Apollo3 Blue Datasheet

Doc. ID: DS-A3-0p9p1

Revision 0.9.1

Feb 2019

IMPORTANT NOTICE:

This datasheet includes content which is accurate to the extent possible, but is preliminary and certain content may not be fully validated.

Features

Ultra-low supply current:
- 6 µA/MHz executing from flash at 3.3 V
- 6 µA/MHz executing from RAM at 3.3 V
- 1 µA deep sleep mode (BLE Off) with RTC at 3.3 V

High-performance ARM Cortex-M4 Processor
- 48 MHz nominal clock frequency, with 96 MHz high performance “TurboSPOT” Burst Mode
- Floating point unit
- Memory protection unit
- Wake-up interrupt controller with 32 interrupts

Integrated Bluetooth[1] 5 low-energy module
- RF sensitivity: -93 dBm (typical)
- TX: 3 mA @ 0 dBm, RX: 3 mA
- Tx peak output power: 4.0 dBm (max)

Ultra-low power memory:
- Up to 1 MB of flash memory for code/data
- Up to 384 KB of low leakage RAM for code/data
- 16 kB 2-way Associative/Direct-Mapped Cache

Ultra-low power interface for off-chip sensors:
- 14 bit ADC at up to 1.2 MS/s, 15 selectable input channels available
- Voltage Comparator
- Temperature sensor with +/-3ºCaccuracy

ISO7816 Secure interface

Flexible serial peripherals:
- 1x 2/4/8-bit SPI master interface
- 6x I^2C/SPI masters for peripheral communication
- I^2C/SPI slave for host communications
- 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 – 48/96 MHz
- RTC based on Ambiq’s AM08X5/18X5 families

Wide operating range: 1.755-3.63 V, –40 to 85°C

1. The *Bluetooth*® word mark and logos are registered trademarks owned by the Bluetooth SIG, Inc. and any use of such marks is under license. Other trademarks and trade names are those of their respective owners.

Compact package option:
- 3.37 x 3.25 mm(<0.35mm thk pkg) 66-pin CSP with 37 GPIO
- 5 x 5 mm (<0.5mm thk pkg) 81-pin BGA with 50 GPIO

Applications

- Always-listening keyword detect for voice assistants
- Always-listening local audio command recognition for smart home, IoT, and industrial devices
- Hearable devices including bluetooth headsets, earbuds, and truly wireless earbuds
- Hearing aids
- Remote and Gaming Controls
- Digital Health Monitoring and Sensing Devices
- Wearable electronics including smart watches and fitness/activity trackers
- Home Automation, Security and Lighting control applications

Description

The Apollo3 MCU Family is an ultra-low power, highly integrated microcontroller designed for battery-powered and portable, mobile devices. At the heart of the Apollo3 Blue MCU is Ambiq Micro’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.

By combining ultra-low power sensor conversion electronics with the powerful ARM Cortex-M4 processor with Floating Point Unit, the Apollo3 Blue MCU enables complex sensor processing tasks to be completed with unprecedented battery life. Weeks, months, and years of battery life are achievable while doing complex context detection, gesture recognition, and activity monitoring.

The Apollo3 Blue MCU builds upon the industry’s best power management efficiency of the Apollo2 MCU Family to deliver a totally integrated Bluetooth low energy connectivity solution. The device offers Bluetooth 5 ready radio and integrates 1 MB of flash memory and 384 KB of RAM to accommodate radio and sensor overhead while leaving space for application code. This microcontroller also includes several masters and one slave SPI and I2C ports and two UART ports for communicating with sensors including accelerometers, gyroscopes, and magnetometers.

