERFLOW	0x0	RW	COUNTER over flowed from 0xFFFFFFFF back to 0x00000000. OFLOW_INT = 0x1 - Overflow interrupt status bit was set.
7	COMPAREH	0x0	RW	COUNTER is greater than or equal to COMPARE register H. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
6	COMPAREG	0x0	RW	COUNTER is greater than or equal to COMPARE register G. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
5	COMPAREF	0x0	RW	COUNTER is greater than or equal to COMPARE register F. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
4	COMPAREE	0x0	RW	COUNTER is greater than or equal to COMPARE register E. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
3	COMPARED	0x0	RW	COUNTER is greater than or equal to COMPARE register D. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
2	COMPAREC	0x0	RW	COUNTER is greater than or equal to COMPARE register C. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
1	COMPAREB	0x0	RW	COUNTER is greater than or equal to COMPARE register B. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
0	COMPAREA	0x0	RW	COUNTER is greater than or equal to COMPARE register A. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.

11.2.2.21 STMINTCLR

STIMER Interrupt registers: Clear
OFFSET: 0x00000308
INSTANCE 0 ADDRESS: 0x40008308
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 11-42: STMINTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																			CAPTURED	CAPTUREC	CAPTUREB	CAPTUREA	OVERFLOW	COMPAREH	COMPAREG	COMPAREF	COMPAREE	COMPARED	COMPAREC	COMPAREB	COMPAREA

Table 11-43: STMINTCLR Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED.
12	CAPTURED	0x0	RW	CAPTURE register D has grabbed the value in the counter CAPD_INT = 0x1 - Capture D interrupt status bit was set.
11	CAPTUREC	0x0	RW	CAPTURE register C has grabbed the value in the counter CAPC_INT = 0x1 - CAPTURE C interrupt status bit was set.
10	CAPTUREB	0x0	RW	CAPTURE register B has grabbed the value in the counter CAPB_INT = 0x1 - CAPTURE B interrupt status bit was set.
9	CAPTUREA	0x0	RW	CAPTURE register A has grabbed the value in the counter CAPA_INT = 0x1 - CAPTURE A interrupt status bit was set.
8	OVERFLOW	0x0	RW	COUNTER over flowed from 0xFFFFFFFF back to 0x00000000. OFLOW_INT = 0x1 - Overflow interrupt status bit was set.
7	COMPAREH	0x0	RW	COUNTER is greater than or equal to COMPARE register H. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
6	COMPAREG	0x0	RW	COUNTER is greater than or equal to COMPARE register G. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
5	COMPAREF	0x0	RW	COUNTER is greater than or equal to COMPARE register F. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
4	COMPAREE	0x0	RW	COUNTER is greater than or equal to COMPARE register E. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
3	COMPARED	0x0	RW	COUNTER is greater than or equal to COMPARE register D. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
2	COMPAREC	0x0	RW	COUNTER is greater than or equal to COMPARE register C. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
1	COMPAREB	0x0	RW	COUNTER is greater than or equal to COMPARE register B. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
0	COMPAREA	0x0	RW	COUNTER is greater than or equal to COMPARE register A. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.

11.2.2.22 STMINTSET

STIMER Interrupt registers: Set
OFFSET: 0x0000030C
INSTANCE 0 ADDRESS: 0x4000830C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 11-44: STMINTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																			CAPTURED	CAPTUREC	CAPTUREB	CAPTUREA	OVERFLOW	COMPAREH	COMPAREG	COMPAREF	COMPAREE	COMPARED	COMPAREC	COMPAREB	COMPAREA

Table 11-45: STMINTSET Register Bits

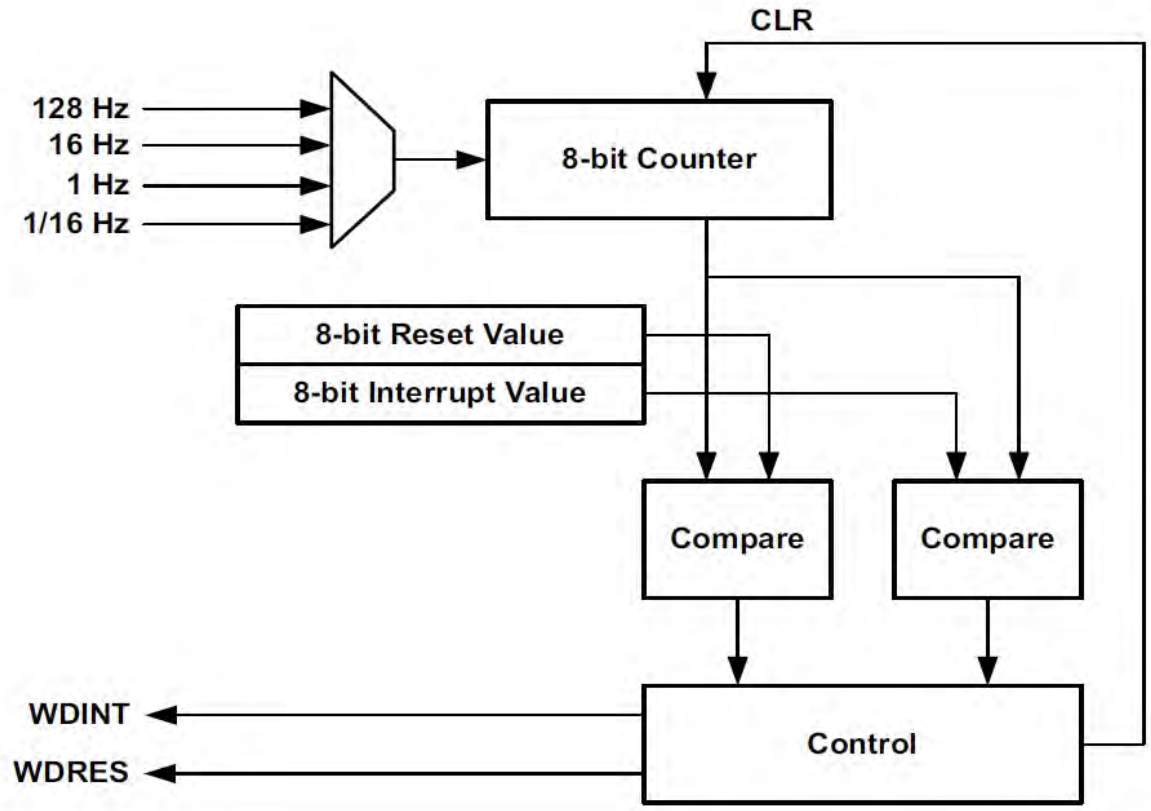
Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED.
12	CAPTURED	0x0	RW	CAPTURE register D has grabbed the value in the counter CAPD_INT = 0x1 - Capture D interrupt status bit was set.
11	CAPTUREC	0x0	RW	CAPTURE register C has grabbed the value in the counter CAPC_INT = 0x1 - CAPTURE C interrupt status bit was set.
10	CAPTUREB	0x0	RW	CAPTURE register B has grabbed the value in the counter CAPB_INT = 0x1 - CAPTURE B interrupt status bit was set.
9	CAPTUREA	0x0	RW	CAPTURE register A has grabbed the value in the counter CAPA_INT = 0x1 - CAPTURE A interrupt status bit was set.
8	OVERFLOW	0x0	RW	COUNTER over flowed from 0xFFFFFFFF back to 0x00000000. OFLOW_INT = 0x1 - Overflow interrupt status bit was set.
7	COMPAREH	0x0	RW	COUNTER is greater than or equal to COMPARE register H. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
6	COMPAREG	0x0	RW	COUNTER is greater than or equal to COMPARE register G. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
5	COMPAREF	0x0	RW	COUNTER is greater than or equal to COMPARE register F. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
4	COMPAREE	0x0	RW	COUNTER is greater than or equal to COMPARE register E. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
3	COMPARED	0x0	RW	COUNTER is greater than or equal to COMPARE register D. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
2	COMPAREC	0x0	RW	COUNTER is greater than or equal to COMPARE register C. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
1	COMPAREB	0x0	RW	COUNTER is greater than or equal to COMPARE register B. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.
0	COMPAREA	0x0	RW	COUNTER is greater than or equal to COMPARE register A. COMPARED = 0x1 - COUNTER greater than or equal to COMPARE register.

SECTION

12

Watchdog Timer Module

Figure 12-1: Block Diagram for the Watchdog Timer Module



12.1 Functional Overview

The Watchdog Timer (WDT), shown in Figure 12-1 on page 400, is used to insure that software is operational, by resetting the Apollo2 SoC if the WDT reaches a configurable value before being cleared by software. The WDT can be clocked by one of four selectable prescalers of the always active low-power LFRC clock, but is nominally clocked at 128 Hz. The WDT may be locked to ensure that software cannot disable its functionality, in which case the WDTCFG register cannot be updated. An interrupt can also be generated at a different counter value to implement an early warning function.

NOTE: The RESEN bit in the WDTCFG register must be set and the WDREN bit in the RSTCFG register must be set to enable a watchdog timer reset condition.

12.2 WDT Registers

Watchdog Timer
INSTANCE 0 BASE ADDRESS:0x40024000

12.2.1 Register Memory Map

Table 12-1: WDT Register Map

Address(es)	Registered Name	Description
0x40024000	CFG	Configuration Register
0x40024004	RSTRT	Restart the watchdog timer
0x40024008	LOCK	Locks the WDT
0x4002400C	COUNT	Current Counter Value for WDT
0x40024200	INTEN	WDT Interrupt register: Enable
0x40024204	INTSTAT	WDT Interrupt register: Status
0x40024208	INTCLR	WDT Interrupt register: Clear
0x4002420C	INTSET	WDT Interrupt register: Set

12.2.2 WDT Registers

12.2.2.1 CFG Register

Configuration Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40024000
Configuration Register

Table 12-2: CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					CLKSEL			INTVAL								RESVAL								RSVD					RESEN	INTEN	WDTEN

Table 12-3: CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bit field is reserved for future use.
26:24	CLKSEL	0x0	RW	Select the frequency for the WDT. All values not enumerated below are undefined. OFF = 0x0 - Low Power Mode. 128HZ = 0x1 - 128 Hz LFRC clock. 16HZ = 0x2 - 16 Hz LFRC clock. 1HZ = 0x3 - 1 Hz LFRC clock. 1_16HZ = 0x4 - 1/16th Hz LFRC clock.
23:16	INTVAL	0xff	RW	This bit field is the compare value for counter bits 7:0 to generate a watchdog interrupt.
15:8	RESVAL	0xff	RW	This bit field is the compare value for counter bits 7:0 to generate a watchdog reset.
7:3	RSVD	0x0	RO	This bit field is reserved for future use.
2	RESEN	0x0	RW	This bit field enables the WDT reset.
1	INTEN	0x0	RW	This bit field enables the WDT interrupt. Note This bit must be set before the interrupt status bit will reflect a watchdog timer expiration. The IER interrupt register must also be enabled for a WDT interrupt to be sent to the NVIC.
0	WDTEN	0x0	RW	This bit field enables the WDT.

12.2.2.2 RSTRT Register

Restart the watchdog timer
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40024004
Restart the watchdog timer

Table 12-4: RSTRT Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	
RSVD	RSTRT

Table 12-5: RSTRT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7:0	RSTRT	0x0	WO	Writing 0xB2 to WDTRSTRT restarts the watchdog timer. KEYVALUE = 0xB2 - This is the key value to write to WDTRSTRT to restart the WDT.

12.2.2.3 LOCK Register

Locks the WDT
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40024008
Locks the WDT

Table 12-6: LOCK Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	
RSVD	LOCK

Table 12-7: LOCK Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7:0	LOCK	0x0	WO	Writing 0x3A locks the watchdog timer. Once locked, the WDTCFG reg cannot be written and WDTEN is set. KEYVALUE = 0x3A - This is the key value to write to WDTLOCK to lock the WDT.

12.2.2.4 COUNT Register

Current Counter Value for WDT
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4002400C
Current Counter Value for WDT

Table 12-8: COUNT Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	
RSVD	COUNT

Table 12-9: COUNT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7:0	COUNT	0x0	RO	Read-Only current value of the WDT counter.

12.2.2.5 INTEN Register

WDT Interrupt register: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x40024200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 12-10: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															WDT

Table 12-11: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bit field is reserved for future use.
0	WDT	0x0	RW	Watchdog Timer Interrupt.

12.2.2.6 INTSTAT Register

WDT Interrupt register: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x40024204
Read bits from this register to discover the cause of a recent interrupt.

Table 12-12: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															WDT

Table 12-13: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bit field is reserved for future use.
0	WDT	0x0	RW	Watchdog Timer Interrupt.

12.2.2.7 INTCLR Register

WDT Interrupt register: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x40024208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 12-14: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															WDT

Table 12-15: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bit field is reserved for future use.
0	WDT	0x0	RW	Watchdog Timer Interrupt.

12.2.2.8 INTSET Register

WDT Interrupt register: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x4002420C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 12-16: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															WDT

Table 12-17: INTSET Register Bits

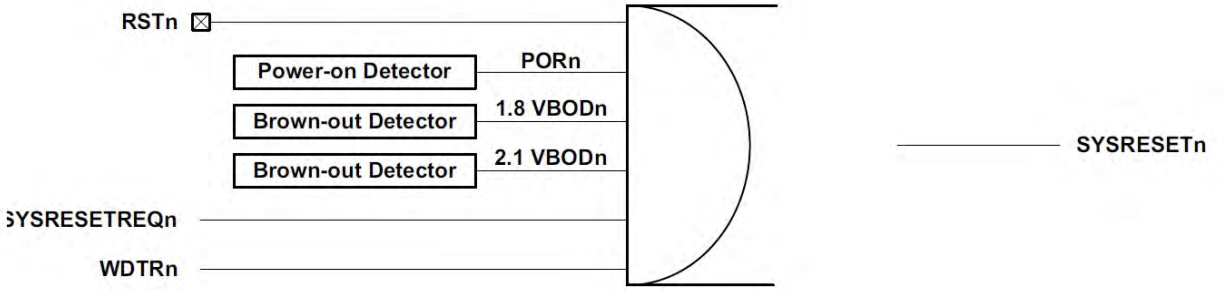
Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bit field is reserved for future use.
0	WDT	0x0	RW	Watchdog Timer Interrupt.

SECTION

13

Reset Generator Module

Figure 13-1: Block Diagram for the Reset Generator Module



13.1 Functional Overview

The Reset Generator Module (RSTGEN) monitors a variety of reset signals and asserts the active low system reset (SYSRESETn) accordingly. A reset causes the entire system to be re-initialized, and the cause of the most recent reset is indicated by the STAT register.

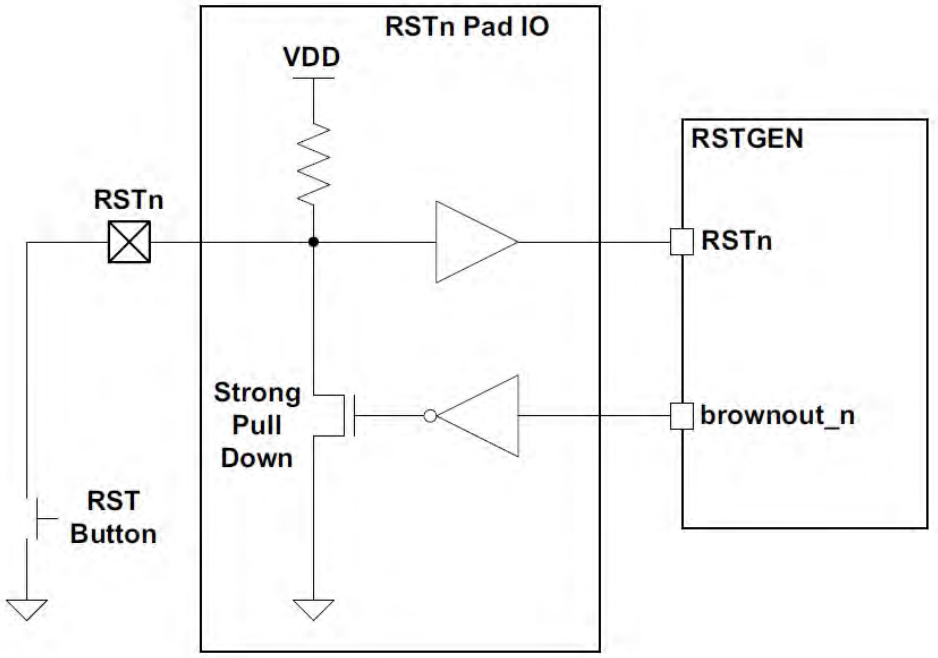
Reset sources are described in the subsequent sections and include:

- External reset pin (RSTn)
- Power-on event
- Brown-out event
- Software request (SYSRESETREQn)
- Watchdog expiration

13.2 External Reset Pin

The active-low RSTn pin can be used to generate a reset using an off-chip component (e.g., a pushbutton). An internal pull-up resistor in the RSTn pad enables optional floating of the RSTn pin, and a debounce circuit ensures that bounce glitches on RSTn does not cause unintentional resets. The RSTn pin is not maskable. An internal pull-down device will be active during a brownout event pulling the RSTn pin low. See Figure 13-2.

Figure 13-2: Block Diagram for Circuitry for Reset Pin



13.3 Power-on Event

An integrated power-on detector monitors the supply voltage and keeps SYSRESETn asserted while VDD is below the rising power-on voltage, V_{POR+} (1.755 V). When VDD rises above V_{POR} at initial power on, the reset module will initialize the low power analog circuitry followed by de-assertion of SYSRESETn, and normal operation proceeds. SYSRESETn is re-asserted as soon as VDD falls below the falling power-on voltage, V_{POR-} (1.755 V). The power-on reset signal, PORn, is not maskable.

13.4 Brown-out Event

An integrated brown-out detector monitors the supply voltage and causes an automatic and nonconfigurable reset when the voltage has fallen below the 1.755 V threshold. An optional reset or interrupt can be enabled when the brown-out detector indicates the supply voltage has fallen below the 2.1 V threshold. In the event the supply voltage falls below the 1.755 V threshold, or 2.1 V threshold any if enabled, the reset module will initiate a system reset, enabling the RSTn pull-down and driving the reset pin low. A 1.755 V or 2.1 V BOD reset will be reflected by the setting of the BORSTAT bit in the RSTGEN's STAT Register after reset.

In the event of a brownout detection, the following functionality is maintained until a power down detection occurs.

- All RTC registers retain state
- RTC and STIMER counters continue operation from 32 KHz XTAL or from LFRC (if below BODL). If clock sources stop oscillating at very low voltage, the RTC and STIMER will continue to maintain state.
- Clock configuration registers retain state

13.5 Software Reset

A reset may be generated via software using the Application Interrupt and Reset Control Register (AIRCR) defined in the Cortex-M4. For additional information on the AIRCR, see the ARM document titled “Cortex-M4 Devices Generic User Guide.” The software reset request is not maskable. A second source for the identical software reset functionality is made available through the SWPOR register in the RSTGEN peripheral module.

13.6 Software Power On Initialization

The SWPOI register enables the capability for software to perform a substantial reset that includes reloading the low power analog circuitry trim settings set in the flash information space. These values are not re-loaded from flash info space for Software Reset or External Reset events.

13.7 Watchdog Reset

The Watchdog Timer sub-module generates an interrupt if it has not been properly managed by software within a pre-defined time. The watchdog reset is maskable.

13.8 RSTGEN Registers

SoC Reset Generator
INSTANCE 0 BASE ADDRESS:0x40000000

13.8.1 Register Memory Map

Table 13-1: RSTGEN Register Map

Address(es)	Registered Name	Description
0x40000000	CFG	Configuration Register
0x40000004	SWPOI	Software POI Reset
0x40000008	SWPOR	Software POR Reset
0x4000000C	STAT	Status Register
0x40000010	CLRSTAT	Clear the status register
0x40000014	TPIU_RST	TPIU reset
0x40000200	INTEN	Reset Interrupt register: Enable
0x40000204	INTSTAT	Reset Interrupt register: Status
0x40000208	INTCLR	Reset Interrupt register: Clear
0x4000020C	INTSET	Reset Interrupt register: Set

13.8.1.1 CFG Register

Configuration Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x40000000
Configuration Register

Table 13-2: CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														WDREN	BODHREN

Table 13-3: CFG Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	WDREN	0x0	RW	Watchdog Timer Reset Enable. Note: The WDT module must also be configured for WDT reset.
0	BODHREN	0x0	RW	Brown out high (2.1v) reset enable.

13.8.1.2 SWPOI Register

Software POI Reset
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x40000004
Software POI Reset

Table 13-4: SWPOI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								SWPOIKEY							

Table 13-5: SWPOI Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:0	SWPOIKEY	0x0	WO	0x1B generates a software POI reset. KEYVALUE = 0x1B - Writing 0x1B key value generates a software POI reset.

13.8.1.3 SWPOR Register

Software POR Reset
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x40000008
Software POR Reset

Table 13-6: SWPOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								SWPORKEY							

Table 13-7: SWPOR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:0	SWPORKEY	0x0	WO	0xD4 generates a software POR reset. KEYVALUE = 0xD4 - Writing 0xD4 key value generates a software POR reset.

13.8.1.4 STAT Register

Status Register
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x4000000C
Status Register

Table 13-8: STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																									WDRSTAT	DBGRSTAT	POIRSTAT	SWRSTAT	BORSTAT	PORSTAT	EXRSTAT

Table 13-9: STAT Register Bits

Bit	Name	Reset	RW	Description
31:7	RSVD	0x0	RO	RESERVED.
6	WDRSTAT	0x0	RO	Reset was initiated by a Watchdog Timer Reset.
5	DBGRSTAT	0x0	RO	Reset was a initiated by Debugger Reset.
4	POIRSTAT	0x0	RO	Reset was a initiated by Software POI Reset.
3	SWRSTAT	0x0	RO	Reset was a initiated by SW POR or AIRCR Reset.
2	BORSTAT	0x0	RO	Reset was initiated by a Brown-Out Reset.
1	PORSTAT	0x0	RO	Reset was initiated by a Power-On Reset.
0	EXRSTAT	0x0	RO	Reset was initiated by an External Reset.

13.8.1.5 CLRSTAT Register

Clear the status register
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x40000010
Clear the status register

Table 13-10: CLRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															CLRSTAT

Table 13-11: CLRSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	CLRSTAT	0x0	WO	Writing a 1 to this bit clears all bits in the RST_STAT.

13.8.1.6 TPIU_RST Register

TPIU reset
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x40000014
TPIU reset

Table 13-12: TPIU_RST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															TPIURST

Table 13-13: TPIU_RST Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RW	RESERVED.
0	TPIURST	0x0	RW	Static reset for the TPIU. Write to '1' to assert reset to TPIU. Write to '0' to clear the reset.

13.8.1.7 INTEN Register

Reset Interrupt register: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x40000200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 13-14: INTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															BODH

Table 13-15: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	BODH	0x0	RW	Enables an interrupt that triggers when VCC is below BODH level.

13.8.1.8 INTSTAT Register

Reset Interrupt register: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x40000204
Read bits from this register to discover the cause of a recent interrupt.

Table 13-16: INTSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															BODH

Table 13-17: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	BODH	0x0	RW	Enables an interrupt that triggers when VCC is below BODH level.

13.8.1.9 INTCLR Register

Reset Interrupt register: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x40000208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 13-18: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															BODH

Table 13-19: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	BODH	0x0	RW	Enables an interrupt that triggers when VCC is below BODH level.

13.8.1.10 INTSET Register

Reset Interrupt register: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x4000020C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 13-20: INTSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															BODH

Table 13-21: INTSET Register Bits

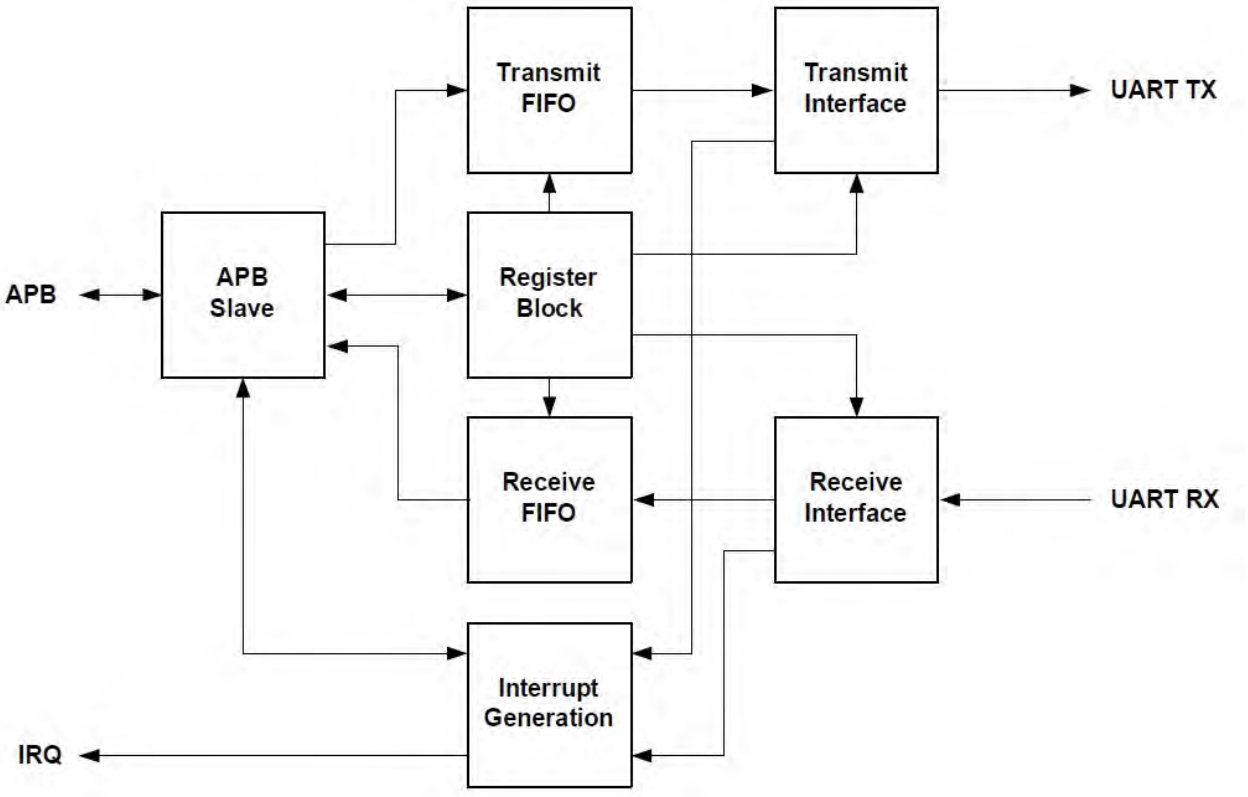
Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	BODH	0x0	RW	Enables an interrupt that triggers when VCC is below BODH level.

SECTION

14

UART Module

Figure 14-1: Block Diagram for the UART Module



14.1 Features

The UART Module includes the following key features:

- Operates independently, allowing the MCU to enter a low power sleep mode during communication
- 32 x 8 transmit FIFO and 32 x 12 receive FIFO to reduce SoC computational load
- Programmable baud rate generator capable of a maximum rate of 921,600 bits per second
- Fully programmable data size, parity, and stop bit length
- Programmable hardware flow control
- Support for full-duplex and half-duplex communication
- Loopback functionality for diagnostics and testing

14.2 Functional Overview

Shown in Figure 14-1 on page 414, the UART Module converts parallel data written through the APB Slave port into serial data which is transmitted to an external device. It also receives serial data from an external device and converts it to parallel data, which is then stored in a buffer until the CPU reads the data.

The UART Module includes a programmable baud rate generator which is capable of operating at a maximum of 921,600 bits per second. An interrupt generator will optionally send interrupts to the CPU core for transmit, receive and error events.

Internally, the UART Module maintains two FIFOs. The transmit FIFO is 1-byte wide with 32 locations. The receive FIFO is 12-bits wide with 32 locations. The extra four bits in the receive FIFO are used to capture any error status information that the SoC needs to analyze. Clocking to the UART serial logic is generated by a dedicated UARTCLK from the Clock Generator Module. The frequency of this clock is determined by the desired baud rate. For maximum baud rates, this clock would be clocked at the 24 MHz maximum as generated by the HFRC.

The major functional blocks of the UART are discussed briefly in the subsequent sections.

14.3 Enabling and Selecting the UART Clock

The UART module receives two clocks - UART_clk which is used to derive the UART serial clock and UART_hclk, which is the bus interface clock of the UART module. Unlike other Apollo2 SoC modules, the UART requires a bus clock whenever it is transmitting or receiving, so special controls are required when the UART is to transfer data while the Apollo2 SoC is in a sleep mode and its normal bus clocks are not operating.

UART_clk is selected in the UARTx_CR_CLKSEL field, with values from 24 MHz to 3 MHz plus a disabled value NOCLK, and is enabled by the UARTx_CR_CLKEN bit. If the UART is inactive, CLKSEL should be set to the NOCLK value (0) to minimize power, and the CLKEN bit should be 0. When the UART is active, the serial clock is created by the baud rate generator based on UART_clk. A higher UART_clk frequency can produce more precise serial clock frequencies, but will cause the UART to use more power. It is thus recommended that UART_clk be set to the minimum frequency which produces acceptable serial clocks.

When software is accessing the UART, UART_hclk must be equivalent to the Apollo2 SoC bus clock frequency of 48 MHz, but for transmit and receive purposes UART_hclk is only required to be at least as fast as UART_clk. It is thus possible to manage the frequency of UART_hclk to minimize power used by the UART. This is controlled by the CLK_GEN_UARTEN_UARTxEN fields, as defined in the table below.

Table 14-1: UART Bus Interface Clock Selection

UARTxEN	UART_hclk Function
0	Disable UART_hclk. Select this when the UART is inactive.
1	Force UART_hclk to 48MHz. This is not a recommended mode.
2	Force UART_hclk to match UART_clk. This mode may be used when the UART is actively transmitting or receiving, or is expected to receive a transmission. This minimizes power in the UART but does not allow software access to UART registers.
3	Automatic. In this mode, UART_hclk will be set to 48 MHz when Apollo2 SoC is awake and set to match UART_clk when Apollo2 SoC is in a sleep mode. This is a normal safe mode of operation.

In general, it is safe to leave the UARTxEN field at 3, which will minimize UART power in sleep modes but always allow UART register access. Power will be improved if UARTxEN is normally left at 2, and shifted to 3 whenever UART register access is required. Note that the UARTEN register is in the CLK_GEN module which always has bus access enabled.

14.4 Configuration

The UART Register Block in Figure 14-1 on page 414 may be set to configure the UART Module. The data width, number of stop bits, and parity may all be configured using the UART_LCRH register.

The baud rate is configured using the integer UART_IBRD and UART_FBRD registers. The correct values for UART_IBRD and UART_FBRD may be determined according to the following equation:

$F_{UART}/(16 \cdot BR) = IBRD + FBRD$

F_{UART} is the frequency of the UART clock. BR is the desired baud rate. IBRD is the integer portion of the baud rate divisor. FBRD is the fractional portion of the baud rate divisor.

14.5 Transmit FIFO and Receive FIFO

The transmit and receive FIFOs may both be accessed via the same 8-bit word in the UART_DR register. The transmit FIFO stores up to 32 8-bit words and can be written using writes to UART_DR. The receive FIFO stores up to 32 12-bit words and can be read using reads to UART_DR. Note that each 12-bit receive FIFO word includes an 8-bit data word and a 4-bit error status word.

14.6 UART Registers

Serial UART
INSTANCE 0 BASE ADDRESS:0x4001C000
INSTANCE 1 BASE ADDRESS:0x4001D000

14.6.1 Register Memory Map

Table 14-2: UART Register Map

Address(es)	Registered Name	Description
0x4001C000 0x4001D000	DR	UART Data Register
0x4001C004 0x4001D004	RSR	UART Status Register
0x4001C018 0x4001D018	FR	Flag Register
0x4001C020 0x4001D020	ILPR	IrDA Counter
0x4001C024 0x4001D024	IBRD	Integer Baud Rate Divisor
0x4001C028 0x4001D028	FBRD	Fractional Baud Rate Divisor
0x4001C02C 0x4001D02C	LCRH	Line Control High
0x4001C030 0x4001D030	CR	Control Register
0x4001C034 0x4001D034	IFLS	FIFO Interrupt Level Select
0x4001C038 0x4001D038	IER	Interrupt Enable
0x4001C03C 0x4001D03C	IES	Interrupt Status

Table 14-2: UART Register Map *(Continued)*

Address(es)	Registered Name	Description
0x4001C040 0x4001D040	MIS	Masked Interrupt Status
0x4001C044 0x4001D044	IEC	Interrupt Clear

14.6.2 UART Registers

14.6.2.1 DR Register

UART Data Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x4001C000
INSTANCE 1 ADDRESS: 0x4001D000
UART Data Register

Table 14-3: DR Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12	11	10	09	08	07 06 05 04 03 02 01 00
RSVD		OEDATA	BEDATA	PEDATA	FEDATA	DATA

Table 14-4: DR Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	This bit field is reserved for future use.
11	OEDATA	0x0	RO	This is the overrun error indicator. NOERR = 0x0 - No error on UART OEDATA, overrun error indicator. ERR = 0x1 - Error on UART OEDATA, overrun error indicator.
10	BEDATA	0x0	RO	This is the break error indicator. NOERR = 0x0 - No error on UART BEDATA, break error indicator. ERR = 0x1 - Error on UART BEDATA, break error indicator.
9	PEDATA	0x0	RO	This is the parity error indicator. NOERR = 0x0 - No error on UART PEDATA, parity error indicator. ERR = 0x1 - Error on UART PEDATA, parity error indicator.
8	FEDATA	0x0	RO	This is the framing error indicator. NOERR = 0x0 - No error on UART FEDATA, framing error indicator. ERR = 0x1 - Error on UART FEDATA, framing error indicator.
7:0	DATA	0x0	RW	This is the UART data port.

14.6.2.2 RSR Register

UART Status Register
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x4001C004

INSTANCE 1 ADDRESS: 0x4001D004
UART Status Register

Table 14-5: RSR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																												OESTAT	BESTAT	PESTAT	FESTAT

Table 14-6: RSR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	This bit field is reserved for future use.
3	OESTAT	0x0	RW	This is the overrun error indicator. NOERR = 0x0 - No error on UART OESTAT, overrun error indicator. ERR = 0x1 - Error on UART OESTAT, overrun error indicator.
2	BESTAT	0x0	RW	This is the break error indicator. NOERR = 0x0 - No error on UART BESTAT, break error indicator. ERR = 0x1 - Error on UART BESTAT, break error indicator.
1	PESTAT	0x0	RW	This is the parity error indicator. NOERR = 0x0 - No error on UART PESTAT, parity error indicator. ERR = 0x1 - Error on UART PESTAT, parity error indicator.
0	FESTAT	0x0	RW	This is the framing error indicator. NOERR = 0x0 - No error on UART FESTAT, framing error indicator. ERR = 0x1 - Error on UART FESTAT, framing error indicator.

14.6.2.3 FR Register

Flag Register
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x4001C018
INSTANCE 1 ADDRESS: 0x4001D018
Flag Register

Table 14-7: FR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																							TXBUSY	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS

Table 14-8: FR Register Bits

Bit	Name	Reset	RW	Description
31:9	RSVD	0x0	RO	This bit field is reserved for future use.
8	TXBUSY	0x0	RO	This bit holds the transmit BUSY indicator.
7	TXFE	0x0	RO	This bit holds the transmit FIFO empty indicator. XMTFIFO_EMPTY = 0x1 - Transmit fifo is empty.

Table 14-8: FR Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
6	RXFF	0x0	RO	This bit holds the receive FIFO full indicator. RCVFIFO_FULL = 0x1 - Receive fifo is full.
5	TXFF	0x0	RO	This bit holds the transmit FIFO full indicator. XMTFIFO_FULL = 0x1 - Transmit fifo is full.
4	RXFE	0x0	RO	This bit holds the receive FIFO empty indicator. RCVFIFO_EMPTY = 0x1 - Receive fifo is empty.
3	BUSY	0x0	RO	This bit holds the busy indicator. BUSY = 0x1 - UART busy indicator.
2	DCD	0x0	RO	This bit holds the data carrier detect indicator. DETECTED = 0x1 - Data carrier detect detected.
1	DSR	0x0	RO	This bit holds the data set ready indicator. READY = 0x1 - Data set ready.
0	CTS	0x0	RO	This bit holds the clear to send indicator. CLEARTOSEND = 0x1 - Clear to send is indicated.

14.6.2.4 ILPR Register

IrDA Counter
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x4001C020
INSTANCE 1 ADDRESS: 0x4001D020
IrDA Counter

Table 14-9: ILPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								ILPDVSR							

Table 14-10: ILPR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7:0	ILPDVSR	0x0	RW	These bits hold the IrDA counter divisor.

14.6.2.5 IBRD Register

Integer Baud Rate Divisor
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x4001C024
INSTANCE 1 ADDRESS: 0x4001D024
Integer Baud Rate Divisor

Table 14-11: IBRD Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
RSVD	DIVINT

Table 14-12: IBRD Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	This bit field is reserved for future use.
15:0	DIVINT	0x0	RW	These bits hold the baud integer divisor.

14.6.2.6 FBRD Register

Fractional Baud Rate Divisor
OFFSET: 0x00000028
INSTANCE 0 ADDRESS: 0x4001C028
INSTANCE 1 ADDRESS: 0x4001D028
Fractional Baud Rate Divisor

Table 14-13: FBRD Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06	05 04 03 02 01 00
RSVD	DIVFRAC

Table 14-14: FBRD Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	This bit field is reserved for future use.
5:0	DIVFRAC	0x0	RW	These bits hold the baud fractional divisor.

14.6.2.7 LCRH Register

Line Control High
OFFSET: 0x0000002C
INSTANCE 0 ADDRESS: 0x4001C02C
INSTANCE 1 ADDRESS: 0x4001D02C
Line Control High

Table 14-15: LCRH Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08	07	06	05	04	03	02	01	00
RSVD	SPS	WLEN		FEN	STP2	EPS	PEN	BRK

Table 14-16: LCRH Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bit field is reserved for future use.
7	SPS	0x0	RW	This bit holds the stick parity select.
6:5	WLEN	0x0	RW	These bits hold the write length. WLEN Data Bits 00 5 01 6 10 7 11 8
4	FEN	0x0	RW	This bit holds the FIFO enable.
3	STP2	0x0	RW	This bit holds the two stop bits select.
2	EPS	0x0	RW	This bit holds the even parity select.
1	PEN	0x0	RW	This bit holds the parity enable.
0	BRK	0x0	RW	This bit holds the break set.

14.6.2.8 CR Register

Control Register
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x4001C030
INSTANCE 1 ADDRESS: 0x4001D030
Control Register

Table 14-17: CR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																CTSEN	RTSEN	OUT2	OUT1	RTS	DTR	RXE	TXE	LBE	CLKSEL			CLKEN	SIRLP	SIREN	UARTEN

Table 14-18: CR Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	This bit field is reserved for future use.
15	CTSEN	0x0	RW	This bit enables CTS hardware flow control.
14	RTSEN	0x0	RW	This bit enables RTS hardware flow control.
13	OUT2	0x0	RW	This bit holds modem Out2.
12	OUT1	0x0	RW	This bit holds modem Out1.
11	RTS	0x0	RW	This bit enables request to send.
10	DTR	0x0	RW	This bit enables data transmit ready.
9	RXE	0x1	RW	This bit is the receive enable.
8	TXE	0x1	RW	This bit is the transmit enable.
7	LBE	0x0	RW	This bit is the loopback enable.

Table 14-18: CR Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
6:4	CLKSEL	0x0	RW	This bit field is the UART clock select. NOCLK = 0x0 - No UART clock. This is the low power default. 24MHZ = 0x1 - 24 MHz clock. 12MHZ = 0x2 - 12 MHz clock. 6MHZ = 0x3 - 6 MHz clock. 3MHZ = 0x4 - 3 MHz clock. RSVD5 = 0x5 - Reserved. RSVD6 = 0x6 - Reserved. RSVD7 = 0x7 - Reserved.
3	CLKEN	0x0	RW	This bit is the UART clock enable.
2	SIRLP	0x0	RW	This bit is the SIR low power select.
1	SIREN	0x0	RW	This bit is the SIR ENDEC enable.
0	UARTEN	0x0	RW	This bit is the UART enable.

14.6.2.9 IFLS Register

FIFO Interrupt Level Select
OFFSET: 0x00000034
INSTANCE 0 ADDRESS: 0x4001C034
INSTANCE 1 ADDRESS: 0x4001D034
FIFO Interrupt Level Select

Table 14-19: IFLS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										RXIFLSEL			TXIFLSEL		

Table 14-20: IFLS Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	This bit field is reserved for future use.
5:3	RXIFLSEL	0x2	RW	These bits hold the receive FIFO interrupt level.
2:0	TXIFLSEL	0x2	RW	These bits hold the transmit FIFO interrupt level.

14.6.2.10 IER Register

Interrupt Enable
OFFSET: 0x00000038
INSTANCE 0 ADDRESS: 0x4001C038
INSTANCE 1 ADDRESS: 0x4001D038
Interrupt Enable

Table 14-21: IER Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00														
RSVD	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSRMIM	DCDMIM	CTSMIM	TXCMPMIM			

Table 14-22: IER Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bit field is reserved for future use.
10	OEIM	0x0	RW	This bit holds the overflow interrupt enable.
9	BEIM	0x0	RW	This bit holds the break error interrupt enable.
8	PEIM	0x0	RW	This bit holds the parity error interrupt enable.
7	FEIM	0x0	RW	This bit holds the framing error interrupt enable.
6	RTIM	0x0	RW	This bit holds the receive timeout interrupt enable.
5	TXIM	0x0	RW	This bit holds the transmit interrupt enable.
4	RXIM	0x0	RW	This bit holds the receive interrupt enable.
3	DSRMIM	0x0	RW	This bit holds the modem DSR interrupt enable.
2	DCDMIM	0x0	RW	This bit holds the modem DCD interrupt enable.
1	CTSMIM	0x0	RW	This bit holds the modem CTS interrupt enable.
0	TXCMPMIM	0x0	RW	This bit holds the modem TXCMP interrupt enable.

14.6.2.11 IES Register

Interrupt Status
OFFSET: 0x0000003C
INSTANCE 0 ADDRESS: 0x4001C03C
INSTANCE 1 ADDRESS: 0x4001D03C
Interrupt Status

Table 14-23: IES Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00														
RSVD	OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	DSRMRIS	DCDMRIS	CTSMRIS	TXCMPMRIS			

Table 14-24: IES Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bit field is reserved for future use.
10	OERIS	0x0	RO	This bit holds the overflow interrupt status.
9	BERIS	0x0	RO	This bit holds the break error interrupt status.
8	PERIS	0x0	RO	This bit holds the parity error interrupt status.

Table 14-24: IES Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
7	FERIS	0x0	RO	This bit holds the framing error interrupt status.
6	RTRIS	0x0	RO	This bit holds the receive timeout interrupt status.
5	TXRIS	0x0	RO	This bit holds the transmit interrupt status.
4	RXRIS	0x0	RO	This bit holds the receive interrupt status.
3	DSRMRIS	0x0	RO	This bit holds the modem DSR interrupt status.
2	DCDMRIS	0x0	RO	This bit holds the modem DCD interrupt status.
1	CTSMRIS	0x0	RO	This bit holds the modem CTS interrupt status.
0	TXCMPMRIS	0x0	RO	This bit holds the modem TXCMP interrupt status.

14.6.2.12 MIS Register

Masked Interrupt Status
OFFSET: 0x00000040
INSTANCE 0 ADDRESS: 0x4001C040
INSTANCE 1 ADDRESS: 0x4001D040
Masked Interrupt Status

Table 14-25: MIS Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11	10	09	08	07	06	05	04	03	02	01	00
RSVD	OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	DSRMMIS	DCDMMIS	CTSMMIS	TXCMPMMIS

Table 14-26: MIS Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bit field is reserved for future use.
10	OEMIS	0x0	RO	This bit holds the overflow interrupt status masked.
9	BEMIS	0x0	RO	This bit holds the break error interrupt status masked.
8	PEMIS	0x0	RO	This bit holds the parity error interrupt status masked.
7	FEMIS	0x0	RO	This bit holds the framing error interrupt status masked.
6	RTMIS	0x0	RO	This bit holds the receive timeout interrupt status masked.
5	TXMIS	0x0	RO	This bit holds the transmit interrupt status masked.
4	RXMIS	0x0	RO	This bit holds the receive interrupt status masked.
3	DSRMMIS	0x0	RO	This bit holds the modem DSR interrupt status masked.
2	DCDMMIS	0x0	RO	This bit holds the modem DCD interrupt status masked.
1	CTSMMIS	0x0	RO	This bit holds the modem CTS interrupt status masked.
0	TXCMPMMIS	0x0	RO	This bit holds the modem TXCMP interrupt status masked.

14.6.2.13 IEC Register

Interrupt Clear
OFFSET: 0x00000044
INSTANCE 0 ADDRESS: 0x4001C044
INSTANCE 1 ADDRESS: 0x4001D044
Interrupt Clear

Table 14-27: IEC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																					OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRMIC	DCDMIC	CTSMIC	TXCMPMIC

Table 14-28: IEC Register Bits

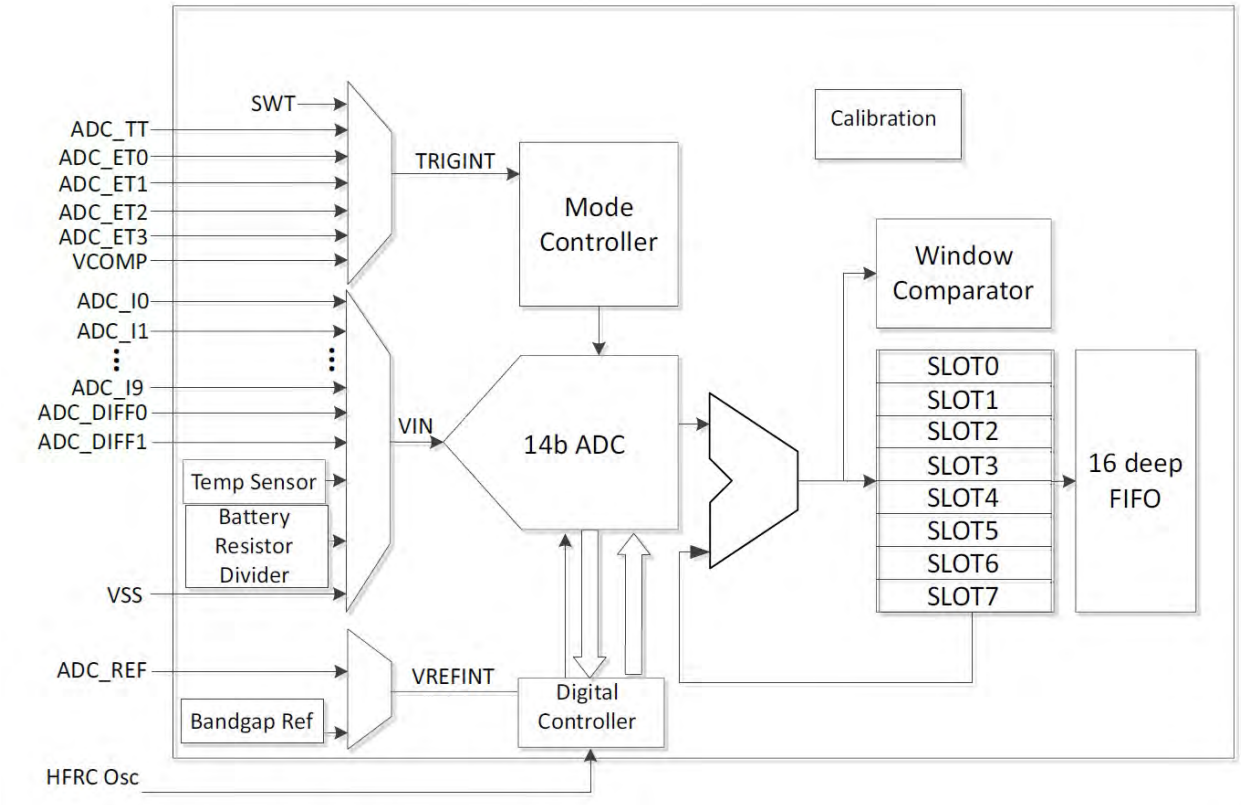
Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bit field is reserved for future use.
10	OEIC	0x0	WO	This bit holds the overflow interrupt clear.
9	BEIC	0x0	WO	This bit holds the break error interrupt clear.
8	PEIC	0x0	WO	This bit holds the parity error interrupt clear.
7	FEIC	0x0	WO	This bit holds the framing error interrupt clear.
6	RTIC	0x0	WO	This bit holds the receive timeout interrupt clear.
5	TXIC	0x0	WO	This bit holds the transmit interrupt clear.
4	RXIC	0x0	WO	This bit holds the receive interrupt clear.
3	DSRMIC	0x0	WO	This bit holds the modem DSR interrupt clear.
2	DCDMIC	0x0	WO	This bit holds the modem DCD interrupt clear.
1	CTSMIC	0x0	WO	This bit holds the modem CTS interrupt clear.
0	TXCMPMIC	0x0	WO	This bit holds the modem TXCMP interrupt clear.