Typical Application Circuit for the Apollo3 MCU

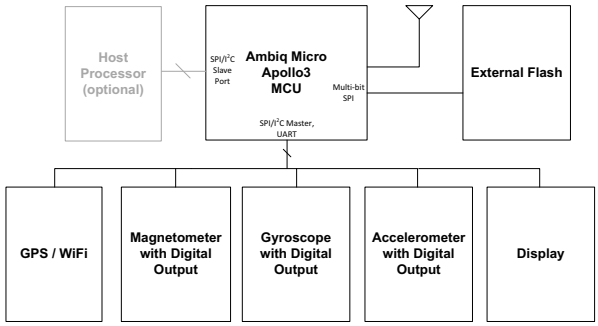


Table of Content

List of Figures

List of Tables

1. Apollo3 Blue MCU Package Pins

1.1 Pin Configuration

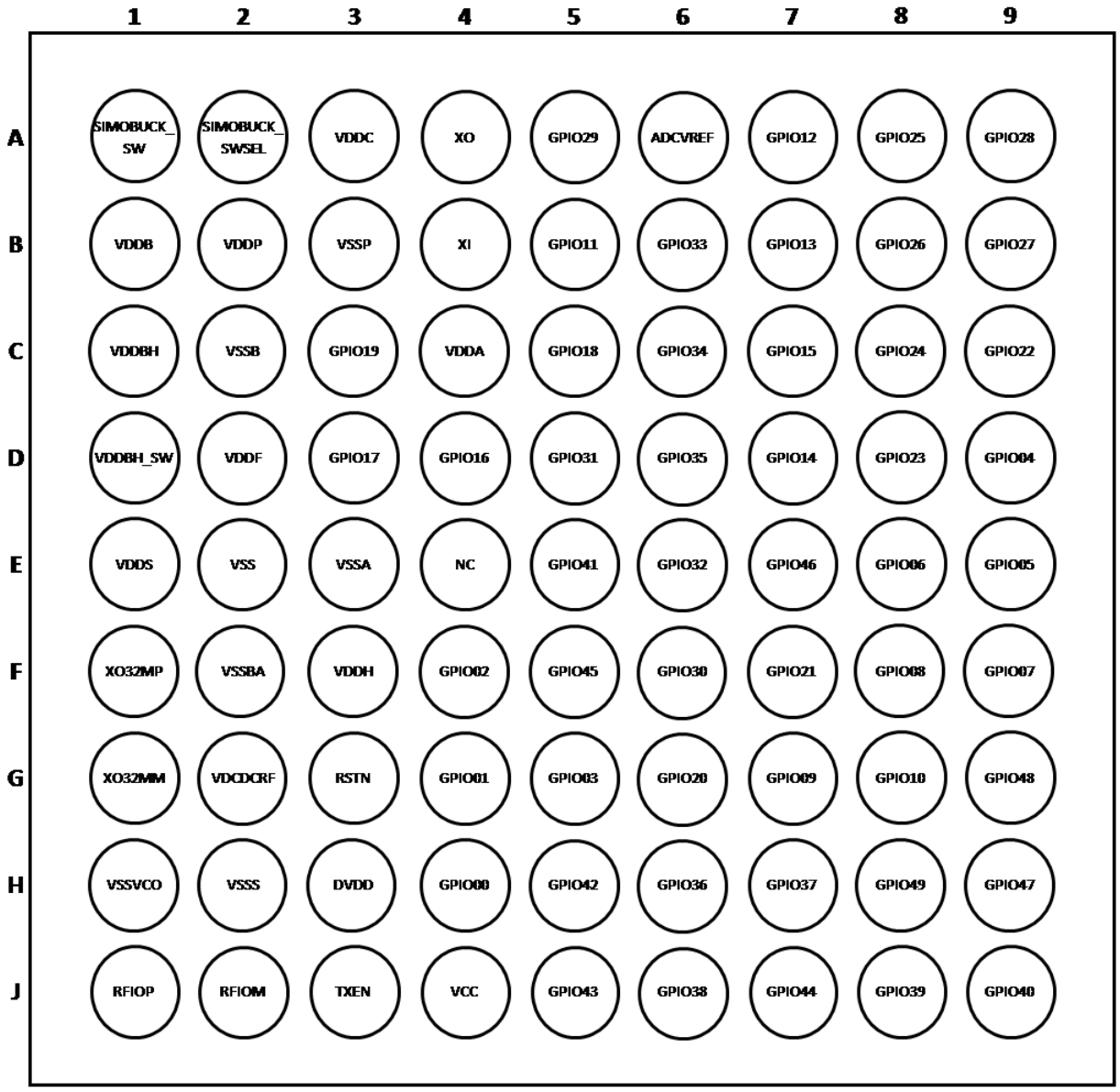


Figure 1. Apollo3 Blue MCU BGA Pin Configuration Diagram

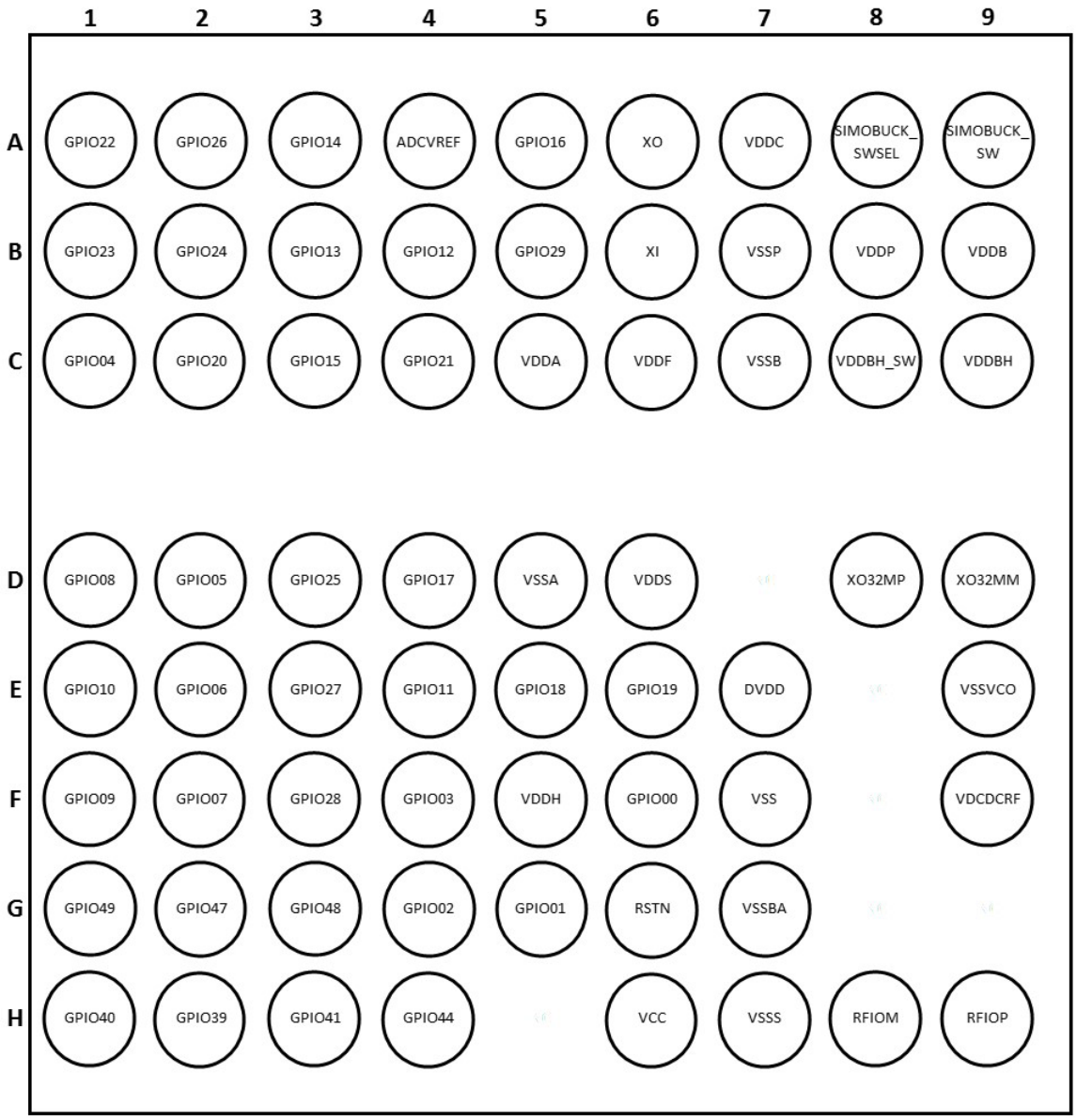


Figure 2. Apollo3 Blue MCU CSP Pin Configuration Diagram - Top View

1.2 Pin Connections

The following table lists the external pins of the Apollo3 Blue MCU and their available functions.