SECTION

15

ADC and Temperature Sensor Module

Figure 15-1: Block Diagram for ADC and Temperature Sensor



15.1 Features

The Analog-to-Digital Converter (ADC) and Temperature Sensor Module includes a single-ended 14 bit multi-channel Successive Approximation Register (SAR) ADC as shown in Figure 15-1 on page 427.

Key features include:

- 14 user-selectable channels with sources including:
  - External pins
  - 10 single ended external pins
  - 2 differential pairs
  - Internal voltage (VSS)
  - Voltage divider (battery)
  - Temperature sensor
- Configurable automatic low power control between scans
- Optional Battery load enable for voltage divider measurement
- Configurable for 14 / 12 / 10 / 8 bit ADC Precision Modes
- User-selectable on-chip and off-chip reference voltages
- Single shot, repeating single shot, scan, and repeating scan modes
- User-selectable clock source for variable sampling rates
- Automatically accumulate and scale module for hardware averaging of samples
- A 16-entry FIFO for storing measurement results and maximizing SoC sleep time
- Window comparator for monitoring voltages excursions into or out of user-selectable thresholds
- Up to 2.67 MS/s effective continuous, multi-slot sampling rate
- Interrupts for FIFO full, FIFO almost full, Scan Complete, Conversion Complete, Window Incursion Window Excursion

15.2 Functional Overview

The Apollo2 SoC integrates a sophisticated 14-bit successive approximation Analog to Digital Converter (ADC) block for sensing both internal and external voltages. The block provides eight separately managed conversion requests, called slots. The result of each conversion requests is delivered to a 16 deep FIFO. Firmware can utilize various interrupt notifications to determine when to collect the sampled data from the FIFO. This block is extremely effective at automatically managing its power states and its clock sources.

15.2.1 Clock Source and Dividers

The ADC runs off of the HFRC clock source. When the ADC block is enabled and has an active scan in progress, it requests a clock source. There is an automatic hardware hand shake between the clock generator and the ADC. If the ADC is the only

block requesting an HFRC based clock, then the HFRC will be automatically started. The ADC can be configured to completely power down the HFRC between scans if the startup latency is acceptable or it can leave the HFRC powered on between scans if the application requires low latency between successive conversions. The ADC supports 2 HFRC clock frequency modes: 24MHz and 48MHz HFRC. 48MHz mode is the default mode of operation.

15.2.2 15 Channel Analog Mux

As shown in Figure 15-1 on page 427, the ADC block contains a 15 channel analog multiplexer on the input port to the analog to digital converter. Twelve (12) of the GPIO pins on the Apollo2 SoC can be selected as analog inputs to the ADC through a combination of settings in the PAD configuration registers in the GPIO block and settings in the configuration registers described below.

The analog mux channels are connected as follows:

1. ADC_EXT0 external GPIO pin connection.
2. ADC_EXT1 external GPIO pin connection.
3. ADC_EXT2 external GPIO pin connection.
4. ADC_EXT3 external GPIO pin connection.
5. ADC_EXT4 external GPIO pin connection.
6. ADC_EXT5 external GPIO pin connection.
7. ADC_EXT6 external GPIO pin connection.
8. ADC_EXT7 external GPIO pin connection.
9. ADC_EXT8 external GPIO pin connection.
10. ADC_EXT9 external GPIO pin connection.
11. ADC_EXT_DIFF0P external GPIO connection (muxed with EXT8)
    ADC_EXT_DIFF0N external GPIO connection (muxed with EXT9)
12. ADC_EXTDIFF1P external GPIO pin connection.
    ADC_EXT DIFF1N external GPIO pin connection.
13. ADC_TEMP internal temperature sensor.
14. ADC_DIV3 internal voltage divide by 3 connection to the input power rail.
15. ADC_VSS internal ground connection.

EXT8-9 can be configured as a differential pair providing an additional differential pair or up to 2 single ended inputs from GPIO.

Refer to the detailed register information below for the exact coding of the channel selection bit field. Also the use of the voltage divider and switchable load resistor are detailed below.

15.2.3 Triggering and Trigger Sources

The ADC block can be initially triggered from one of six sources. Once triggered, it can be repetitively triggered from counter/timer number three (3). Four of the GPIO pins on the Apollo2 SoC can be selected as trigger inputs to the ADC through a combination of settings in the PAD configuration registers in the GPIO block and settings in SLOT configuration registers described below. In addition, there is a software trigger and a vcomp trigger source. The trigger sources are as follows:

0. ADC_EXT0 (TRIG0) external GPIO pin connection.
1. ADC_EXT1 (TRIG1) external GPIO pin connection.
2. ADC_EXT2 (TRIG2) external GPIO pin connection.
3. ADC_EXT3 (TRIG3) external GPIO pin connection.
4. VCOMP Voltage Comparator trigger.
5. <Reserved>
6. <Reserved>
7. ADC_SWT software trigger.

Refer to the ADC Configuration Register in the detailed register information section below. The initial trigger source is selected in the TRIGSEL field, as shown below. In addition, one can select a trigger polarity in this register applicable for any of the trigger sources except the software trigger. A number of GPIO pin trigger sources are provided to allow pin configuration flexibility at the system definition and board layout phases of development.

The software trigger is effected by writing 0x37 to the software trigger register in the ADC block. Note that writing 0x37 to the software trigger register will initiate a scan regardless of which trigger source is selected. However, a hardware trigger source will not initiate a scan if the software trigger has been selected.

When the ADC is configured for repeat mode, the initial trigger must be initiated by a software trigger and subsequent scans will be initiated at a repeating rate set by the counter/timer3 configuration. The discussion of the use of counter/timer three as a source for repetitive triggering is deferred until later in this chapter.

NOTE: A trigger event applies to all enabled slots as a whole. Individual slots can not be separately triggered.

15.3 Voltage Reference Sources

The Apollo2 SoC ADC supports one of two reference sources each with two different voltage options to be used for the analog to digital conversion step:

- Internal 2.0V reference source
- Internal 1.5V reference source
- External 2.0V reference source
- External 1.5V reference source

15.3.1 Eight Automatically Managed Conversion Slots

The ADC block contains eight conversion slot control registers, one for each of the eight slots. These can be thought of as time slots in the conversion process. When a slot is enabled, it participates in a conversion cycle. The ADC's mode controller cycles through up to eight time slots each time it is triggered. For each slot that is enabled, a conversion cycle is performed based on the settings in the slot configuration register for that slot. Slots are enabled when the LSB of the slot configuration is set to one. See Table 15-1.

Table 15-1: One SLOT Configuration Register

31 30 29 28 27	26 25 24	23 22 21 20 19 18 17 16 15 14 13 12	11 10 09 08	07 06 05 04 03 02	01	00
Reserved	# Samples to Accum.	Reserved	CHANNEL SELECT	Reserved	WINDOW_COMP	SLOT_ENABLE

The window comparator enable will be discussed in a subsequent section, below. See *Section 15.3.4 Window Comparator on page 436*. The number of samples to accumulate will also be explained in a subsequent section. See *Section 15.3.2 Automatic Sample Accumulation and Scaling on page 432*.

As described above, the channel select bit field specifies which one of the analog multiplexer channels will be used for the conversions requested for an individual slot. See "*Section 15.2.2 15 Channel Analog Mux on page 429*.

Each of the eight conversion slots can independently specify:

- Analog Multiplexer Channel Selection
- Participation in Window Comparisons
- Automatic Sample Accumulation

15.3.2 Automatic Sample Accumulation and Scaling

The ADC block offers a facility for the automatic accumulation of samples without requiring core involvement. Thus up to 128 samples per slot can be accumulated without waking the core. This facilitates averaging algorithms to smooth out the data samples. Each slot can request from 1 to 128 samples to be accumulated before producing a result in the FIFO.

NOTE: Each slot can independently specify how many samples to accumulate so results can enter the FIFO from different slots at different rates.

All slots write their accumulated results to the FIFO in exactly the same format regardless of how many samples were accumulated to produce the results. Table 15-1 on page 431 shows the format that is used by all conversions. This is a scaled integer format with a 6-bit fractional part. The precision mode for each determines the format for the FIFO data. 14-bit, 12-bit, 10-bit and 8-bit precision modes respectively correspond to 14.6, 12.6, 10.6 and 8.6 formats.

NOTE: If the accumulation control for a slot is set for one sample with 14-bit precision, then the 14-bit value coming from the ADC will be inserted into bits 6 through 19 in this format and the lower 6 bits are zero'd. If the accumulation control for a slot is set for two samples with 8-bit precision, then the 8-bit average integer value will be placed in bits 6 through 13, the 1 bit fractional number is placed in bit 5 and the lower 5 fractional bits are zero'd.

Table 15-2: ADC Sample Format

19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
14-bit Integer														6-bit Fraction					

Each slot contains a 21-bit accumulator as shown in Table 709, “Per Slot Sample Accumulator,” on page 471. When the ADC is triggered for the last sample of an accumulation, the accumulator is cleared and the FIFO will be written with the final average value. When each active slot obtains a sample from the ADC, it is added to the value in its accumulator.

If a slot is set to accumulate 128 samples per result then the accumulator could reach a maximum value of:

$128*(2^{14}-1) = 128*16383 = 2097024 = 2^{21} - 128$, hence the 21 bit accumulator.

Table 15-3: Per Slot Sample Accumulator

20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Accumulator																				

Table 15-3 on page 433 shows the maximum possible accumulated values. Note that 64 sample accumulation produces a result that is exactly correct or the 14.6 format results so it is copied unscaled in to the FIFO.

Furthermore, note that 128 sample accumulation can produce a result that is too large for the 14.6 format since it may result in 7 bits of valid fractional data. All of the remaining sample accumulation settings must have their results left shifted to produce the desired 14.6 format.

Finally, note that for the 128 sample accumulation case, the LSB of the accumulator is discarded when the results are written to the FIFO.

Most importantly, note that for the 1 sample accumulation case, the 14-bit converter value is shifted left by six to produce the 14.6 format to write into the FIFO.

Table 15-4: Accumulator Scaling

# Samples	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
128	14.6																				0
64	X	14.6																			
32	X	X	14.5																		
16	X	X	X	14.4																	
8	X	X	X	X	14.3																
4	X	X	X	X	X	14.2															
2	X	X	X	X	X	X	14.1														
1	X	X	X	X	X	X	X	14													

15.3.3 Sixteen Entry Result FIFO

All results written to the FIFO have exactly the same format as shown in Table 15-5 on page 434. The properly scaled accumulation results are written the lower half word in the aforementioned 14.6 format. Since each slot can produce results at a different rate, the slot number generating the result is also written to the FIFO along with the total valid entry count within the FIFO.

Table 15-5: FIFO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSV	Slot Number			FIFO Count								FIFO DATA																			

Table 15-6: 14-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
128	14.6																			
64	14.6																			
32	14.5																			X
16	14.4																		X	X
8	14.3																	X	X	X
4	14.2																X	X	X	X
2	14.1															X	X	X	X	X
1	14														X	X	X	X	X	X

Table 15-7: 12-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
128	0	0	12.6																	
64	0	0	12.6																	
32	0	0	12.5																	X
16	0	0	12.4																X	X
8	0	0	12.3															X	X	X
4	0	0	12.2														X	X	X	X
2	0	0	12.1													X	X	X	X	X
1	0	0	12												X	X	X	X	X	X

Table 15-8: 10-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
128	0	0	0	0	10										6					
64	0	0	0	0	10										6					
32	0	0	0	0	10										5					X
16	0	0	0	0	10										4				X	X
8	0	0	0	0	10										3			X	X	X
4	0	0	0	0	10										2		X	X	X	X

Table 15-8: 10-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
2	0	0	0	0	10										1	X	X	X	X	X
1	0	0	0	0	10										X	X	X	X	X	X

Table 15-9: 8-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
128	0	0	0	0	0	0	8.6													
64	0	0	0	0	0	0	8.6													
32	0	0	0	0	0	0	8.5													X
16	0	0	0	0	0	0	8.4												X	X
8	0	0	0	0	0	0	8.3											X	X	X
4	0	0	0	0	0	0	8.2										X	X	X	X
2	0	0	0	0	0	0	8.1									X	X	X	X	X
1	0	0	0	0	0	0	8								X	X	X	X	X	X

Software accesses the contents of the FIFO through the ADCFIFO register. This register will be written by the ADC digital controller simultaneous with the conversion complete interrupt (if enabled) after accumulating the number of samples to average configured for the slot. The ADCFIFO register contains the earliest written data, the number of valid entries within the FIFO and the slot number associated with the FIFO data. Thus the interrupt handler servicing ADC interrupts can easily distribute results to different RTOS tasks by simply looking up the target task using the slot number from the FIFO register.

Three other features greatly simplify the task faced by firmware developers of interrupt service routines for the ADC block:

1. The FIFO count bit field is not really stored in the FIFO. Instead it is a live count of the number of valid entries currently residing in the FIFO. If the interrupt service routine was entered because of a conversion then this value will be at least one. When the interrupts routine is entered it can pull successive sample values from the FIFO until this bit field goes to zero. Thus avoiding wasteful re-entry of the interrupt service routine. Note that no further I/O bus read is required to determine the FIFO depth.
2. This FIFO has no read side effects. This is important to firmware for a number of reasons. One important result is that the FIFO register can be freely read repetitively by a debugger without affecting the state of the FIFO. In order to pop this FIFO and look at the next result, if any, one simply writes any value to this register. Any time the FIFO is read, then the compiler has gone to the trouble of generating an address for the read. To pop the FIFO, one simply writes to that

same address with any value. This give firmware a positive handshake mechanism to control exactly when the FIFO pops.

3. When a conversion completes resulting in hardware populating the 12th valid FIFO entry, the FIFOOVR1 (FIFO 75% full) interrupt status bit will be set. When a conversion completes resulting in hardware populating the 8th valid FIFO entry, the FIFOOVR2 interrupt status bit will be set. In a FIFO full condition with 16 valid entries, the ADC will not overwrite existing valid FIFO contents. Before subsequent conversions will populate the FIFO with conversion data, software must free an open FIFO entry by writing to the FIFO Register or by resetting the ADC by disabling and enabling the ADC using the ADC_CFG register.

15.3.4 Window Comparator

A window comparator is provided which can generate an interrupt whenever a sample is determined to be inside the window limits or outside the window limits. These are two separate interrupts with separate interrupt enables. Thus one can request an interrupt any time a specified slot makes an excursion outside the window comparator limits.

The window comparison function has an option for comparing the contents of the limits registers directly with the FIFO data (default) or for scaling the limits register depending on the precision mode selected for the slots.

Firmware has to participate in the determination of whether an actual excursion occurred. The window comparator interrupts set their corresponding interrupt status bits continuously whenever the inside or outside condition is true. Thus if one enables and receives an "excursion" interrupt then the status bit cannot be usefully cleared while the ADC slot is sampling values outside the limits. That is, if one receives an excursion interrupt and clears the status bit, it will immediately set again if the next ADC sample is still outside the limits. Thus firmware should reconfigure the interrupt enables upon receiving an excursion interrupt so that the next interrupt will occur when an ADC sample ultimately goes back inside the window limits. Firmware may also want to change the windows comparator limit at that time to utilize a little hysteresis in these window comparator decisions.

Table 15-10: Window Comparator Lower Limit Register

19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Lower Limit																			

Table 15-11: Window Comparator Upper Limit Register

19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Upper Limit																			

The determination of whether a sample is inside or outside of the window limits is made by comparing the data format of the slot result written to the FIFO with the 20 bit window limits. An ADC sample is inside if the following relation is true:

14.6 Lower Limit <= ADC SAMPLE <= 14.6 Upper Limit

Thus setting both limits to the same value, say 700.0 (0x2BC<<6 = 0xAF00), will only produce an inside interrupt when the ADC sample is exactly 700.0 (0xAF00). Furthermore, note that if the lower limit is set to zero (0x00000) and the upper limit is set to 0xFFFFF then all accumulated results from the ADC will be inside the window limits and no excursion interrupts can ever by generated. In fact, in this case, the incursion interrupt status bit will be set for every sample from any active slot with its window comparator bit enabled. If the incursion interrupt is enabled then an interrupt will be generated for every such sample written to the FIFO.

The window comparator limits are a shared resource and apply to all active slots which have their window comparator bits enabled. If window limits are enabled for multiple enabled slots with different precision modes, the window comparison function can be configured to automatically scale the 14.6 upper and lower limits value to match the corresponding precision mode format for the enabled slots through the ADCSCWLIM register.

15.4 Operating Modes and the Mode Controller

The mode controller of Figure 15-1 on page 427 is a sophisticated state machine that manages not only the time slot conversions but also the power state of the ADC analog components and the hand shake with the clock generator to start the HFRC clock source if required. Thus once the various control registers are initialized, the core can go to sleep and only wake up when there are valid samples in the FIFO for the interrupt service routine to distribute. Firmware does not have to keep track of which block is using the HFRC clock source since the devices in conjunction with the clock generator manage this automatically. The ADC block's mode controller participates in this clock management protocol.

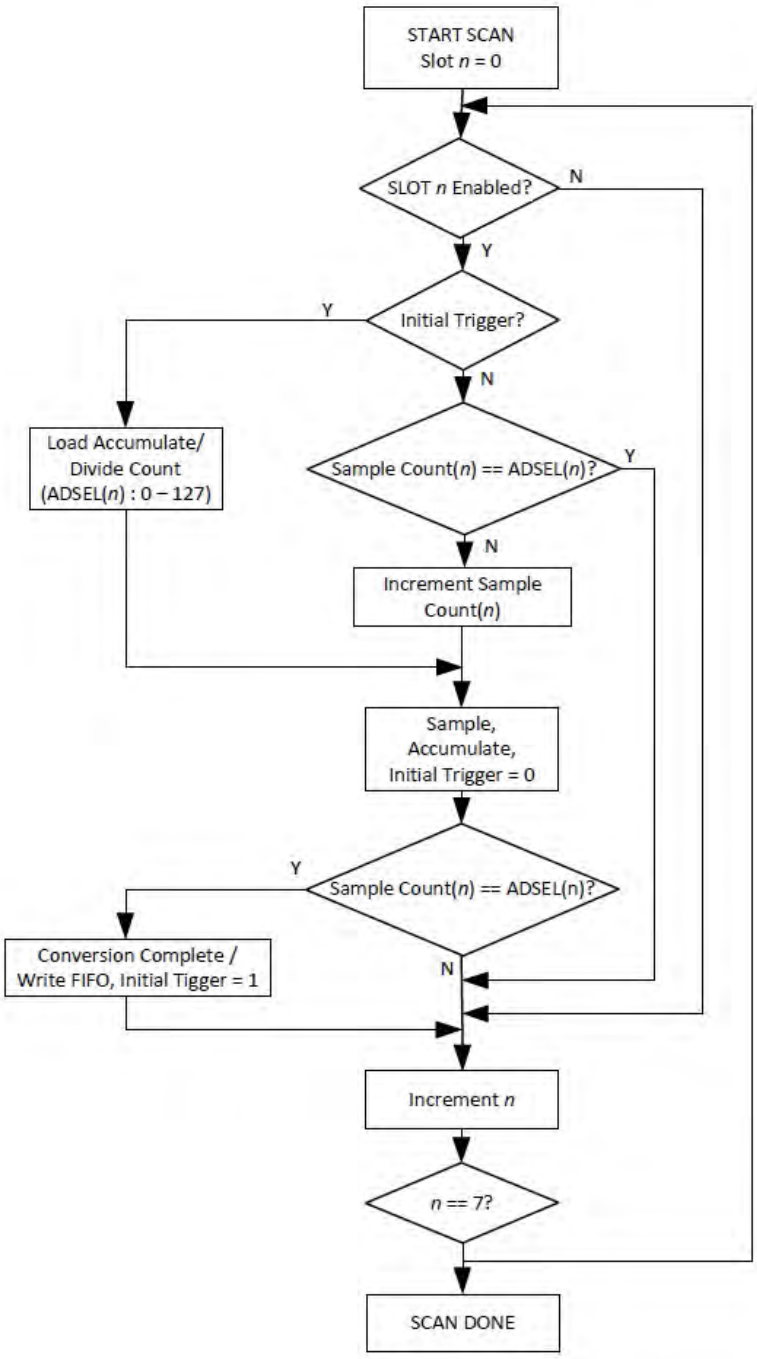
From a firmware perspective, the ADC mode controller is controlled from bit fields in the ADC configuration register and from the various bit fields in the eight slot configuration registers.

The most over-riding control is the ADC enable bit in the PWR_CTRL_DEVICE_EN register of the power control block. This bit must be set to '1' to enable power to the ADC subsystem. Furthermore, the ADCEN bit in the ADC configuration register is a global functional enable bit for general ADC operation. Setting this bit to zero has many of the effects of a software reset, such as reseting the FIFO pointers. Setting this bit to one enables the mode controller to examine its inputs and proceed to autonomously handle analog to digital conversions.

An ADC scan is the process of sampling the analog voltages at each input of the ADC of Figure 15-1 on page 427 following a trigger event. If the ADC is enabled and one or more slots are enabled, a scan is initiated after the ADC receives a trigger through one of the configured trigger sources. The scan flowchart diagram can be found in Figure 15-2.

An ADC conversion is the process of averaging measurements following one or more scans for each slot that is enabled.

Figure 15-2: Scan Flowchart



15.4.1 Single Mode

In single mode, one trigger event produces one scan of all enabled slots. Depending on the settings of the accumulate and scale bit field for the active slots, this may or may not result in writing a result to the FIFO. When the trigger source is an external pin then one external pin transition of the proper polarity will result in one complete scan of all enabled slots. If the external pin is connected to a repetitive pulse source then repeating scans of all enabled slots are run at the input trigger rate.

15.4.2 Repeat Mode

Counter/Timer 3A has a bit in its configuration register that allows it to be a source of repetitive triggers for the ADC. If counter/timer 3 is initialized for this purpose then one only needs to turn on the RPTEN bit in the ADC configuration registers to enable this mode in the ADC.

NOTE: The mode controller does not process these repetitive triggers from the counter/timer until a first triggering event occurs from the normal trigger sources. Thus one can select software triggering in the TRIGSEL field and set up all of the other ADC registers for the desired sample acquisitions. Then one can write to the software trigger register and the mode controller will enter REPEAT mode. In repeat mode, the mode controller waits only for each successive counter/timer 3A input to launch a scan of all enabled slots.

15.4.3 Low Power Modes

An application may use the ADC in one of three power modes. Each mode has different implications from overall energy perspective relative to the startup latency from trigger-to-data as well as the standby power consumed. Table 15-12 is intended to provide guidance on which mode may be more effective based on latency tolerance. This table should only be used as a reference.

Table 15-12: UART Bus Interface Clock Selection

LPMODE	Definition	Entry Latency
0	ADC is kept active continuously (used in continuous sampling scenarios)	0 (requires initial calibration)
1	ADC is mostly powered off between samples, HFRC is duty cycled between samples. No calibration required after initial calibration)	< 70µs (shorter for lower resolution)
2	ADC is completely powered off between samples, HFRC is duty cycled between samples. Requires recalibration for each conversion.	< 660µs

15.4.3.1 Low Power Mode 0

Low Power Mode 0 (LPMODE0) enables the lowest latency from trigger to conversion data available. This mode leaves the reference buffer powered on between scans to bypass any startup latency between triggers[1].

15.4.3.2 Low Power Mode 1

Low power mode 1 (LPMODE1) is a power mode whereby the ADC Digital Controller will automatically power off the ADC clocks, analog ADC and reference buffer between scans while maintaining ADC calibration data. This mode may operate autonomously without CPU interaction, even while the CPU is in sleep or deep sleep mode for repeat mode triggers or hardware triggers. While operating in this mode, the ADC Digital Controller may be used to burst through multiple scans enabling max sample rate data collection if the triggers are running at a rate at least 2x the maximum sample rate until the final scan has completed. When a scan completes without a pending trigger latched, the ADC subsystem will enter a low power state until the next trigger event.

15.4.3.3 Low Power Mode 2

If desirable, for applications requiring infrequent conversions, software may choose to operate the ADC in LPMODE2, whereby the full ADC Analog and Digital subsystem remains completely powered off between samples. In this use case, the software configures the power control ADC enable register followed by configuring the ADC slots and the ADC configuration register between conversion data collections, followed by disabling the ADC in the power control ADC enable register. Although this mode provides extremely low power operation, using the ADC in this mode will result in a cold start latency including reference buffer stabilization delay and a calibration sequence 100's of microseconds, nominally. In this mode, the ADC must be reconfigured prior to any subsequent ADC operation.

15.5 Interrupts

The ADC has 6 interrupt status bits with corresponding interrupt enable bits, as follows:

1. Conversion Complete Interrupt
2. Scan Complete Interrupt
3. FIFO Overflow Level 1
4. FIFO Overflow Level 2
5. Window Comparator Excursion Interrupt (a.k.a. outside interrupt)
6. Window Comparator Incursion Interrupt (a.k.a. inside interrupt)

[1] The reference buffer will not be powered on when the ADC is configured for external reference.

The window comparator interrupts are discussed above. See *Section 15.3.4 Window Comparator on page 436*.

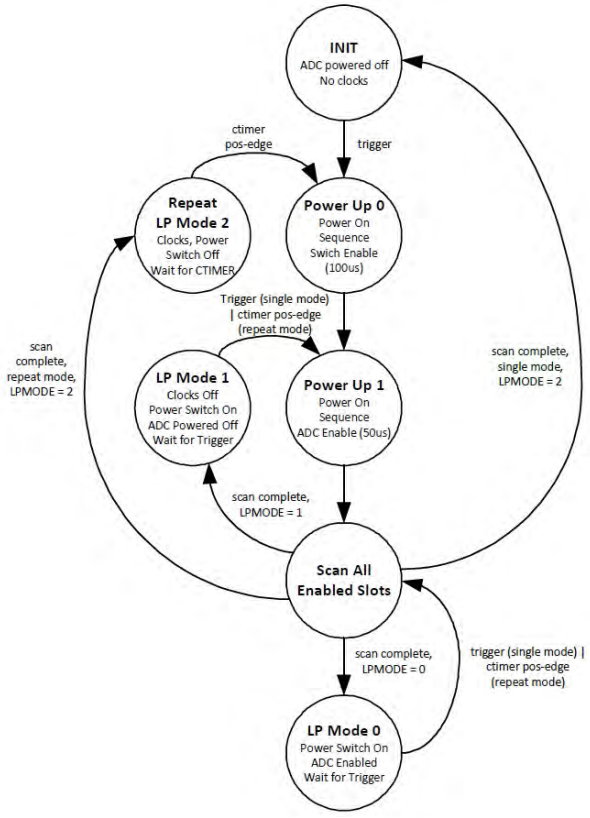
There are two interrupts based on the fullness of the FIFO. When the respective interrupts are enabled, Overflow 1 fires when the FIFO reaches 75% full, viz. 6 entries. Overflow 2 fires when the FIFO is completely full.

When enabled, the conversion complete interrupt fires when a single slot completes its conversion and the resulting conversion data is pushed into the FIFO.

When enabled, the scan complete interrupt indicates that all enabled slots have sampled their respective channels following a trigger event.

When a single slot is enabled and programmed to average over exactly one measurement and the scan complete and conversion complete interrupts are enabled, a trigger event will result in the conversion complete and scan complete interrupts firing simultaneously upon completion of the ADC scan. Again, if both respective interrupts are enabled and a single slot is enabled and programmed to average over 128 measurements, 128 trigger events result in 128 scan complete interrupts and exactly one conversion complete interrupt following the 128 ADC scans. When multiple slots are enabled with different settings for the number of measurements to average, the conversion complete interrupt signifies that one or more of the conversions have completed and the FIFO contains valid data for one or more of the slot conversions.

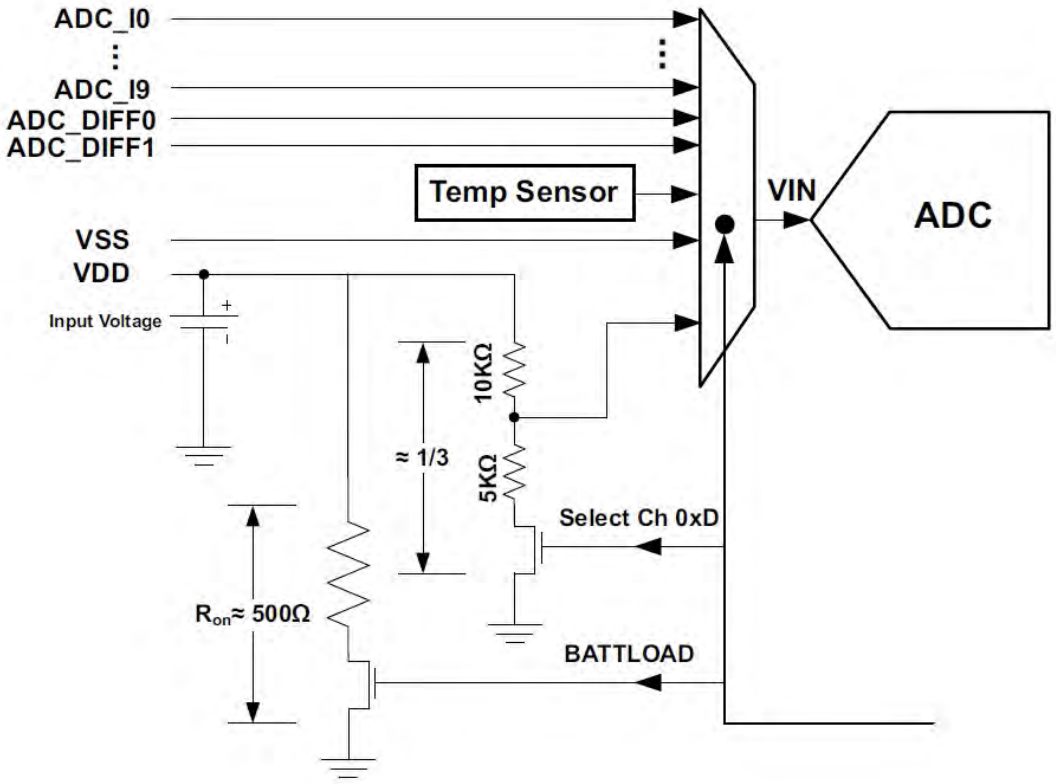
Figure 15-3: ADC State Diagram



15.6 Voltage Divider and Switchable Battery Load

The Apollo2 SoC's ADC includes a switchable voltage divider that enables the ADC to measure the input voltage to the VDD rail. In most systems this will be the battery voltage applied to the SoC chip. The voltage divider is only switched on when one of the active slots is selecting analog mux channel 15. That is only when the mode controller is ultimately triggered and powers up the ADC block for a conversion scan of all active slots. Otherwise, the voltage divider is turned off.

Figure 15-4: Switchable Battery Load



The switchable load resistor is enabled by the BATTLOAD bit as shown in the ADC-BATTLOAD Register of the MCUCTRL Registers.

This feature is used to help estimate the health of the battery chemistry by estimating the internal resistance of the battery.

15.7 ADC Registers

Analog Digital Converter Control
INSTANCE 0 BASE ADDRESS:0x50010000

This is the detailed description of the Analog Digital Converter Register Block. The ADC Register Block contains the software control for enablement, slot configuration, clock configuration, trigger configuration, temperature sensor enablement, power modes, accumulate/divide, window comparison and interrupt control for the ADC functional unit.

15.7.1 Register Memory Map

Table 15-13: ADC Register Map

Address(es)	Registered Name	Description
0x50010000	CFG	Configuration Register
0x50010004	STAT	ADC Power Status
0x50010008	SWT	Software trigger
0x5001000C	SL0CFG	Slot 0 Configuration Register
0x50010010	SL1CFG	Slot 1 Configuration Register
0x50010014	SL2CFG	Slot 2 Configuration Register
0x50010018	SL3CFG	Slot 3 Configuration Register
0x5001001C	SL4CFG	Slot 4 Configuration Register
0x50010020	SL5CFG	Slot 5 Configuration Register
0x50010024	SL6CFG	Slot 6 Configuration Register
0x50010028	SL7CFG	Slot 7 Configuration Register
0x5001002C	WULIM	Window Comparator Upper Limits Register
0x50010030	WLLIM	Window Comparator Lower Limits Register
0x50010038	FIFO	FIFO Data and Valid Count Register
0x50010200	INTEN	ADC Interrupt registers: Enable
0x50010204	INTSTAT	ADC Interrupt registers: Status
0x50010208	INTCLR	ADC Interrupt registers: Clear
0x5001020C	INTSET	ADC Interrupt registers: Set

15.7.2 ADC Registers

15.7.2.1 CFG Register

Configuration Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x50010000

The ADC Configuration Register contains the software control for selecting the clock frequency used for the SAR conversions, the trigger polarity, the trigger select, the reference voltage select, the low power mode, the operating mode (single scan per trigger vs. repeating mode) and ADC enable.

Table 15-14: CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD						CLKSEL		RSVD				TRIGPOL	TRIGSEL			RSVD						REFSEL		RSVD			CKMODE	LPMODE	RPTEN	RSVD	ADCEN

Table 15-15: CFG Register Bits

Bit	Name	Reset	RW	Description
31:26	RSVD	0x0	RO	RESERVED.
25:24	CLKSEL	0x0	RW	Select the source and frequency for the ADC clock. All values not enumerated below are undefined. OFF = 0x0 - Off mode. The HFRC or HFRC_DIV2 clock must be selected for the ADC to function. The ADC controller automatically shuts off the clock in it's low power modes. When setting ADCEN to '0', the CLKSEL should remain set to one of the two clock selects for proper power down sequencing. HFRC = 0x1 - HFRC Core Clock Frequency HFRC_DIV2 = 0x2 - HFRC Core Clock / 2
23:20	RSVD	0x0	RO	RESERVED.
19	TRIGPOL	0x0	RW	This bit selects the ADC trigger polarity for external off chip triggers. RISING_EDGE = 0x0 - Trigger on rising edge. FALLING_EDGE = 0x1 - Trigger on falling edge.
18:16	TRIGSEL	0x0	RW	Select the ADC trigger source. EXT0 = 0x0 - Off chip External Trigger0 (ADC_ET0) EXT1 = 0x1 - Off chip External Trigger1 (ADC_ET1) EXT2 = 0x2 - Off chip External Trigger2 (ADC_ET2) EXT3 = 0x3 - Off chip External Trigger3 (ADC_ET3) VCOMP = 0x4 - Voltage Comparator Output SWT = 0x7 - Software Trigger
15:10	RSVD	0x0	RO	RESERVED.
9:8	REFSEL	0x0	RW	Select the ADC reference voltage. INT2P0 = 0x0 - Internal 2.0V Bandgap Reference Voltage INT1P5 = 0x1 - Internal 1.5V Bandgap Reference Voltage EXT2P0 = 0x2 - Off Chip 2.0V Reference EXT1P5 = 0x3 - Off Chip 1.5V Reference

Table 15-15: CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
7:5	RSVD	0x0	RO	RESERVED.
4	CKMODE	0x0	RW	Clock mode register LPCKMODE = 0x0 - Disable the clock between scans for LPMODE0. Set LPCKMODE to 0x1 while configuring the ADC. LLCKMODE = 0x1 - Low Latency Clock Mode. When set, HFRC and the adc_clk will remain on while in functioning in LPMODE0.
3	LPMODE	0x0	RW	Select power mode to enter between active scans. MODE0 = 0x0 - Low Power Mode 0. Leaves the ADC fully powered between scans with minimum latency between a trigger event and sample data collection. MODE1 = 0x1 - Low Power Mode 1. Powers down all circuity and clocks associated with the ADC until the next trigger event. Between scans, the reference buffer requires up to 50us of delay from a scan trigger event before the conversion will commence while operating in this mode.
2	RPTEN	0x0	RW	This bit enables Repeating Scan Mode. SINGLE_SCAN = 0x0 - In Single Scan Mode, the ADC will complete a single scan upon each trigger event. REPEATING_SCAN = 0x1 - In Repeating Scan Mode, the ADC will complete it's first scan upon the initial trigger event and all subsequent scans will occur at regular intervals defined by the configuration programmed for the CTTMRA3 internal timer until the timer is disabled or the ADC is disabled. When disabling the ADC (setting ADCEN to '0'), the RPTEN bit should be cleared.
1	RSVD	0x0	RO	RESERVED.
0	ADCEN	0x0	RW	This bit enables the ADC module. While the ADC is enabled, the ADCCFG and SLOT Configuration register settings must remain stable and unchanged. All configuration register settings, slot configuration settings and window comparison settings should be written prior to setting the ADCEN bit to '1'. DIS = 0x0 - Disable the ADC module. EN = 0x1 - Enable the ADC module.

15.7.2.2 STAT Register

ADC Power Status
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x50010004

This register indicates the basic power status for the ADC. For detailed power status, see the power control power status register. ADC power mode 0 indicates the ADC is in it's full power state and is ready to process scans. ADC Power mode 1 indicates the ADC enabled and in a low power state.

Table 15-16: STAT Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																															PWDSTAT

Table 15-17: STAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	PWDSTAT	0x0	RO	Indicates the power-status of the ADC. ON = 0x0 - Powered on. POWERED_DOWN = 0x1 - ADC Low Power Mode 1.

15.7.2.3 SWT Register

Software trigger
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x50010008
This register enables initiating an ADC scan through software.

Table 15-18: SWT Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																								SWT							

Table 15-19: SWT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:0	SWT	0x0	RW	Writing 0x37 to this register generates a software trigger. GEN_SW_TRIGGER = 0x37 - Writing this value generates a software trigger.

15.7.2.4 SL0CFG Register

Slot 0 Configuration Register
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x5001000C
Slot 0 Configuration Register.

Table 15-20: SL0CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					ADSEL0			RSVD						PRMODE0		RSVD				CHSEL0				RSVD						WCEN0	SLEN0

Table 15-21: SL0CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL0	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE0	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.

Table 15-21: SL0CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
11:8	CHSEL0	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN0	0x0	RW	This bit enables the window compare function for slot 0. WCEN = 0x1 - Enable the window compare for slot 0.
0	SLEN0	0x0	RW	This bit enables slot 0 for ADC conversions. SLEN = 0x1 - Enable slot 0 for ADC conversions.

15.7.2.5 SL1CFG Register

Slot 1 Configuration Register
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x50010010
Slot 1 Configuration Register.

Table 15-22: SL1CFG Register Bits

31 30 29 28 27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02	01	00
RSVD	ADSEL1 RSVD	PRMODE1	RSVD CHSEL1	RSVD	WCEN1	SLEN1

Table 15-23: SL1CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL1	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE1	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL1	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN1	0x0	RW	This bit enables the window compare function for slot 1. WCEN = 0x1 - Enable the window compare for slot 1.
0	SLEN1	0x0	RW	This bit enables slot 1 for ADC conversions. SLEN = 0x1 - Enable slot 1 for ADC conversions.

15.7.2.6 SL2CFG Register

Slot 2 Configuration Register
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x50010014
Slot 2 Configuration Register.

Table 15-24: SL2CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					ADSEL2			RSVD						PRMODE2		RSVD				CHSEL2				RSVD						WCEN2	SLEN2

Table 15-25: SL2CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL2	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE2	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.

Table 15-25: SL2CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
11:8	CHSEL2	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN2	0x0	RW	This bit enables the window compare function for slot 2. WCEN = 0x1 - Enable the window compare for slot 2.
0	SLEN2	0x0	RW	This bit enables slot 2 for ADC conversions. SLEN = 0x1 - Enable slot 2 for ADC conversions.

15.7.2.7 SL3CFG Register

Slot 3 Configuration Register
OFFSET: 0x00000018
INSTANCE 0 ADDRESS: 0x50010018
Slot 3 Configuration Register.

Table 15-26: SL3CFG Register Bits

31 30 29 28 27 26 25 24	23 22 21 20 19 18	17	16 15 14 13 12	11 10 09 08 07 06 05 04	03 02	01	00
RSVD	ADSEL3 RSVD	PRMODE3	RSVD	CHSEL3 RSVD		WCEN3	SLEN3

Table 15-27: SL3CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL3	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE3	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL3	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN3	0x0	RW	This bit enables the window compare function for slot 3. WCEN = 0x1 - Enable the window compare for slot 3.
0	SLEN3	0x0	RW	This bit enables slot 3 for ADC conversions. SLEN = 0x1 - Enable slot 3 for ADC conversions.

15.7.2.8 SL4CFG Register

Slot 4 Configuration Register
OFFSET: 0x0000001C
INSTANCE 0 ADDRESS: 0x5001001C
Slot 4 Configuration Register.

Table 15-28: SL4CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					ADSEL4			RSVD						PRMODE4		RSVD				CHSEL4				RSVD						WCEN4	SLEN4

Table 15-29: SL4CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL4	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE4	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.

Table 15-29: SL4CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
11:8	CHSEL4	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN4	0x0	RW	This bit enables the window compare function for slot 4. WCEN = 0x1 - Enable the window compare for slot 4.
0	SLEN4	0x0	RW	This bit enables slot 4 for ADC conversions. SLEN = 0x1 - Enable slot 4 for ADC conversions.

15.7.2.9 SL5CFG Register

Slot 5 Configuration Register
OFFSET: 0x00000020
INSTANCE 0 ADDRESS: 0x50010020
Slot 5 Configuration Register.

Table 15-30: SL5CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					ADSEL5			RSVD						PRMODE5		RSVD				CHSEL5				RSVD						WCEN5	SLEN5

Table 15-31: SL5CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL5	0x0	RW	Select number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE5	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL5	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN5	0x0	RW	This bit enables the window compare function for slot 5. WCEN = 0x1 - Enable the window compare for slot 5.
0	SLEN5	0x0	RW	This bit enables slot 5 for ADC conversions. SLEN = 0x1 - Enable slot 5 for ADC conversions.

15.7.2.10 SL6CFG Register

Slot 6 Configuration Register
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x50010024
Slot 6 Configuration Register.

Table 15-32: SL6CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD					ADSEL6			RSVD						PRMODE6		RSVD				CHSEL6				RSVD						WCEN6	SLEN6

Table 15-33: SL6CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL6	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE6	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.

Table 15-33: SL6CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
11:8	CHSEL6	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN6	0x0	RW	This bit enables the window compare function for slot 6. WCEN = 0x1 - Enable the window compare for slot 6.
0	SLEN6	0x0	RW	This bit enables slot 6 for ADC conversions. SLEN = 0x1 - Enable slot 6 for ADC conversions.

15.7.2.11 SL7CFG Register

Slot 7 Configuration Register
OFFSET: 0x00000028
INSTANCE 0 ADDRESS: 0x50010028
Slot 7 Configuration Register.

Table 15-34: SL7CFG Register Bits

31 30 29 28 27 26 25 24	23 22 21 20 19 18	17	16 15 14 13 12 11 10 09	08 07 06 05 04 03 02	01	00
RSVD	ADSEL7 RSVD	PRMODE7	RSVD CHSEL7	RSVD	WCEN7	SLEN7

Table 15-35: SL7CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL7	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:18	RSVD	0x0	RO	RESERVED.
17:16	PRMODE7	0x0	RW	Set the Precision Mode For Slot. P14B = 0x0 - 14-bit precision mode P12B = 0x1 - 12-bit precision mode P10B = 0x2 - 10-bit precision mode P8B = 0x3 - 8-bit precision mode
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL7	0x0	RW	Select one of the 14 channel inputs for this slot. SE0 = 0x0 - single ended external GPIO connection to pad16. SE1 = 0x1 - single ended external GPIO connection to pad29. SE2 = 0x2 - single ended external GPIO connection to pad11. SE3 = 0x3 - single ended external GPIO connection to pad31. SE4 = 0x4 - single ended external GPIO connection to pad32. SE5 = 0x5 - single ended external GPIO connection to pad33. SE6 = 0x6 - single ended external GPIO connection to pad34. SE7 = 0x7 - single ended external GPIO connection to pad35. SE8 = 0x8 - single ended external GPIO connection to pad13. SE9 = 0x9 - single ended external GPIO connection to pad12. DF0 = 0xA - differential external GPIO connections to pad12(N) and pad13(P). DF1 = 0xB - differential external GPIO connections to pad15(N) and pad14(P). TEMP = 0xC - internal temperature sensor. BATT = 0xD - internal voltage divide-by-3 connection. VSS = 0xE - Input VSS
7:2	RSVD	0x0	RO	RESERVED.
1	WCEN7	0x0	RW	This bit enables the window compare function for slot 7. WCEN = 0x1 - Enable the window compare for slot 7.
0	SLEN7	0x0	RW	This bit enables slot 7 for ADC conversions. SLEN = 0x1 - Enable slot 7 for ADC conversions.

15.7.2.12 WULIM Register

Window Comparator Upper Limits Register
OFFSET: 0x0000002C
INSTANCE 0 ADDRESS: 0x5001002C
Window Comparator Upper Limits Register.

Table 15-36: WULIM Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												ULIM																			

Table 15-37: WULIM Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	RESERVED.
19:0	ULIM	0x0	RW	Sets the upper limit for the wondow comparator.

15.7.2.13 WLLIM Register

Window Comparator Lower Limits Register
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x50010030
Window Comparator Lower Limits Register.

Table 15-38: WLLIM Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												LLIM																			

Table 15-39: WLLIM Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	RESERVED.
19:0	LLIM	0x0	RW	Sets the lower limit for the wondow comparator.

15.7.2.14 FIFO Register

FIFO Data and Valid Count Register
OFFSET: 0x00000038
INSTANCE 0 ADDRESS: 0x50010038

The ADC FIFO Register contains the slot number and fifo data for the oldest conversion data in the FIFO. The COUNT field indicates the total number of valid entries in the FIFO. A write to this register will pop one of the FIFO entries off the FIFO and decrease the COUNT by 1 if the COUNT is greater than zero.

Table 15-40: FIFO Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	SLOTNUM			COUNT								DATA																			

Table 15-41: FIFO Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED.
30:28	SLOTNUM	0x0	RO	Slot number associated with this FIFO data.
27:20	COUNT	0x0	RO	Number of valid entries in the ADC FIFO.
19:0	DATA	0x0	RO	Oldest data in the FIFO.

15.7.2.15 INTEN Register

ADC Interrupt registers: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x50010200
Set bits in this register to allow this module to generate the corresponding interrupt.

Table 15-42: INTEN Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 15-43: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

15.7.2.16 INTSTAT Register

ADC Interrupt registers: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x50010204
Read bits from this register to discover the cause of a recent interrupt.

Table 15-44: INTSTAT Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 15-45: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

15.7.2.17 INTCLR Register

ADC Interrupt registers: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x50010208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 15-46: INTCLR Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 15-47: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

15.7.2.18 INTSET Register

ADC Interrupt registers: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x5001020C
Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes)

Table 15-48: INTSET Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																										WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 15-49: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.

Table 15-49: INTSET Register Bits *(Continued)*

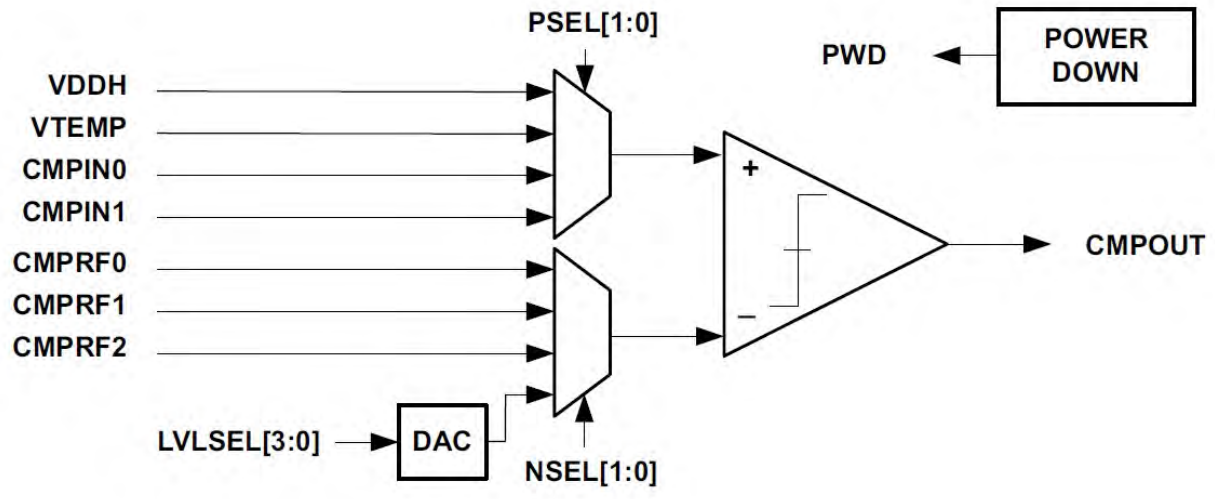
Bit	Name	Reset	RW	Description
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

SECTION

16

Voltage Comparator Module

Figure 16-1: Block Diagram for the Voltage Comparator Module



16.1 Functional Overview

The Voltage Comparator Module, shown in Figure 16-1, measures a user-selectable voltage at all times. It provides interrupt and software access to the comparator output with multiple options for input and reference voltages. It can be configured to generate an interrupt when the monitored voltage rises above a user-configurable threshold or when the monitored voltage drops below a user-configurable threshold.

The voltage to be monitored is selected by programming the comparator's positive terminal signal, PSEL[1:0], and may be any of:

1. The supply voltage (VDDH), or
2. The PTAT voltage from the temperature sensor (VTEMP), or
3. Two external voltage channels (CMPIN0 or CMPIN1), or
4. The filtered PGA outputs (PGA_A0, PGA_A1, PGA_B0, PGA_B1)

The reference voltage is selected by programming the comparator's negative terminal, NSEL[1:0] and may be any of:

1. Three external voltage channels (CMPRF0, CMPRF1 or CMPRF2), or
2. The internally generated reference (VREFINT)

The internal reference voltage is tuned using an on-chip DAC with level select signal LVLSEL[3:0]. When using external inputs or reference inputs, the associated pads must be configured using the GPIO function selects explained in the GPIO document section.

The Voltage Comparator CMPOUT output will remain high while the voltage at the positive input is above the voltage at reference input. The CMPOUT output will transition low when the voltage at the positive input to the comparator falls below the reference input taking into account hysteresis (see *Section 18.10 Voltage Comparator (VCOMP) on page 482*). The CMPOUT output is directly accessible by software by reading the CMPOUT field in the status register. The OUTHI interrupt will be set if enabled and the CMPOUT transitions high or if it is high at the time the interrupt is enabled. Similarly, the OUTLOW interrupt will be set if enabled and the CMPOUT output transitions low or if it is low at the time the interrupt is enabled.

The Voltage Comparator Module is enabled by default and may be powered off by writing 0x37 to the PWDKEY register.

16.2 VCOMP Registers

Voltage Comparator
INSTANCE 0 BASE ADDRESS:0x4000C000

This is the detailed description of the Voltage Comparator Register Block. The Voltage Comparator Register Block contains the software control for selecting the comparator inputs, power down control, observing comparator output status and enabling interrupts.

16.2.1 Register Memory Map

Table 16-1: VCOMP Register Map

Address(es)	Registered Name	Description
0x4000C000	CFG	Configuration Register
0x4000C004	STAT	Status Register
0x4000C008	PWDKEY	Key Register for Powering Down the Voltage Comparator

Table 16-1: VCOMP Register Map *(Continued)*

Address(es)	Registered Name	Description
0x4000C200	INTEN	Voltage Comparator Interrupt registers: Enable
0x4000C204	INTSTAT	Voltage Comparator Interrupt registers: Status
0x4000C208	INTCLR	Voltage Comparator Interrupt registers: Clear
0x4000C20C	INTSET	Voltage Comparator Interrupt registers: Set

16.2.2 VCOMP Registers

16.2.2.1 CFG Register

Configuration Register
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x4000C000

The Voltage Comparator Configuration Register contains the software control for selecting between the 4 options for the positive input as well as the multiple options for the reference input.

Table 16-2: CFG Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD												LVLSEL				RSVD						NSEL		RSVD							PSEL

Table 16-3: CFG Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bit field is reserved for future use.
19:16	LVLSEL	0x0	RW	When the reference input NSEL is set to DAC, this bit field selects the voltage level for the negative input to the comparator. If a voltage which is higher than VDDH is selected, the reference voltage will be clamped to VDDH. 0P58V = 0x0 - Set Reference input to 0.58 Volts. 0P77V = 0x1 - Set Reference input to 0.77 Volts. 0P97V = 0x2 - Set Reference input to 0.97 Volts. 1P16V = 0x3 - Set Reference input to 1.16 Volts. 1P35V = 0x4 - Set Reference input to 1.35 Volts. 1P55V = 0x5 - Set Reference input to 1.55 Volts. 1P74V = 0x6 - Set Reference input to 1.74 Volts. 1P93V = 0x7 - Set Reference input to 1.93 Volts. 2P13V = 0x8 - Set Reference input to 2.13 Volts. 2P32V = 0x9 - Set Reference input to 2.32 Volts. 2P51V = 0xA - Set Reference input to 2.51 Volts. 2P71V = 0xB - Set Reference input to 2.71 Volts. 2P90V = 0xC - Set Reference input to 2.90 Volts. 3P09V = 0xD - Set Reference input to 3.09 Volts. 3P29V = 0xE - Set Reference input to 3.29 Volts. 3P48V = 0xF - Set Reference input to 3.48 Volts.
15:10	RSVD	0x0	RO	This bit field is reserved for future use.

Table 16-3: CFG Register Bits *(Continued)*

Bit	Name	Reset	RW	Description
9:8	NSEL	0x0	RW	This bit field selects the negative input to the comparator. VREFEXT1 = 0x0 - Use CMPRF0 for reference input. VREFEXT2 = 0x1 - Use CMPRF1 for reference input. VREFEXT3 = 0x2 - Use CMPRF2 for reference input. DAC = 0x3 - Use DAC output selected by LVLSEL for reference input.
7:2	RSVD	0x0	RO	This bit field is reserved for future use.
1:0	PSEL	0x0	RW	This bit field selects the positive input to the comparator. VDDADJ = 0x0 - Use VDDH for the positive input. VTEMP = 0x1 - Use the temperature sensor output for the positive input. Note: If this channel is selected for PSEL, the bandgap circuit required for temperature comparisons will automatically turn on. The band-gap circuit requires 11 us to stabilize. VEXT1 = 0x2 - Use CMPIN0 for positive input. VEXT2 = 0x3 - Use CMPIN1 for positive input.

16.2.2.2 STAT Register

Status Register
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x4000C004
Status Register

Table 16-4: STAT Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														PWDSTAT	CMPOUT

Table 16-5: STAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bit field is reserved for future use.
1	PWDSTAT	0x0	RO	This bit indicates the power down state of the voltage comparator. POWERED_DOWN = 0x1 - The voltage comparator is powered down.
0	CMPOUT	0x0	RO	This bit is 1 if the positive input of the comparator is greater than the negative input. VOUT_LOW = 0x0 - The negative input of the comparator is greater than the positive input. VOUT_HIGH = 0x1 - The positive input of the comparator is greater than the negative input.

16.2.2.3 PWDKEY Register

Key Register for Powering Down the Voltage Comparator
OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x4000C008
Key Register for Powering Down the Voltage Comparator.

Table 16-6: PWDKEY Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PWDKEY																															

Table 16-7: PWDKEY Register Bits

Bit	Name	Reset	RW	Description
31:0	PWDKEY	0x0	RW	Key register value. Key = 0x37 - Key

16.2.2.4 INTEN Register

Voltage Comparator Interrupt registers: Enable
OFFSET: 0x00000200
INSTANCE 0 ADDRESS: 0x4000C200

Set bits in this register to allow this module to generate the corresponding interrupt.

Table 16-8: INTEN Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														OUTHI	OUTLOW

Table 16-9: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bit field is reserved for future use.
1	OUTHI	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

16.2.2.5 INTSTAT Register

Voltage Comparator Interrupt registers: Status
OFFSET: 0x00000204
INSTANCE 0 ADDRESS: 0x4000C204
Read bits from this register to discover the cause of a recent interrupt.

Table 16-10: INTSTAT Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														OUTHI	OUTLOW

Table 16-11: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bit field is reserved for future use.
1	OUTHI	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

16.2.2.6 INTCLR Register

Voltage Comparator Interrupt registers: Clear
OFFSET: 0x00000208
INSTANCE 0 ADDRESS: 0x4000C208
Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 16-12: INTCLR Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														OUTHI	OUTLOW

Table 16-13: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bit field is reserved for future use.
1	OUTHI	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

16.2.2.7 INTSET Register

Voltage Comparator Interrupt registers: Set
OFFSET: 0x0000020C
INSTANCE 0 ADDRESS: 0x4000C20C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 16-14: INTSET Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																														OUTHI	OUTLOW

Table 16-15: INTSET Register Bits

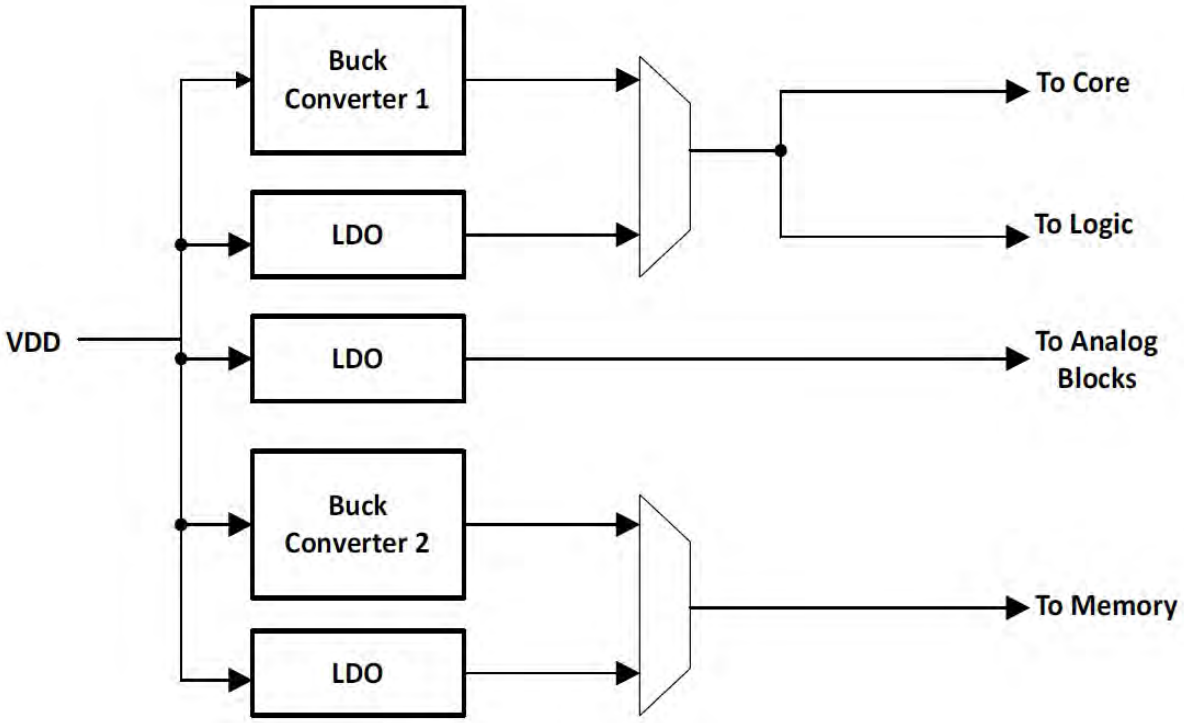
Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bit field is reserved for future use.
1	OUTHI	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

SECTION

17

Voltage Regulator Module

Figure 17-1: Block Diagram for the Voltage Regulator Module



17.1 Functional Overview

The Voltage Regulator Module down-converts and regulates the supply voltage, VDD, with extremely high efficiency. A pair of Buck Converters enables down-conversion from the power supply input (e.g., a battery) at efficiency of >80%. With ultra-low quiescent current, the Buck Converters are optimized for low power envi-

ronments. There are also integrated low dropout linear regulators which are used in very low power modes and can also be utilized to provide a lower cost system solution by eliminating the need for the external inductors required in buck mode. The VDDC and VDDF capacitors are still required for the internal LDOs.

The Buck Converters and LDOs of the Voltage Regulator Module are tightly coupled to the various low power modes in the Apollo2 SoC. When the Apollo2 SoC enters deep sleep mode, the Buck Converters can be optionally powered down and bypassed, and the LDOs can be placed in an extreme low power mode with only nanoamps of quiescent current.

Table 17-1: VREG Status

Position	Name	R/W	Default	Description
1	BUCK2STABLE	R	0x0	Indicates whether the output of Buck Converter 2 is available and stable after activation (0: output not stable, 1: output stable). Buck Converter 2 will not be used by the Apollo2 SoC until its output is stable.
0	BUCK1STABLE	R	0x0	Indicates whether the output of Buck Converter 1 is available and stable after activation (0: output not stable, 1: output stable). Buck Converter 1 will not be used by the Apollo2 SoC until its output is stable.

SECTION

18 Electrical Characteristics

For all tables T_A = -40ºC to 85ºC, Typical values at 25ºC, 1.8v, unless otherwise specified.

18.1 Absolute Maximum Ratings

The absolute maximum ratings are the limits to which the device can be subjected without permanently damaging the device and are stress ratings only. Device reliability may be adversely affected by exposure to absolute-maximum ratings for extended periods. Functional operation of the device at the absolute maximum ratings or any other conditions beyond the recommended operating conditions is not implied.

Table 18-1: Absolute Maximum Ratings

Symbol	Parameter	Test Conditions	Min	Max	Unit
V_{DDP}	Pad supply voltage		-	3.8	V
V_{DDH}	Digital supply voltage		-	3.8	V
V_{DDA}	Analog supply voltage		-	3.8	V
V_{INPUT}	Voltage on any input pin		0	V_{DDP}	V
I_{SRC_STD}	Standard output pin source continuous current		-	16	mA
I_{SINK_STD}	Standard output pin sink continuous current		-	16	mA
I_{HSC_PWR}	High side power switch continuous source current[1]		-	50	mA
I_{HSP_PWR}	High side power switch pulsed source current[1]	10 ms pulse, 1% duty cycle		150	mA
I_{LSC_PWR}	Low side power switch continuous sink current[2]			50	mA
I_{LSP_PWR}	Low side power switch pulsed sink current[2]	10 ms pulse, 1% duty cycle		150	mA
T_S	Storage temperature		55	-125	℃
T_J	Junction temperature		-40	86	℃
T_{OP}	Operating temperature		-40	85	℃

Table 18-1: Absolute Maximum Ratings *(Continued)*

Symbol	Parameter	Test Conditions	Min	Max	Unit
θ_{JA}	Thermal resistance, junction to ambient	BGA Package on 4 layer PCB in still air, 3mW power dissipation		76.2	°C/W
θ_{JC}	Thermal resistance, junction to package case	BGA Package on 4 layer PCB in still air, 3mW power dissipation		17.0	°C/W
T_{REFLOW}	Reflow temperature	Reflow Profile per JEDEC J-STD-020D.1		260	°C
I_{LU}	Latch-up current			10	mA
V_{ESDHBM}	ESD Human Body Model (HBM)			2000	V
V_{ESDCDM}	ESD Charged Device Model (CDM)			250	V

[1] High side power switches are available on PAD22 and PAD41

[2] A low side power switch is available on PAD4

18.2 Recommended Operating Conditions

Table 18-2: Recommended Operating Conditions[1]

Symbol	Parameter	Min	Typ	Max	Unit
V_{DDP}	Pad supply voltage	1.755		3.63	V
V_{DDH}	Digital supply voltage	1.755		3.63	V
V_{DDA}	Analog supply voltage	1.755		3.63	V
V_{INPUT}	Voltage on any input pin	0		V_{DDH}/VIO	V
V_{OPLV_32K}	VDDH voltage (below BODL) for RTC and STIMER operation from the 32kHz crystal clock	1.2			V
V_{OPLV_LFRC}	VDDH voltage (below BODL) for RTC and STIMER operation from the LFRC clock	1.2			V
V_{OPLV_RET}	VDDH voltage (below BODL) for retention of RTC and CLKGEN registers, IOS LRAM, and STIMER SNVR0-2 registers	1.2			V
T_A	Ambient operating temperature	-40	-	85	°C

[1] $V_{DD} = V_{DDP} = V_{DDA} = V_{DDH}$

18.3 Current Consumption

Table 18-3: Current Consumption

Symbol	Parameter	Test Conditions[1,2]	VDD	Min	Typ	Max	Unit
I_{RUNFB}	Flash program run current, bucks enabled	Executing CoreMark from internal flash memory, cache enabled, HFRC = 48 MHz, all peripherals disabled, buck converters enabled, 8K SRAM, FLASH1 off	3.3 V	-	14.5	-	µA/MHz
			1.8 V	-	24.0	-	µA/MHz
I_{RUNF}	Flash program run current	Executing CoreMark from internal flash memory, cache enabled, HFRC = 48 MHz, all peripherals disabled, buck converters disabled, 8K SRAM, FLASH1 off	3.3 V	-	46.4	-	µA/MHz
			1.8 V	-	46.5	-	µA/MHz
$I_{RUNFNCB}$	Flash program run current, no cache	Executing CoreMark from internal flash memory, cache disabled, HFRC = 48 MHz, all peripherals disabled, buck converters enabled, 8K SRAM, FLASH1 off	3.3 V	-	13.6	-	µA/MHz
			1.8 V	-	21.5	-	µA/MHz
I_{RUNFNC}	Flash program run current, no cache	Executing CoreMark from internal flash memory, cache disabled, HFRC = 48 MHz, all peripherals disabled, buck converters disabled, 8K SRAM, FLASH1 off	3.3 V	-	40.9	-	µA/MHz
			1.8 V	-	40.9	-	µA/MHz
I_{RUNFSB}	SRAM program run current	Executing CoreMark from internal SRAM memory, HFRC = 48 MHz, all peripherals disabled, buck converters enabled, 8K SRAM, FLASH1 off	3.3 V	-	14.7	-	µA/MHz
			1.8 V	-	24.6	-	µA/MHz
I_{RUNFS}	SRAM program run current	Executing CoreMark from internal SRAM memory, HFRC = 48 MHz, all peripherals disabled, buck converters disabled, 64K SRAM, FLASH1 off	3.3 V	-	41.5	-	µA/MHz
			1.8 V	-	41.3	-	µA/MHz
I_{RUNPNB}	Flash program run current	Executing prime number calculator, buck converters enabled, all Flash/SRAM on	3.3 V	-	9.6	-	µA/MHz
			1.8 V	-	16.4	-	µA/MHz
I_{SS1}	Sleep mode 1 current	WFI instruction with SLEEP = 1, clocks gated, osc's ON, all power domains active idle, 8K SRAM	3.3 V	-	105	-	µA
			1.8 V	-	130	-	µA
$I_{DS2-256RET}$	System Deep Sleep mode 2 w/ full SRAM retention current	WFI instruction with SLEEPDEEP = 1, 256KB SRAM retained, registers retained, cache retained, XTAL running	3.3 V	-	6.4	-	µA
			1.8 V	-	5.7	-	µA
I_{DS2}	Deep sleep mode 2 current	WFI instruction with SLEEPDEEP = 1, 8kB SRAM retained, registers retained, cache retained, XTAL running	3.3 V	-	3.2	-	µA
			1.8 V	-	2.6	-	µA
I_{DS3}	Deep sleep mode 3 current	WFI instruction with SLEEPDEEP = 1, no SRAM/cache retained, XTAL/ HFRC off	3.3 V	-	2.9	-	µA
			1.8 V	-	2.3	-	µA
ADC Operating Current[3]							

Table 18-3: Current Consumption *(Continued)*

Symbol	Parameter	Test Conditions[1,2]	VDD	Min	Typ	Max	Unit
$I_{ADC\_RUN\_LPM0}$	ADC run mode low power mode 0	Average run current (LPMODE0: max conversion rate, single slot, 14-bit, CPU in deep sleep otherwise with 16kB SRAM and cache in retention. ADC / gated domains ON. All other IO domains OFF	3.3 V		795		µA
$I_{ADC\_RUN\_LPM1}$	ADC run mode low power mode 1	Average run current (LPMODE1: 1kHz sample period, single slot, 14-bit, CPU in deep sleep otherwise with 16kB SRAM and cache in retention, main ADC power domain ON. HW controlled ADC gated domain duty cycled between samples.	3.3 V		61		µA
$I_{ADC\_RUN\_LPM2}$	ADC run mode low power mode 2	Average run current (LPMODE2: 10Hz sample period, single slot, 14-bit, CPU in deep sleep otherwise with 16kB SRAM and cache in retention, main ADC power domain duty cycled by software with calibration each sample conversion	3.3 V		17		µA
Flash Memory Operating Current							
$I_{PROGRAM}$	Supply current during a page program	Buck Mode	3.3 V		1.8		mA
			1.8 V		1.3		mA
$I_{PROGRAM}$	Supply current during a page program	LDO Mode	3.3 V		4.1		mA
			1.8 V		3.9		mA
I_{ERASE}	Supply current during a page erase	Buck Mode	3.3 V		1.1		mA
			1.8 V		0.73		mA
I_{ERASE}	Supply current during a page erase	LDO Mode	3.3 V		3.4		mA
			1.8 V		3.3		mA
$I_{MASSERASE}$	Supply current during a mass erase	Buck Mode	3.3 V		1.1		mA
			1.8 V		0.72		mA
$I_{MASSERASE}$	Supply current during a mass erase	LDO Mode	3.3 V		3.4		mA
			1.8 V		3.3		mA

[1] Core clock (HCLK) is 48 MHz for each parameter unless otherwise noted.

[2] All values measured at 25°C.

[3] Base SoC power state is S_{DS2} with minimal SRAM retention. Current represents total current at battery.

18.4 Power Mode Transitions

Table 18-4: Power Mode Transitions

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
Buck Mode						
$T_{RUN\_TO\_SLEEP}$	Run to Sleep mode transition time	HCLK frequency = 48 MHz	-	$2/F_{HFRC}$	-	ns
$T_{RUN\_TO\_DEEPSLEEP}$	Run mode to Deep Sleep mode transition time	HCLK frequency = 48 MHz	-	11.6	12.5	µs
$T_{SLEEP\_TO\_RUN}$	Sleep to Run mode transition time		-	1.9	2.1	µs
$T_{DEEPSLEEP\_TO\_RUN}$	Deep-Sleep to Run mode transition time		-	24.5	25.5	µs
LDO Mode						
$T_{RUN\_TO\_SLEEP}$	Run to Sleep mode transition time	HCLK frequency = 48 MHz	-	$2/F_{HFRC}$	-	ns
$T_{RUN\_TO\_DEEPSLEEP}$	Run mode to Deep Sleep mode transition time	HCLK frequency = 48 MHz	-	11.5	12.5	µs
$T_{SLEEP\_TO\_RUN}$	Sleep to Run mode transition time		-	1.9	2.1	µs
$T_{DEEPSLEEP\_TO\_RUN}$	Deep-Sleep to Run mode transition time		-	16.5	17.5	µs

18.5 Clocks/Oscillators

Table 18-5: Clocks/Oscillators

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{HFRC}	HFRC frequency		46.5	48	49.5	MHz
F_{LFRC}	LFRC frequency		840	1024	1114	Hz
F_{XT}	XT frequency		-	32.768	-	kHz
F_{RTC}	RTC frequency		-	100	-	Hz
T_{J_LFRC}	LFRC RMS jitter	25°C, STDEV of period	-	3	13.5	µs
T_{J_XT}	32.768kHz crystal oscillator clock output jitter	At 25°C using a 32.768kHz tuning fork crystal, 7pF load caps on XI and XO, XTALBIASTRIM = 0	-	20	50	ns
DC_{HFRC}	HFRC duty cycle			50		%
DC_{LFRC}	LFRC duty cycle on CLKOUT	CLKGEN_CLKOUT_CKSEL = LFRC_DIV2		50		%
DC_{XT}	32.768kHz crystal oscillator clock output mean duty cycle	At 25°C using a 32.768kHz tuning fork crystal, 7pF load caps on XI and XO, XTALBIASTRIM = 24	-	42	-	%
C_{INT_XT}	Internal XI/XO pin capacitance			2.5		pF
$C_{EXT\_XT\_TOL}$	Allowed external XI/XO pin capacitance per pin		-	-	7	pF

Table 18-5: Clocks/Oscillators *(Continued)*

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{OF_XT}	XT oscillator failure detection frequency	LFRC @ 1024Hz, over 32ms sample period		8		kHz
OA_{XT}	XT oscillation allowance	At 25°C using a 32.768kHz tuning fork crystal, no external crystal load caps on XI and X0, XTALBIAS-TRIM = -25	400	-	-	kΩ
OA_{XT7}	XT oscillation allowance, 7pF load	At 25°C using a 32.768kHz tuning fork crystal, 7pF load caps on XI and XO, XTALBIASTRIM = 14	400	-	-	kΩ
V_{OH}	High-level output voltage		0.8 * V_{DDH}	-	-	V
V_{OL}	Low-level output voltage		-	-	0.2 * V_{DDH}	V
V_{IH}	Positive going input threshold voltage		-	-	0.7 * V_{DDH}	v
V_{IL}	Negative going input threshold voltage		0.3 * V_{DDH}	-	-	V
C_{GPI}	Input capacitance		-	2.5	-	
I_{IN}	Input pin leakage current	GPIO 0 measured at 25°C	-	0.13	50	nA
F_{EXTXT}	Externally sourced input XT clock frequency over temp and voltage			0.03277	20	MHz
T_{EXTXT_PW}	External XT Pulse Width		20			ns
F_{32KHz_XT}	Sourced to external XT frequency over temp and voltage			32.768		kHz

18.6 Analog-to-Digital Converter (ADC)

Table 18-6: Analog-to-Digital Converter (ADC)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
ANALOG INPUT						
V_{ADCIN}	Input voltage range single-ended input		0		V_{ADCREF}	V
V_{ADCIN_DIFF}	Input voltage range in differential mode		-V_{ADCREF}/2		+V_{ADCREF}/2	V
V_{ADCINN} V_{ADCINP}	Absolute differential input voltage range		0		VDDH	V
V_{ADCREF_15E}	External reference voltage range (1.5v mode)		1.425	1.5	1.575	V
V_{ADCREF_20E}	External reference voltage range (2.0v mode)		1.9	2.0	2.1	V
V_{ADCREF_15I}	Internal reference voltage range (1.5v mode)		1.475	1.5	1.525	V

Table 18-6: Analog-to-Digital Converter (ADC) *(Continued)*

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
V_{ADCREF_20I}	Internal reference voltage range (2.0v mode)		1.975	2.0	2.025	V
I_{ADCIN}	ADC channel pin input leakage current (static)	GPIO 16 measured at 85°C, TTT part, 2.05 V; 2 V on pad; 10.8 nA with 3.63 or 1.8 V on VDD and pad	-	0.5	50	nA
Z_{ADC_CH0}	ADC Channel 0 Input Impedance	VDD = 3.63 V	360	720		kΩ
Z_{ADC_CH1} - Z_{ADC_CH7}	ADC Channel 1 - Channel 7 Input Impedance	VDD = 3.63 V	180	3600		MΩ
C_{ADCIN}	Input source capacitance			4		pF
$C_{ADCVREF}$	External ADC capacitance for internal reference		400	470	540	nF
SAMPLING DYNAMICS						
RES	Resolution		8		14	bit
F_{ADCONV}	Conversion rate			1.2 (14b) 1.6 (12b) 2.0 (10b) 2.66 (8b)		MS/s
TTRIG_C-START_REF0	Delay from cold start trigger to start of scan, Internal Ref				652	µs
TTRIG_C-START_REF1	Delay from cold start trigger to start of scan, External Ref				137	µs
TTRIG_W-START_LP1_REF0	Delay from warm start trigger to start of scan, LPMODE1, Internal Ref				65.6	µs
TTRIG_W-START_LP1_REF1	Delay from warm start trigger to start of scan, LPMODE1, External Ref				1.52	µs
TTRIG_WSTART_LP0_REF0	Delay from warm start trigger to start of scan, LPMODE0, Internal Ref				0	µs
TTRIG_W-START_LP0_REF1	Delay from warm start trigger to start of scan, LPMODE0, External Ref				0	µs
TSNGLSLOT_SCNCMP_PM14	Delay from scan start to scan complete, precision mode 14				40	cycles
TSNGLSLOT_SCNCMP_PM12	Delay from scan start to scan complete, precision mode 12				28	cycles
TSNGLSLOT_SCNCMP_PM10	Delay from scan start to scan complete, precision mode 10				22	cycles

Table 18-6: Analog-to-Digital Converter (ADC) *(Continued)*

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
TSNGLSLOT_SCNCMP_PM8	Delay from scan start to scan complete, precision mode 8				18	cycles
T_{CAL}	Calibration Period				6415	cycles
DYNAMIC CHARACTERISTICS, External 2v Reference **(LDO or Buck Mode[1], Single/Diff. Ended Input, 1 kHz Input, ADC Running in 14-bit Mode**						
$ENOB_{CAL}$	Calibrated ENOB	3.0V	10.3	10.9		ENOB
THD_{ADC}	Total harmonic distortion (THD) - 1st 7 harmonics	3.0V		-80.4	-71.7	dB
SNR_{ADC}	Signal-to-noise ratio (SNR)	3.0V	64.1	68		dB
$SFDR_{ADC}$	Spurious-free dynamic range (SFDR)	3.0V	78.55	85.9		dB
$SINAD_{ADC}$	Signal-to-noise and distortion ratio (SINAD)	3.0V	64.04	67.7		dB
DYNAMIC CHARACTERISTICS, Internal 1.5V Reference **(LDO Mode, Single/Diff. Ended Input, 1 kHz Input, ADC Running in 14-bit Mode)**						
$ENOB_{CAL}$	Calibrated ENOB	3.0V	10.2	10.6		ENOB
		1.8V	10.6	10.9		ENOB
THD_{ADC}	Total harmonic distortion (THD) - 1st 7 harmonics	3.0V		-70.4	-61	dB
		1.8V		-73.9	-65	dB
SNR_{ADC}	Signal-to-noise ratio (SNR)	3.0V		66.1		dB
		1.8V		67.5		dB
$SFDR_{ADC}$	Spurious-free dynamic range (SFDR)	3.0V	65.7	72.7		dB
		1.8V	70.7	76.1		dB
$SINAD_{ADC}$	Signal-to-noise and distortion ratio (SINAD)	3.0V	63.1	65.8		dB
		1.8V	65.7	67.2		dB
DYNAMIC CHARACTERISTICS, Internal 1.5V Reference **(Buck Mode, Single/Diff. Ended Input, 1 kHz Input, ADC Running in 14-bit Mode)**						
$ENOB_{CAL}$	Calibrated ENOB	3.0V	9.6	10.2		ENOB
		1.8V	9.4	10.1		ENOB
THD_{ADC}	Total harmonic distortion (THD) - 1st 7 harmonics	3.0V		-70	-60	dB
		1.8V		-74	-64	dB
SNR_{ADC}	Signal-to-noise ratio (SNR)	3.0V		63.4		dB
		1.8V		63.2		dB
$SFDR_{ADC}$	Spurious-free dynamic range (SFDR)	3.0V	65.6	72.7		dB
		1.8V	64.7	75.3		dB
$SINAD_{ADC}$	Signal-to-noise and distortion ratio (SINAD)	3.0V	59.8	63.1		dB
		1.8V	58.2	62.9		dB
PERFORMANCE						
NMC_{ADC}	No missing codes			14		bits
INL_{ADC}	Integral nonlinearity	Full input range		±2.4	±3.5	LSB
DNL_{ADC}	Differential nonlinearity	Full input range		±0.9	±1.7	LSB

Table 18-6: Analog-to-Digital Converter (ADC) *(Continued)*

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
E_{ADC_OFFEST}	Offset error		1		1	%FS
E_{ADC_GAIN}	Gain error				1	%FS
INTERNAL TEMPERATURE SENSOR						
E_{TEMP}	Temperature sensor accuracy	After calibration		±3		°C
S_{TEMP}	Temperature sensor slope			3.8		mV/ °C
BATTERY RESISTANCE						
R_{BATT}	Internal resistance for Battery Measurement		487.32	524	560.68	Ω
$V_{BATTDIV}$	Battery divider voltage		-2.5%	0.333 * V_{DDH}	+1.5%	V

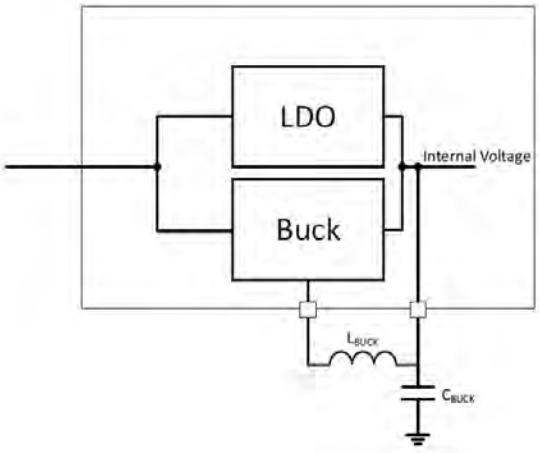
[1] Buck Mode not supported on wafer package.

18.7 Buck Converter[1]

Table 18-7: Buck Converter

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
L_{BUCK}	Buck converter inductance (V_{DDC}, V_{DDF})			2.2		µH
C_{BUCK}	Buck converter output capacitance (V_{DDC}, V_{DDF})			1		µF
V_{VDDC}	VDDC output voltage			695		mV
V_{VDDF}	VDDF output voltage			945		mV
C_{LOAD}	Load capacitance		0.8	1	1.2	µF

Figure 18-1: Block Diagram for the Voltage Regulator Module



[1] Buck Mode not supported on wafer package.

18.8 Power-On RESET (POR) and Brown-Out Detector (BOD)

Table 18-8: Power-On Reset (POR) and Brown-Out Detector (BOD)

Symbol	Parameter	Min	Typ	Max	Unit
V_{POR_RISING}	POR rising threshold voltage	1.62	1.72	1.755	V
$V_{POR\_FALLING}$	POR falling threshold voltage	1.62	1.69	1.754	V
V_{BODH_RISING}	BODH rising threshold voltage	2.27		2.42	v
$V_{BODH\_FALLING}$	Brown-out falling threshold voltage	2.12		2.28	v
T_{FALL_SLEW}	Maximum VDDH falling slew rate for BODL trigger			2.5	V/ms
T_{RESPR}	Response Time - Rise		10		µs
T_{RESPF}	Response Time - Fall			15	µs

18.9 Resets

Table 18-9: Resets

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
T_{RST}	nRST pulse width to guarantee reset assertion	2KV/s supply slew rate	400			ns
T_{POR}	POR detect to nRST deassertion delay	2KV/s supply slew rate	30	34	50	ms
$T_{POR2HRST}$	Delay from nRST deassertion to HRESET deassertion	2KV/s supply slew rate	12		24	ms
T_{RSTDLY}	nRST reset delay from internal BODL	2KV/s supply slew rate		12		µs
T_{SOFT}	Software initiated reset delay	2KV/s supply slew rate		11		µs

18.10 Voltage Comparator (VCOMP)

Table 18-10: Voltage Comparator (VCOMP)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
V_{COMPIN}	Input voltage range		0		V_{DDA}	V
V_{COMPIN_OV}	Input offset voltage			60		mV
I_{COMPIN_LEAK}	Input leakage current	GPIO 16 measured at 85°C, TTT part, 3.63 V; 3.63 V on pad; also at 1.8 V and 1.8 V	-	0.5	50	nA
T_{COMP_RTRIG}	Rising voltage trigger response time			3	30	µs
T_{COMP_FTRIG}	Falling voltage trigger response time			8	50	µs
V_{HYST}	Hysteresis		35		110	mV

18.11 Inter-Integrated Circuit (I^2C) Interface

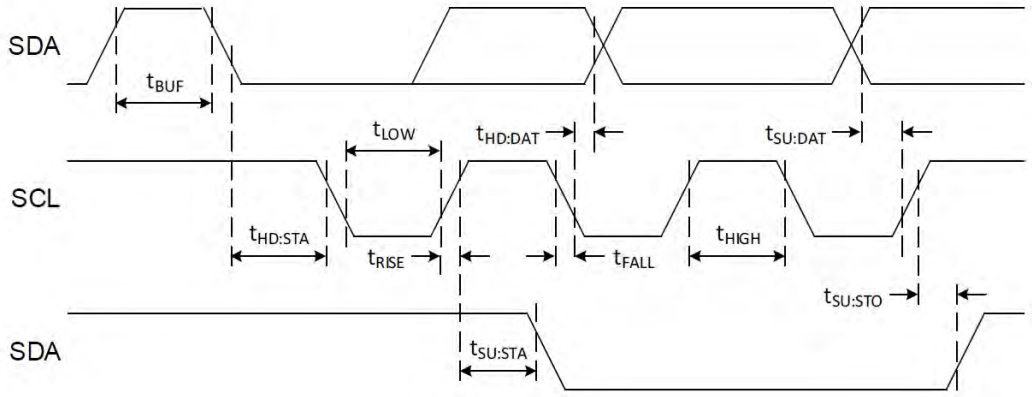
Table 18-11: Inter-Integrated Circuit (I^2C) Interface

Symbol	Parameter	Test Conditions	VCC	Min	Typ	Max	Unit
f_{SCL}	SCL input clock frequency		1.755 V - 3.63 V	10		1000	kHz
t_{LOW}	Low period of SCL clock	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.7			µs
t_{HIGH}	High period of SCL clock	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.0			µs
t_{RISE}	Rise time of SDA and SCL	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V			1.0	µs
t_{FALL}	Fall time of SDA and SCL	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V			300	ns
$t_{HD:STA}$	START condition hold time	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.0			µs
$t_{SU:STA}$	START condition setup time	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.7			µs
$t_{SU:DAT}$	SDA setup time	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	250			ns
$t_{HD:DAT}$	SDA hold time	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	0		3.45	µs
$t_{SU:STO}$	STOP condition setup time	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.0			µs
t_{BUF}	Bus free time before a new transmission	f_{SCL} = 100 kHz (Standard Mode)	1.755 V - 3.63 V	4.7			µs
t_{LOW}	Low period of SCL clock	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	1.3			µs
t_{HIGH}	High period of SCL clock	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	600			ns
t_{RISE}	Rise time of SDA and SCL	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V			300	ns
t_{FALL}	Fall time of SDA and SCL	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V			300	ns
$t_{HD:STA}$	START condition hold time	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	600			ns
$t_{SU:STA}$	START condition setup time	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	600			ns
$t_{SU:DAT}$	SDA setup time	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	100			ns
$t_{HD:DAT}$	SDA hold time	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	0		750	ns
$t_{SU:STO}$	STOP condition setup time	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	600			ns
t_{BUF}	Bus free time before a new transmission	f_{SCL} = 400 kHz (Fast Mode)	1.755 V - 3.63 V	1.3			µs

Table 18-11: Inter-Integrated Circuit (I^2C) Interface *(Continued)*

Symbol	Parameter	Test Conditions	VCC	Min	Typ	Max	Unit
t_{LOW}	Low period of SCL clock	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	500			µs
t_{HIGH}	High period of SCL clock	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	270			ns
t_{RISE}	Rise time of SDA and SCL	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V			120	ns
t_{FALL}	Fall time of SDA and SCL	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V			120	ns
$t_{HD:STA}$	START condition hold time	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	250			ns
$t_{SU:STA}$	START condition setup time	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	250			ns
$t_{SU:DAT}$	SDA setup time	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	40			ns
$t_{HD:DAT}$	SDA hold time	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	0		300	ns
$t_{SU:STO}$	STOP condition setup time	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	250			ns
t_{BUF}	Bus free time before a new transmission	f_{SCL} = 1 MHz (Fast Mode Plus)	1.755 V - 3.63 V	600			ns

Figure 18-2: I^2C Timing



18.12 Serial Peripheral Interface (SPI) Master Interface (IOM1, 2, 3 and 5)

Table 18-12: Serial Peripheral Interface (SPI) Master Interface

Symbol	Parameter	Conditions	Min	Typ	Max
F_{SCLK}	SCLK frequency range			8	12
B_{FIFO}	FIFO size			128	
T_{SCLK_LO}	Clock low time	F_{SCLK} = 12 MHz	31	1/2 FSCLK	
T_{SCLK_HI}	Clock high time	F_{SCLK} = 12 MHz	31	1/2 FSCLK	
T_{SU_MI}	MISO input data setup time		15		
T_{HD_MI}	MISO input data hold time		10		
T_{HD_MO}	MOSI output data hold time	VDDH = 3.6 V	11		
$T_{VALID,MO}$	MOSI output data valid time	VDDH = 1.755 V			32

Figure 18-3: SPI Master Mode, Phase = 0

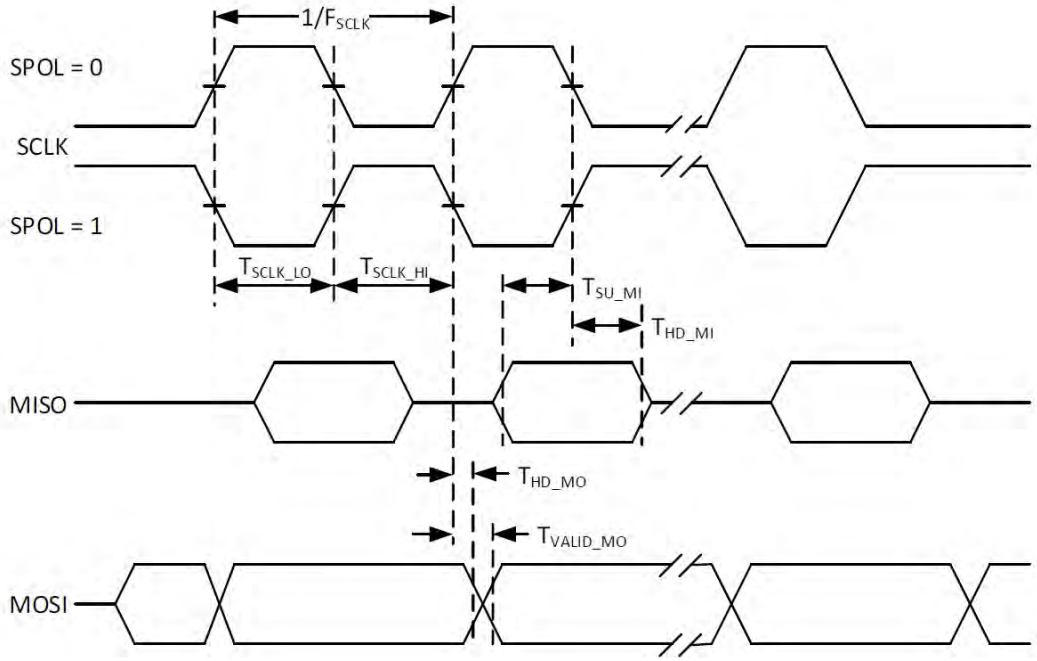
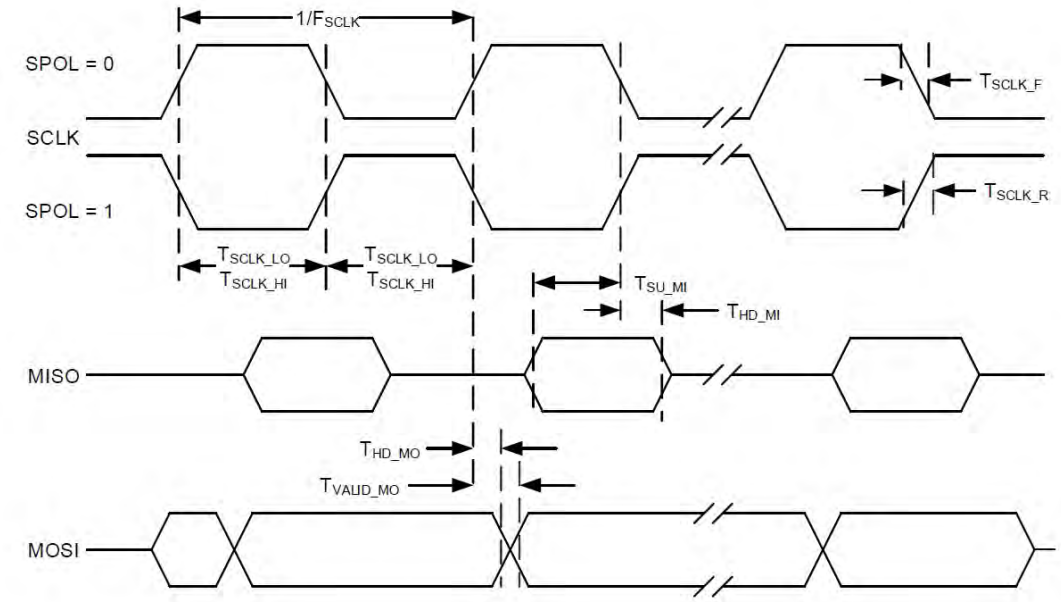


Figure 18-4: SPI Master Mode, Phase = 1



18.13 High Speed Serial Peripheral Interface (SPI) Master Interface (IOM 0, 4)

Table 18-13: High Speed Serial Peripheral Interface (SPI) Master Interface

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range			12	24	MHz
B_{FIFO}	FIFO size			128		Bytes
T_{SCLK_LO}	Clock low time	F_{SCLK} = 24 MHz	19	1/2 FSCLK		ns
T_{SCLK_HI}	Clock high time	F_{SCLK} = 24 MHz	19	1/2 FSCLK		s
T_{SU_MI}	MISO input data setup time		7			ns
T_{HD_MI}	MISO input data hold time		2			ns
T_{HD_MO}	MOSI output data hold time		3			ns
$T_{VALID,MO}$	MOSI output data valid time	VDDH = 1.755			6	ns

Figure 18-5: SPI Master Mode, Phase = 0

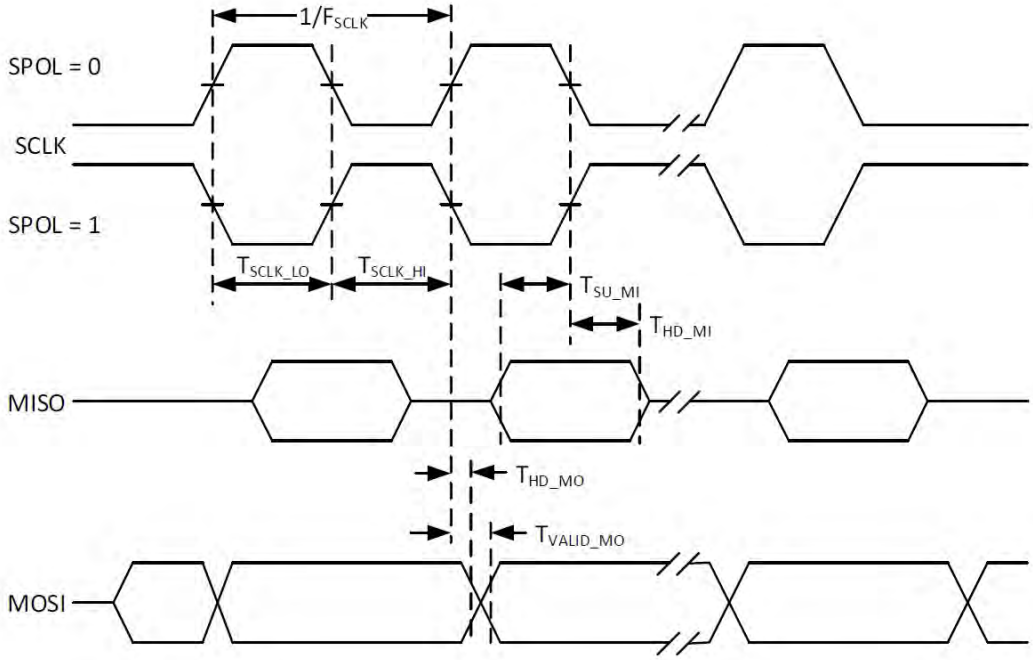
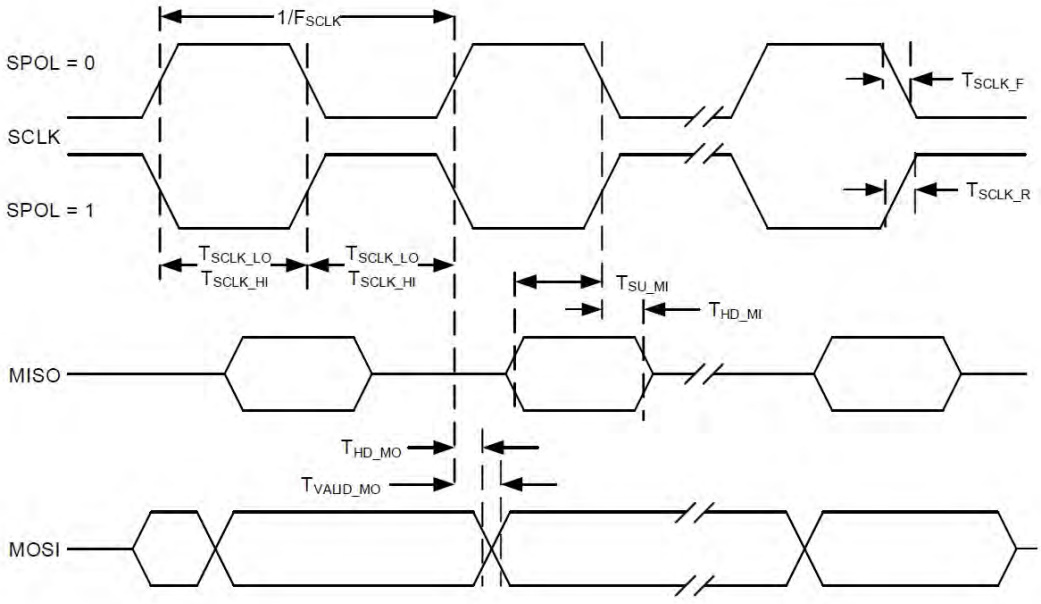


Figure 18-6: SPI Master Mode, Phase = 1



18.14 Serial Peripheral Interface (SPI) Slave Interface

Table 18-14: Serial Peripheral Interface (SPI) Slave Interface

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range				8	MHz
B_{FIFO}	FIFO size			128		Bytes
T_{SCLK_LO}	Clock low time	F_{SCLK} = 8 MHz	35			ns
T_{SCLK_HI}	Clock high time	F_{SCLK} = 8 MHz	35			ns
T_{CE_LEAD}	Chip enable low to first SCLK edge		60			ns
T_{CE_LAG}	Chip enable high to last SCLK edge		60			ns
T_{CE_SDO}	Chip enable low to MISO data output		60			ns
T_{CE_SDZ}	Chip enable high to MISO data tri-state				40	ns
T_{SU_SI}	MOSI input data setup time		6			ns
T_{HD_SI}	MOSI input data hold time		6			ns
T_{HD_SO}	MISO output data hold time		10			ns
$T_{VALID,SO}$	MISO output data valid time				35	ns

Figure 18-7: SPI Slave Mode, Phase = 0

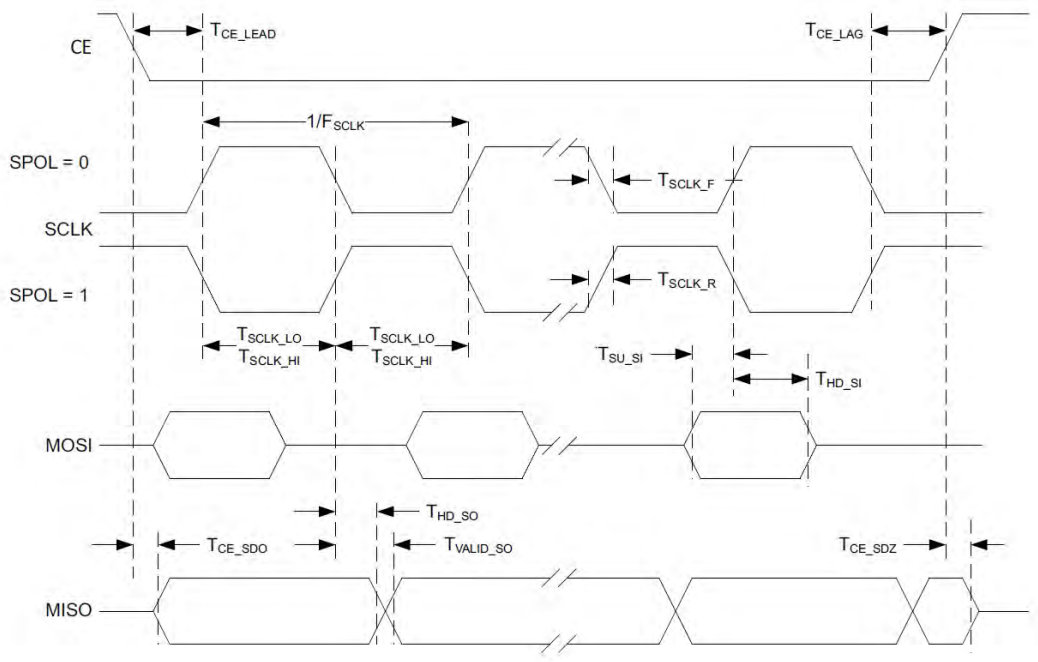
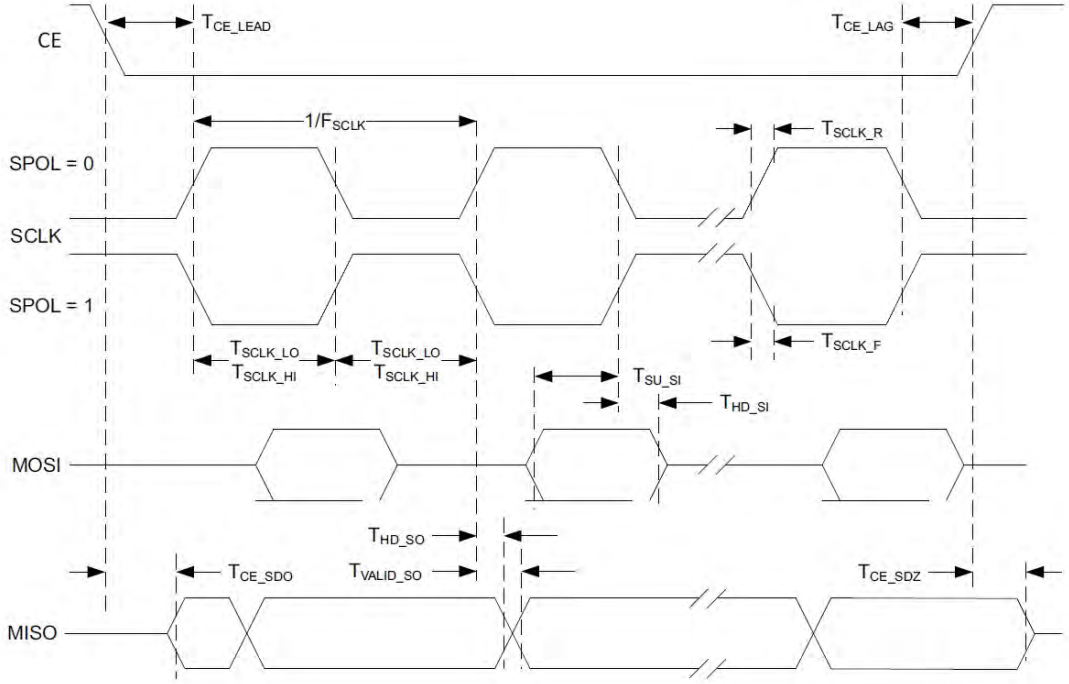


Figure 18-8: SPI Slave Mode, Phase = 1



18.15 PDM Interface

Table 18-15: Pulse Density Modulation (PDM) Interface

Symbol	Parameter	Conditions	VDD	Min	Typ	Max	Unit
F_{PDM_CLK}	PDM output clock frequency		1.755V - 3.63V	0.187		3.072	MHz
DC_{PDMCLK}	PDM clock duty cycle[1]	20pF load, 12mA/4mA drive strength, GPIO12/ GPIO37	1.755V - 3.63V	40		55	%
J_{PDM_CLK}	PDM clock jitter	20pF load, 12mA drive strength, GPIO12, 3MHz clock freq	1.755V - 3.63V		131	300	ps
T_{PDM_RISE}	PDM clock rise time - 12 mA Drive Strength	20pF load, 12mA drive strength, GPIO12	3.3V		4.6		ns
			1.8V		5.3		ns
T_{PDM_RISE}	PDM clock rise time - 4 mA Drive Strength	20pF load, 4mA drive strength, GPIO30 & GPIO37	3.3V		5.2		ns
			1.8V		6.4		ns
T_{PDM_FALL}	PDM clock fall time - 12 mA Drive Strength	20pF load, 12mA drive strength, GPIO12	3.3V		4.7		ns
			1.8V		5.4		ns
T_{PDM_FALL}	PDM clock fall time - 4 mA Drive Strength	20pF load, 4mA drive strength, GPIO30 & GPIO37	3.3V		5.6		ns
			1.8V		7.3		ns

Table 18-15: Pulse Density Modulation (PDM) Interface *(Continued)*

Symbol	Parameter	Conditions	VDD	Min	Typ	Max	Unit
T_{SU_PDM}	PDM input data setup time	F_{PDM_CLK} = 3.072 MHz	1.755V - 3.63V			35	ns
T_{HD_PDM}	PDM input data hold time	F_{PDM_CLK} = 3.072 MHz	1.755V - 3.63V	35			ns

[1] Applicable when F_{PDMCLK} <= 2.4 MHz and PDM_PCFG_MCLKDIV set to MCKDIV1, MCKDIV2 or MCKDIV4 only. PDM_PCFG_MCLKDIV setting of MCKDIV3 has a duty cycle of 67%.

18.16 I²S Interface

Table 18-16: Inter-Integrated Serial (I^2S) Interface

Symbol	Parameter	Conditions	VDD	Min	Typ	Max	Unit
F_{BCLK}	I^2S input BCLK frequency range		1.755V - 3.63V	0.187		12	MHz
F_{WDCLK}	I^2S input WDCLK frequency range		1.755V - 3.63V	11.7		750	kHz
DC_{BCLK}	I^2S BCLK duty cycle		1.755V - 3.63V	40		60	%
DC_{WDCLK}	I^2S WDCLK duty cycle		1.755V - 3.63V	40		60	%
T_{I2S_RISE}	I^2S clock and data rise time	20pF load, 4mA drive strength, GPIO30	3.3V		5.4		ns
			1.8V		6.8		ns
T_{I2S_FALL}	I^2S clock and data fall time	20pF load, 4mA drive strength, GPIO30	3.3V		5.6		ns
			1.8V		7.6		ns
T_{SU_I2S}	I^2S input data setup time		1.755V - 3.63V			40	ns
T_{HD_I2S}	I^2S input data hold time		1.755V - 3.63V	40			ns

18.17 Universal Asynchronous Receiver/Transmitter (UART)

Table 18-17: Universal Asynchronous Receiver/Transmitter (UART)

Symbol	Parameter	Min	Typ	Max	Unit
F_{BAUD}	UART baud rate		-	921600	bps

18.18 Counter/Timer (CTIMER)

Table 18-18: Counter/Timer (CTIMER)

Symbol	Parameter	Min	Typ	Max	Unit
F_{CTIMER}	External clock frequency			12	MHz
T_{CTIMER}	External clock pulse width	40			ns

18.19 System Timer (STIMER)

Table 18-19: System Timer (STIMER)

Symbol	Parameter	Min	Typ	Max	Unit
F_{STIMER}	Input frequency			F_{HFRC}/16	MHz
T_{STIMER}	Capture pulse width to trigger an interrupt	50			ns

18.20 Watchdog Timer (WDT)

Table 18-20: Watchdog Timer (WDT)

Symbol	Parameter	Min	Typ	Max	Unit
T_{WD}	Watchdog timer resolution	0.0625	128	128	Hz

18.21 Flash Memory

Table 18-21: Flash Memory

Symbol	Parameter	Min	Typ	Max	Unit
PE_{CYC}	Program/erase cycles before failure	10,000	-	-	cycles
T_{FDR}	Data retention @85°C	10	-	-	years
T_{PAGE_ERASE}	Single page erase time (8192 bytes)	10	15	20	ms
T_{MASS_ERASE}	Mass erase time	10	15	20	ms
T_{PAGE_PGM}	Single page program time (8192 bytes)	23	28	33	ms

18.22 General Purpose Input/Output (GPIO)

All GPIOs have Schmitt trigger inputs.

Table 18-22: General Purpose Input/Output (GPIO)

Symbol	Parameter	Test Conditions	VDD	Min	Typ	Max	Unit
ALL GPIOs							
V_{OH}	High-level output voltage			0.8 * V_{DDH}			V
V_{OL}	Low-level output voltage					0.2 * V_{DDH}	V
V_{IH}	Positive going input threshold voltage			0.7 * V_{DDH}			V
V_{IL}	Negative going input threshold voltage					0.3 * V_{DDH}	V
V_{HYS}	Input Hysteresis			0.1 * V_{DDH}			V
C_{GPI}	Input capacitance	Pad input capacitance standard GPIO	1.755V - 3.63V		2.5		pF
R_{PU}	Pull-up resistance			8	35	150	kΩ
R_{PD}	Pull-down resistance			9	20	150	kΩ
$R_{PUI2C00}$	I^2C pad pull-up resistance, RSEL = 0x00			1.2	1.4	1.6	kΩ
$R_{PUI2C01}$	I^2C pad pull-up resistance, RSEL = 0x01			4	4.5	6	kΩ
$R_{PUI2C10}$	I^2C pad pull-up resistance, RSEL = 0x10			8	10	12	kΩ
$R_{PUI2C11}$	I^2C pad pull-up resistance, RSEL = 0x11			14	22	32	kΩ
I_{IN}	Input pin leakage current	GPIO0 measured at 25°C	1.755V - 3.63V		0.13	50	nA
I_{INOD}	Open drain output leakage current	GPIO0 measured at 25°C, TTT part, 3.3 V VDD, 3.3 V pad voltage, GPIO configured as open drain, WTA set to 1	1.755V - 3.63V		0.1	50	nA
STANDARD GPIOs							
T_{RISE_STD}	Rise time	20pF load, 2mA drive strength, GPIO39	3.3V		7.35		ns
			1.8V		14.5		ns
T_{RISE_STD}	Rise time	20pF load, 4mA drive strength, GPIO39	3.3V		2.56		ns
			1.8V		6.4		ns
T_{RISE_STD}	Rise time	20pF load, 8mA drive strength, GPIO39	3.3V		1.66		ns
			1.8V		2.4		ns

Table 18-22: General Purpose Input/Output (GPIO) *(Continued)*

Symbol	Parameter	Test Conditions	VDD	Min	Typ	Max	Unit
T_{RISE_STD}	Rise time	20pF load, 12mA drive strength, GPIO39	3.3V		1.3		ns
			1.8V		1.8		ns
T_{FALL_STD}	Fall time	20pF load, 2mA drive strength, GPIO39	3.3V		10.0		ns
			1.8V		18.1		ns
T_{FALL_STD}	Fall time	20pF load, 4mA drive strength, GPIO39	3.3V		3.61		ns
			1.8V		7.6		ns
T_{FALL_STD}	Fall time	20pF load, 8mA drive strength, GPIO39	3.3V		1.85		ns
			1.8V		3.1		ns
T_{FALL_STD}	Fall time	20pF load, 12mA drive strength, GPIO39	3.3V		1.37		ns
			1.8V		2.1		ns
I_{SRC_STD}	Output source current, 2mA drive strength register setting	Current load required to lower GPIO voltage to V_{OH} applied from GPIO pin to GNDP	3.3V		3.5		mA
			1.8V		1.1		mA
I_{SNK_STD}	Output sink current, 2mA drive strength register setting	Current load required to raise GPIO voltage to V_{OL} applied from VDDH to the GPIO pin	3.3V		3.4		mA
			1.8V		1.3		mA
I_{SRC_STD}	Output source current, 4mA drive strength register setting	Current load required to lower GPIO voltage to V_{OH} applied from GPIO pin to GNDP	3.3V		7.1		mA
			1.8V		2.5		mA
I_{SNK_STD}	Output sink current, 4mA drive strength register setting	Current load required to raise GPIO voltage to V_{OL} applied from VDDH to the GPIO pin	3.3V		6.8		mA
			1.8V		2.5		mA
I_{SRC_STD}	Output source current, 8mA drive strength register setting	Current load required to lower GPIO voltage to V_{OH} applied from GPIO pin to GNDP	3.3V		14.1		mA
			1.8V		4.3		mA
I_{SNK_STD}	Output sink current, 8mA drive strength register setting	Current load required to raise GPIO voltage to V_{OL} applied from VDDH to the GPIO pin	3.3V		13.5		mA
			1.8V		4.5		mA
I_{SRC_STD}	Output source current, 12mA drive strength register setting	Current load required to lower GPIO voltage to VOH applied from GPIO pin to GNDP	3.3V		21.1		mA
			1.8V		6.5		mA
I_{SNK_STD}	Output sink current, 12mA drive strength register setting	Current load required to raise GPIO voltage to VOL applied from VDDH to the GPIO pin	3.3V		20.2		mA
			1.8V		6.5		mA
POWER SWITCH GPIOs							
R_{SRC_PWR}	High side power switch resistance				1.5	2.4	Ω
I_{SRC_PWR}	High side power switch source current					100	mA

Table 18-22: General Purpose Input/Output (GPIO) *(Continued)*

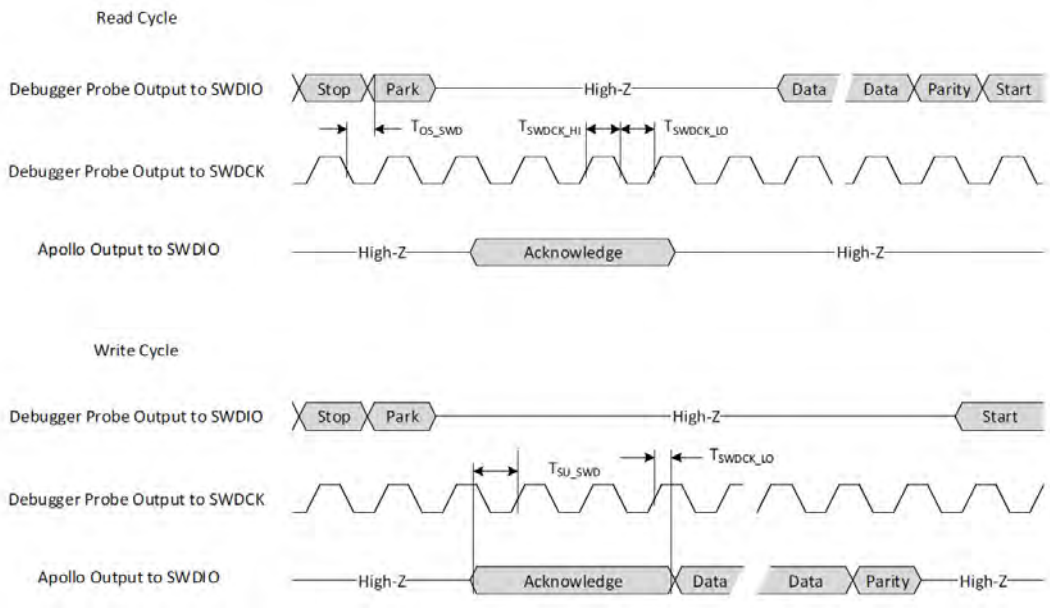
Symbol	Parameter	Test Conditions	VDD	Min	Typ	Max	Unit
$I_{SRC\_PWR\_LKG}$	High side power switch source leakage current	GPIO4 measured at 85°C, TTT part, 3.63 V; 3.63 V on pad; PC_SEL = 1	1.755V - 3.63V		1.3	200	nA
R_{SNK_PWR}	Low side power switch resistance				1.8	2.4	Ω
I_{SNK_PWR}	Low side power switch sink current					100	mA
$I_{SNK\_PWR\_LKG}$	Low side power switch source leakage current	GPIO22 measured at 85°C, TTT part, 3.63 V; 0 V on pad; PC_SEL = 1 (TYP at 3.3 V w/ 0 V on pad)			1.7	200	nA

18.23 Serial Wire Debug (SWD)

Table 18-23: Serial Wire Debug (SWD)

Symbol	Parameter	Min	Typ	Max	Unit
T_{SWDCK_HI}	SWDCK clock high period	0.1		500	µs
T_{SWDCK_LO}	SWDCK clock low period	0.1		500	µs
T_{OS_SWD}	SWDIO output skew to falling edge of SWDCLK	0		45	ns
T_{SU_SWD}	Input setup time between SWDIO and rising edge SWDCK	65			ns
T_{HD_SWD}	Input hold time between SWDIO and rising edge SWDCK	125			ns

Figure 18-9: Serial Wire Debug Timing



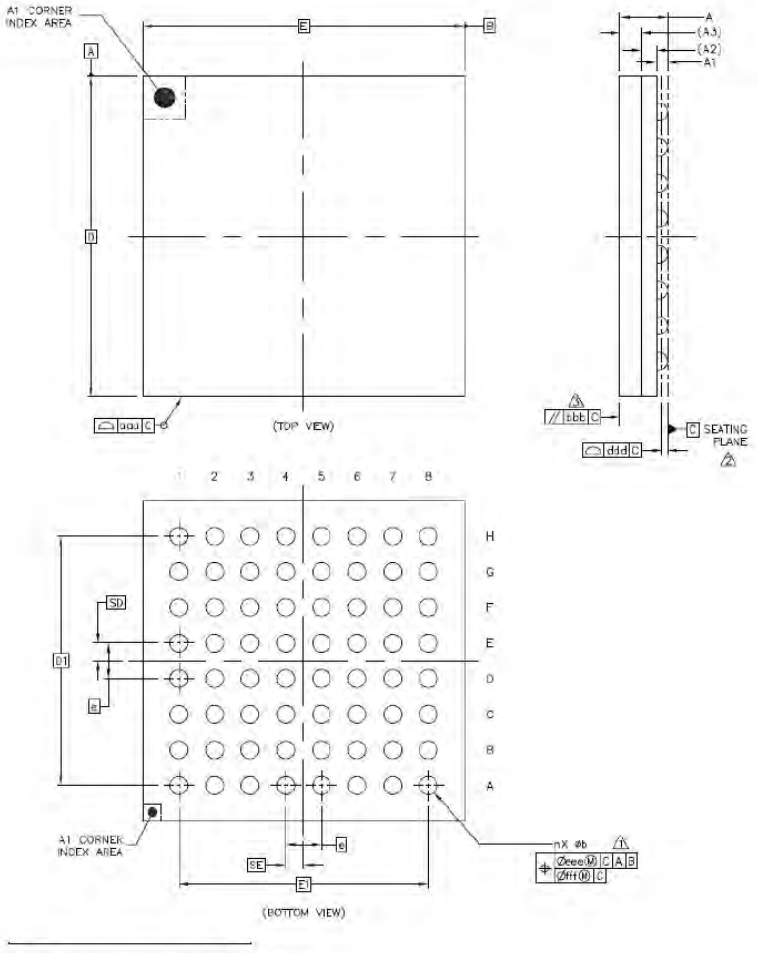
SECTION

19

Package Mechanical Information

19.1 BGA Package[1]

Figure 19-1: BGA Package Drawing



[1] All dimensions in mm unless otherwise noted.

Figure 19-2: BGA Package Drawing - Notes

Drawing Notes:
Dimension B is measured at the maximum solder ball diameter , parallel to datum plane c
Datum C (seating plane) is defined by the spherical crowns of solder balls.
Parallelism measurement shall exclude effect of mark on top surface of package.

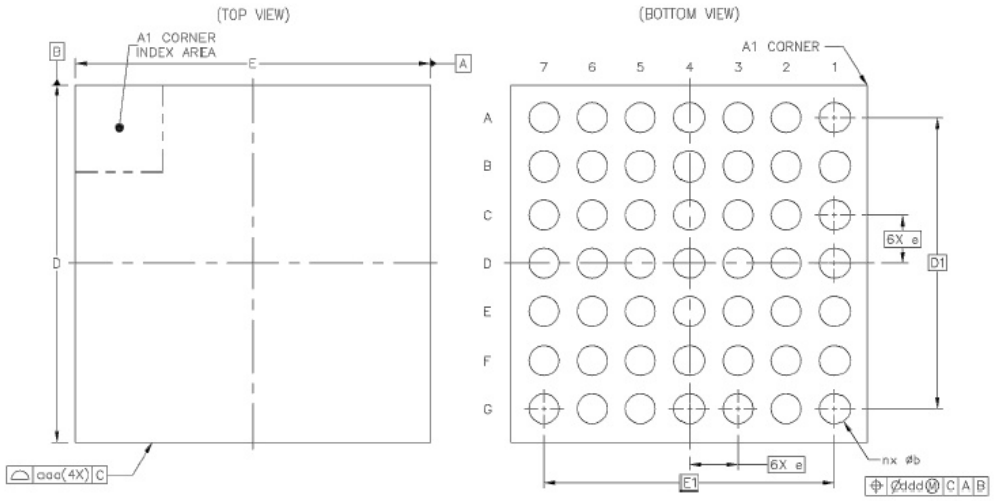
	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	0.8
STAND OFF	A1	0.11	---	0.21
SUBSTRATE THICKNESS	A2		0.21	REF
MOLD THICKNESS	A3		0.3	REF
BODY SIZE	D		4.5	BSC
	E		4.5	BSC
BALL DIAMETER			0.25	
BALL OPENING			0.25	
BALL WIDTH	b	0.2	---	0.3
BALL PITCH	e		0.5	BSC
BALL COUNT	n		84	
EDGE BALL CENTER TO CENTER	D1		3.5	BSC
	E1		3.5	BSC
BODY CENTER TO CONTACT BALL	SD		0.25	BSC
	SE		0.25	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.2	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

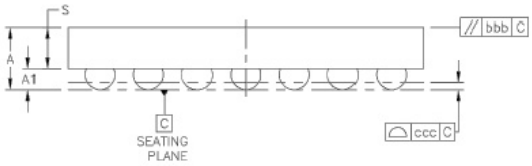
19.1.1 PCB Land Pattern and Solder Stencil

For the BGA package, there should be a 1:1 ratio between the diameter of the PCB pad and the diameter of the BGA ball. If required by routing constraints, the ratio of the PCB pad to the BGA ball may be 0.8:1. The solder stencil should leave a solder mask opening that allows for 0.0762 mm clearance around the PCB pad (diameter 0.1524 mm larger than the PCB pad).

19.2 CSP Package[1]

Figure 19-3: BGA Package Drawing - Notes





		SYMBOL	COMMON DIMENSIONS
TOTAL THICKNESS		A	0.452 ± 0.055
WAFER THICKNESS		S	0.3 ± 0.025
STAND OFF		A1	0.122 ~ 0.182
FILM THICKNESS		A2	--- ± ---
BODY SIZE	X	E	2.557
	Y	D	2.587
BALL/BUMP PITCH	X	SE	--- BSC
	Y	SD	--- BSC
EDGE BALL CENTER TO CENTER	X	E1	2.098 BSC
	Y	D1	2.098 BSC
PITCH		e	0.35 BSC
BALL DIAMETER (SIZE)			0.2
BALL/BUMP WIDTH		b	0.188 ~ 0.248
BALL/BUMP COUNT		n	49
PACKAGE EDGE TOLERANCE		aaa	0.03
WAFER FLATNESS		bbb	0.06
COPLANARITY		ccc	0.05
BALL/BUMP OFFSET (PACKAGE)		ddd	0.015

[1] The Apollo2 CSP package is sensitive to light incident on either the backside or edges of the die. Light exposure can result in increased current and erratic behavior which may include system crash or reset. To prevent exposure of the die to light, the recommendation is to apply an opaque epoxy coating (or similar) over the Apollo2 CSP unless the SoC will be in an enclosure that blocks all light.

19.2.1 PCB Land Pattern and Solder Stencil

Figure 19-4: Land Pattern for CSP Package

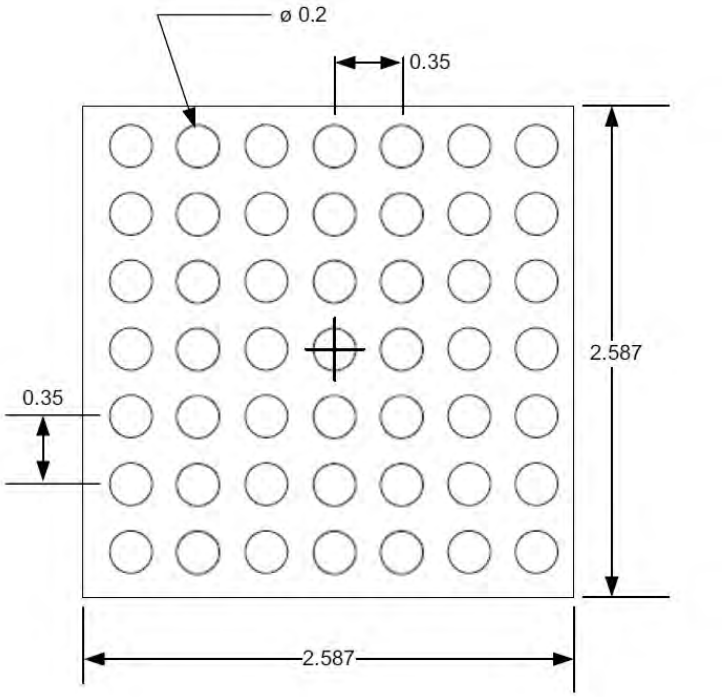
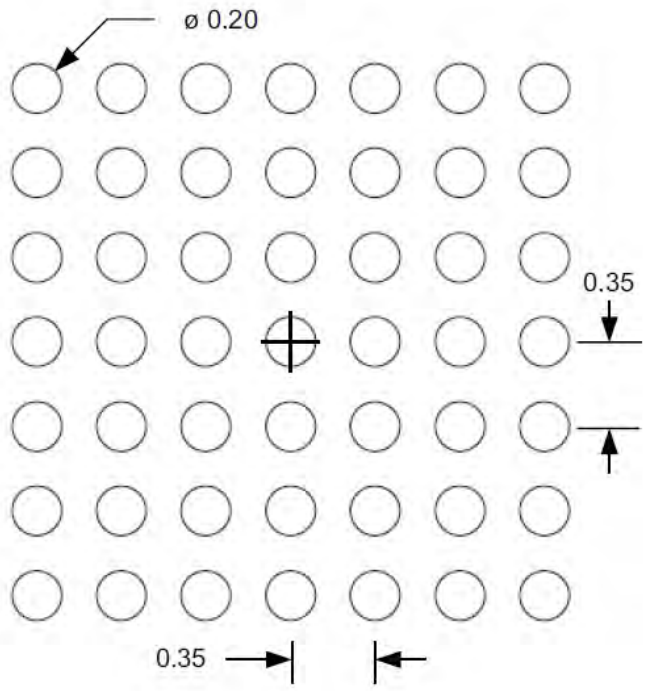


Figure 19-5: Example Solder Stencil Pattern for CSP Package



19.3 Apollo2 Thin WLCSP Package

Figure 19-6: Apollo2 Thin WLCSP Drawing - Top View

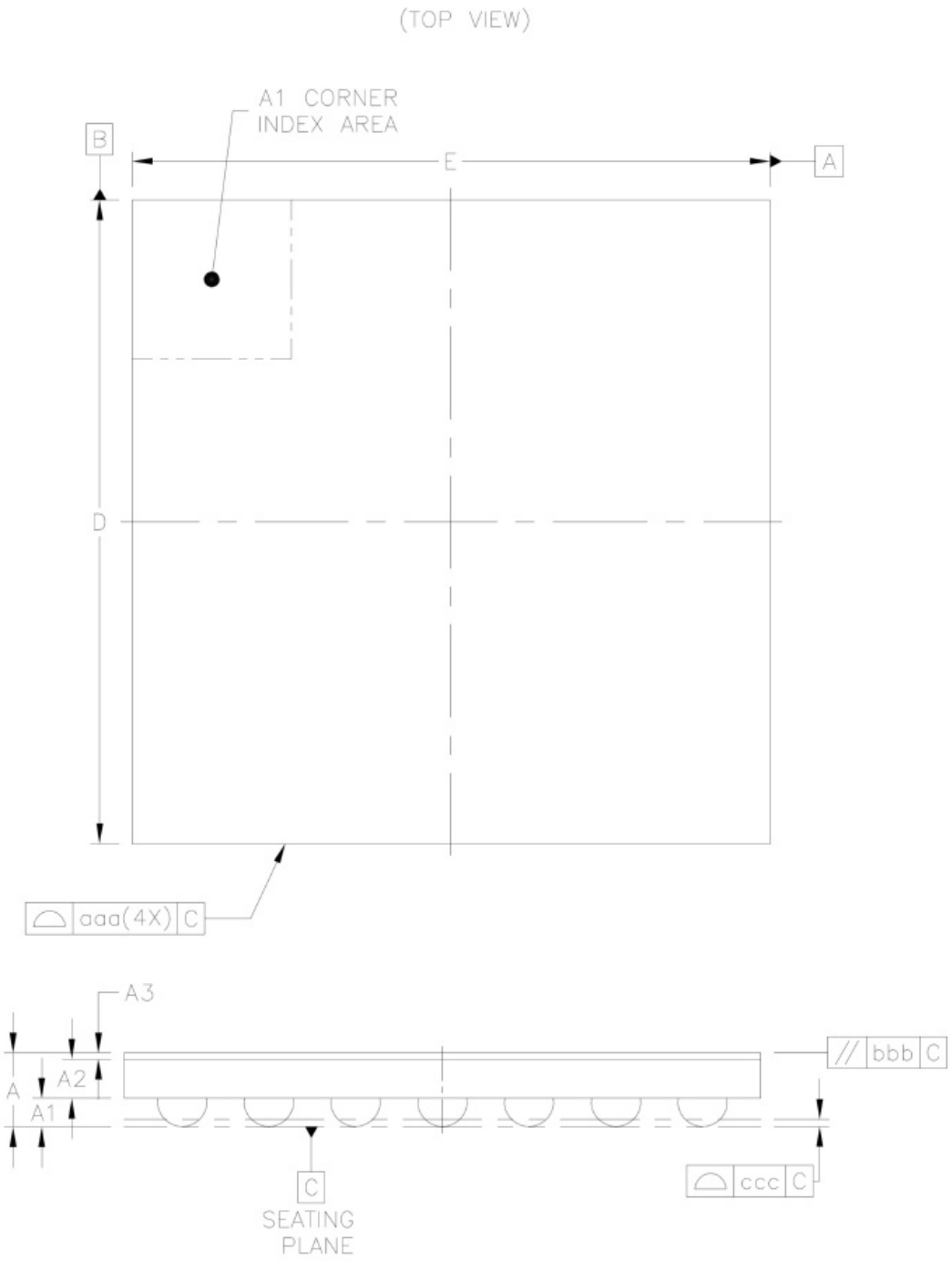


Figure 19-7: Apollo2 Thin WLCSP Drawing - Bottom View

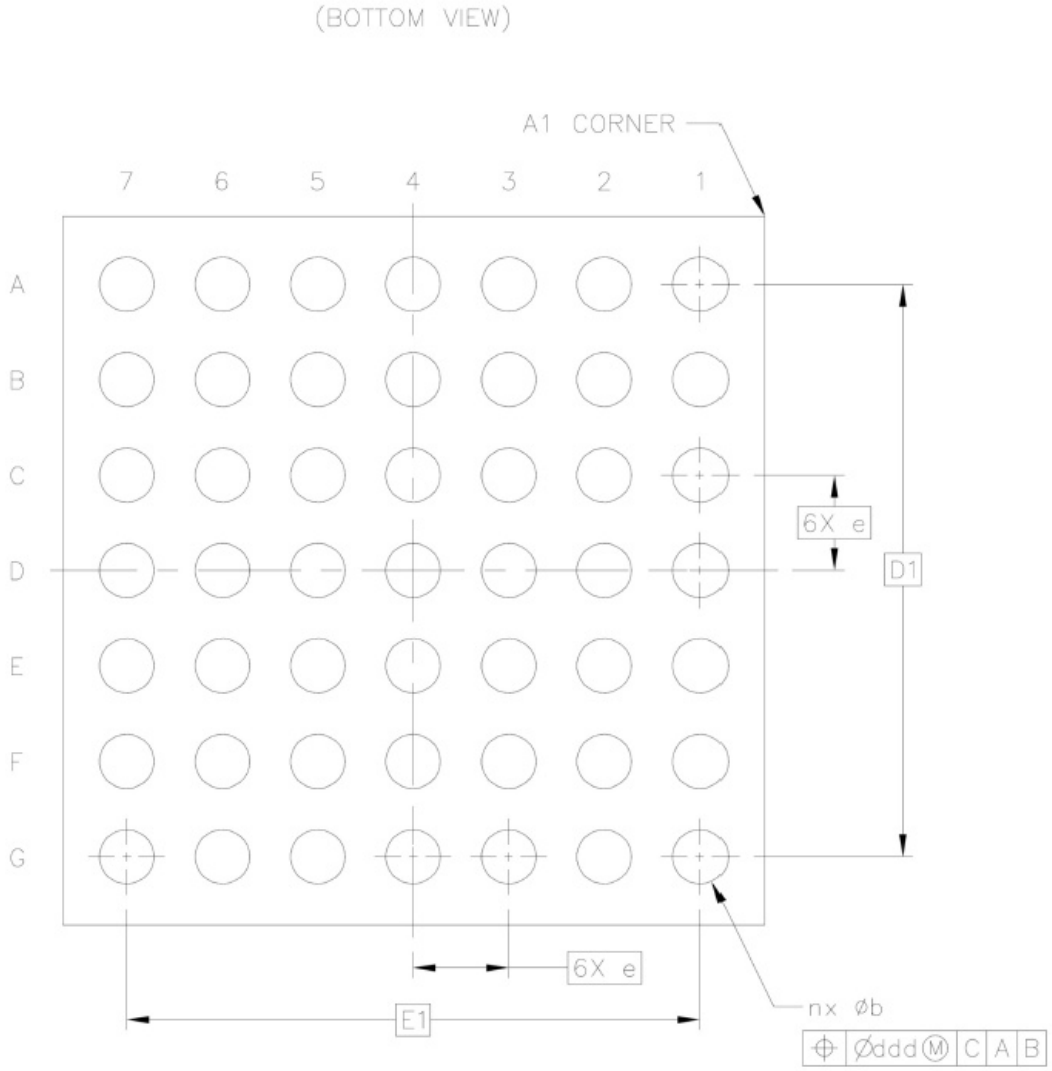
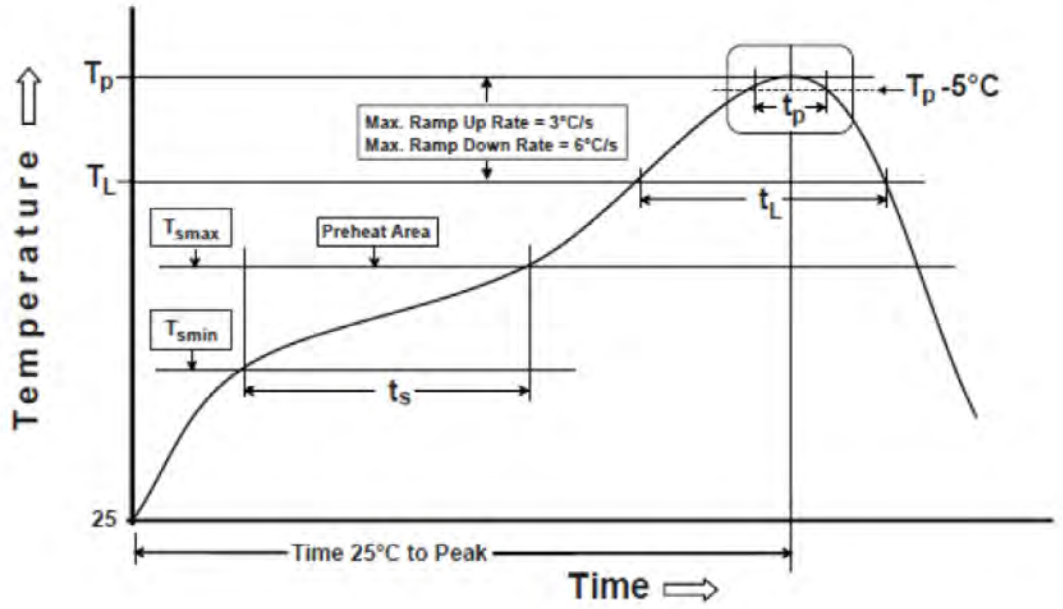


Figure 19-8: Apollo2 Thin WLCSP Drawing - Notes

		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.27	0.3	0.33
STAND OFF		A1	0.098	---	0.132
WAFER THICKNESS		A2	0.135	0.16	0.185
FILM THICKNESS		A3	0.022	0.025	0.028
BODY SIZE	X	E	2.565		
	Y	D	2.595		
BALL/BUMP PITCH	X	SE	---		BSC
	Y	SD	---		BSC
EDGE BALL CENTER TO CENTER	X	E1	2.098		BSC
	Y	D1	2.098		BSC
PITCH		e	0.35		BSC
BALL DIAMETER (SIZE)			0.17		
BALL/BUMP WIDTH		b	0.172	---	0.232
BALL/BUMP COUNT		n	49		
PACKAGE EDGE TOLERANCE		aaa	0.03		
WAFER FLATNESS		bbb	0.06		
COPLANARITY		ccc	0.03		
BALL/BUMP OFFSET (PACKAGE)		ddd	0.015		

19.4 Reflow Profile

Figure 19-9: Reflow Soldering Diagram



SECTION

20

Flash OTP 0 Customer Info Space (Info0)

20.1 Flash OTP INSTANCE0 INFO0 Words

Ambiq Internal Information Block 0 of Instance 0.

INSTANCE 0 BASE ADDRESS:0x00000000

This is the detailed description of the contents of INFO block 0 of FLASH instance 0, also known as the Customer Data, for the Apollo2 SoC.

20.1.1 Register Memory Map

Table 20-1: Flash OTP INSTANCE0 INFO0 Register Map

Address(es)	Registered Name	Description
0x00000000	SIGNATURE0	INFO0 Signature
0x00000004	SIGNATURE1	INFO0 Signature
0x00000008	SIGNATURE2	INFO0 Signature
0x0000000C	SIGNATURE3	INFO0 Signature
0x00000010	SECURITY	Security protection bits
0x00000014	BUCKTRIM	Buck trim values used by software
0x00000020	WRITEPROTECT0	Flash write-protection bits.
0x00000024	WRITEPROTECT1	Flash write-protection bits.
0x00000030	COPYPROTECT0	Flash copy/read-protection bits.
0x00000034	COPYPROTECT1	Flash copy/read-protection bits.

20.1.2 Flash OTP INSTANCE0 INFO0 Words

20.1.2.1 SIGNATURE0 Register

INFO0 Signature
OFFSET: 0x00000000
INSTANCE 0 ADDRESS: 0x00000000
Word 0 (low word, bits 31:0) of the 128-bit INFO0 signature.

Table 20-2: SIGNATURE0 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SIG0																															

Table 20-3: SIGNATURE0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG0	0xffffffff		INFO0 signature word 0 (low 32-bits).

20.1.2.2 SIGNATURE1 Register

INFO0 Signature
OFFSET: 0x00000004
INSTANCE 0 ADDRESS: 0x00000004
Word 1 (bits 63:32) of the 128-bit INFO0 signature.

Table 20-4: SIGNATURE1 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SIG1																															

Table 20-5: SIGNATURE1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG1	0xffffffff		INFO0 signature word 1.

20.1.2.3 SIGNATURE2 Register

INFO0 Signature
OFFSET: 0x00000008
INSTANCE 0 ADDRESS: 0x00000008
Word 2 (bits 95:64) of the 128-bit INFO0 signature.

Table 20-6: SIGNATURE2 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SIG2																															

Table 20-7: SIGNATURE2 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG2	0xffffffff		INFO0 signature word 2.

20.1.2.4 SIGNATURE3 Register

INFO0 Signature
OFFSET: 0x0000000C
INSTANCE 0 ADDRESS: 0x0000000C
Word 3 (high word, bits 127:96) of the 128-bit INFO0 signature.

Table 20-8: SIGNATURE3 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SIG3																															

Table 20-9: SIGNATURE3 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG3	0xffffffff		INFO0 signature word 3 (high 32 bits).

20.1.2.5 SECURITY Register

Security protection bits
OFFSET: 0x00000010
INSTANCE 0 ADDRESS: 0x00000010
This 32-bit word contains the customer programmable security.

Table 20-10: SECURITY Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD																						BOOTLOADER_AT_RESET	EN_CUST_INFO_ERASE	EN_CUST_INFO_PROG					FLASH_WIPE	SRAM_WIPE	SWO_CTRL	DEBUG_PROT

Table 20-11: SECURITY Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x3fffff		Reserved.
9	BOOTLOADER_AT_RESET	0x1		Enable bootloader action at reset. When 1, bootloader spins in an infinite while loop. When 0, bootloader can go to deep sleep.
8	EN_CUST_INFO_ERASE	0x1		Enable customer INFO space erasing. When set to 0, customer INFO space is protected and cannot be erased.
7:4	EN_CUST_INFO_PROG	0xf		Enable customer INFO space programming. When set to 0, customer INFO space is protected and cannot be programmed. Each bit protects a 2KB region of the 8KB info page.
3	FLASH_WIPE	0x1		Flash wipe. When 0, both flash memory arrays will be erased when a debugger is attached to the system.
2	SRAM_WIPE	0x1		SRAM wipe. When 0, the SRAM memory arrays will be cleared when a debugger is attached to the system.
1	SWO_CTRL	0x1		SWO Control. When 0, the SWO (ITM output) is disabled.
0	DEBUG_PROT	0x1		Debugger protection. When 0, the SWCLK/SWDIO debugger interface logic is disabled.

20.1.2.6 BUCKTRIM Register

Buck trim values used by software
OFFSET: 0x00000014
INSTANCE 0 ADDRESS: 0x00000014
Software values for buck trimming

Table 20-12: BUCKTRIM Register Bits

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
RSVD	BUCK_MEM_TRIM	RSVD	BUCK_CORE_TRIM

Table 20-13: BUCKTRIM Register Bits

Bit	Name	Reset	RW	Description
31:26	RSVD	0x3f		Reserved
25:16	BUCK_MEM_TRIM	0x3ff		Buck Mem Trim value
15:10	RSVD	0x3f		Reserved
9:0	BUCK_CORE_TRIM	0x3ff		Buck Core Trim value

20.1.2.7 WRITEPROTECT0 Register

Flash write-protection bits.
OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x00000020
These bits write-protect flash in 16KB chunks.

Table 20-14: WRITEPROTECT0 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CHUNKS																															

Table 20-15: WRITEPROTECT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Write protect flash 0x00000000 - 0x0007FFFF. Each bit provides write protection for 16KB chunks of flash data space.

20.1.2.8 WRITEPROTECT1 Register

Flash write-protection bits.
OFFSET: 0x00000024
INSTANCE 0 ADDRESS: 0x00000024
These bits write-protect flash in 16KB chunks.

Table 20-16: WRITEPROTECT1 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CHUNKS																															

Table 20-17: WRITEPROTECT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Write protect flash 0x00080000 - 0x000FFFFF. Each bit provides write protection for 16KB chunks of flash data space.

20.1.2.9 COPYPROTECT0 Register

Flash copy/read-protection bits.
OFFSET: 0x00000030
INSTANCE 0 ADDRESS: 0x00000030
These bits read-protect flash in 16KB chunks.

Table 20-18: COPYPROTECT0 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CHUNKS																															

Table 20-19: COPYPROTECT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Copy (read) protect flash 0x00000000 - 0x0007FFFF. Each bit provides read protection for 16KB chunks of flash.

20.1.2.10 COPYPROTECT1 Register

Flash copy/read-protection bits.
OFFSET: 0x00000034
INSTANCE 0 ADDRESS: 0x00000034
These bits read-protect flash in 16KB chunks.

Table 20-20: COPYPROTECT1 Register Bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CHUNKS																															

Table 20-21: COPYPROTECT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Copy (read) protect flash 0x00080000 - 0x000FFFFF. Each bit provides read protection for 16KB chunks of flash.

SECTION

21

Ordering Information

Table 21-1: Ordering Information

Orderable Part Number	Flash	RAM	Package[1]	Packing[2]	Temperature Range[3]
AMAPH1KK-KCR	1MB	256KB	49-pin WLCSP	Tape and Reset	-40°C to +85°C
AMAPH1KK-KBR	1MB	256KB	64-pin BGA	Tape and Reset	-40°C to +85°C
AMAPH1KK-KCR-TB	1MB	256KB	49-pin WLCSP	Tape and Reset	-40°C to +85°C

[1] Compliant and certified with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in raw homogeneous materials. The package was designed to be soldered at high temperatures (per reflow profile) and can be used in specified lead-free processes.

[2] For AMAPH1KK-KCR, minimum order quantity of 5,000 pieces.For AMAPH1KK-KBR, minimum order quantity of 4,000 pieces.

[3] Temperature Range is guaranteed -40 to +85°C by design. Current production test limits are -20 to +60°C.

6500 River Place Boulevard, Building 7, Suite 200, Austin, TX 78730
www.ambiq.com
sales@ambiq.com
+1 (512) 879-2850

A-SOCAP2-DSGA01EN v1.3
March 2023