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
POWER						
B2	B8	-	-	VDDP	VDD Supply for SIMO Buck Converter	Power
B1	B9	-	-	VDDB	VDD Supply for BLE/Burst Buck Converter	Power
F3	F5	-	-	VDDH	VDD Supply for I/O Pads	Power
C4	C5	-	-	VDDA	Analog Voltage Supply	Power
J4	H6	-	-	VCC	RF Voltage Supply	Power
E1	D6	-	-	VDDS	High Voltage Digital Supply	Power
G2	F9	-	-	VDCDCRF	RF Voltage Supply	Power
B3	B7	-	-	VSSP	Ground Connection for SIMO Buck Converter	Ground
E3	D5	-	-	VSSA	Ground for Analog Supply	Ground
C2	C7	-	-	VSSB	Ground Connection for BLE/Burst Buck Converter	Ground
E2	F7	-	-	VSS	Ground for Digital	Ground
F2	G7	-	-	VSSBA	Ground for BLE Analog Supply	Ground
H1	E9	-	-	VSSVCO	Ground for BLE VCO Supply	Ground
H2	H7	-	-	VSSS	Ground for BLE RF Supply	Ground
A6	A4	-	-	ADCVREF	Analog to Digital Converter Reference Voltage	Analog
H3	E7	-	-	DVDD	Deoupling Cap for BLE digital supply	Power
E4	-	-	-	NC	No Connect	
BUCK						
A3	A7	-	-	VDDC	SIMO Buck Converter Voltage Core Output Supply	Power
D2	C6	-	-	VDDF	SIMO Buck Converter Voltage Flash/Memory Output Supply	Power
A1	A9	-	-	SIMO-BUCK_SW	SIMO Buck Converter Inductor Switch Output	Power
A2	A8	-	-	SIMO-BUCK_SWSEL	SIMO Buck Converter Inductor Switch Input	Power
D1	C8	-	-	VDDBH_SW	BLE/Burst Buck Converter Inductor Switch	Power
C1	C9	-	-	VDDBH	BLE/Burst Buck Converter Voltage Output Supply	Power
OSCILLATOR						
A4	A6	-	-	XO	32.768 kHz Crystal Output	XT
B4	B6	-	-	XI	32.768 kHz Crystal Input	XT
G1	D9	-	-	XO32MM	32 MHz Crystal Input	XT
F1	D8	-	-	XO32MP	32 MHz Crystal Output	XT
RESET						

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G3	G6	-	-	RSTN	External Reset Input	Input/ Output
RF						
J2	H8	-	-	RFIOM	RF I/O Negative	Analog
J1	H9	-	-	RFIOP	RF I/O Positive	Analog
J3	-	-	-	TXEN	Transmitter Enable	Output
GPIO						
H4	F6	0	0	SLSCL	I^2C Slave Clock	Input
			1	SLSCK	SPI Slave Clock	Input
			2	CLKOUT	Programmable Output Clock	Output
			3	GPIO00	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	MSPI4	MSPI Master Interface Signal 4 See “MSPI Connection” on page 393.	Input/ Output
			6	RSV	Reserved	
			7	NCE0	IO Master N Chip Select 0 Table 564, “NCE Encoding Table,” on page 382	Output
G4	G5	1	0	SLSDAWIR3	I^2C Slave I/O Data SPI Master 3 3 Wire Data	Bidirec-tional Open Drain
			1	SLMOSI	SPI Slave Input Data	Input
			2	UART0TX	UART0 Transmit	Output
			3	GPIO01	General Purpose I/O	Input/ Output
			4	RSV	Reserved	Input
			5	MSPI5	MSPI Master Interface Signal 5 See “MSPI Connection” on page 396.	Input/ Output
			6	RSV	Reserved	
			7	NCE1	IO Master N Chip Select 1 Table 564, “NCE Encoding Table,” on page 382	Output
F4	G4	2	0	UART1RX	UART1 Receive	Input
			1	SLMISO	SPI Slave Output Data	Output
			2	UART0RX	UART0 Receive	Input
			3	GPIO02	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	MSPI6	MSPI Master Interface Signal 6 See “MSPI Connection” on page 393.	Input/ Output
			6	RSV	Reserved	
			7	NCE2	IO Master N Chip Select 2 Table 564, “NCE Encoding Table,” on page 382	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G5	F4	3	0	UA0RTS	UART0 Request To Send (RTS)	Output
			1	SLnCE	SPI Slave Chip Enable	Input
			2	NCE3	IO Master N Chip Select 3 Table 564, “NCE Encoding Table,” on page 382	Output
			3	GPIO03	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	MSPI7	MSPI Master Interface Signal 7 See “MSPI Connection” on page 393.	Input/ Output
			6	TRIG1	ADC Trigger Input	Input
			7	I2SWCLK	I2S Word Clock	Input
D9	C1	4	0	UA0CTS	UART0 Clear To Send (CTS)	Input
			1	SLINT	Configurable Slave Interrupt	Output
			2	NCE4	IO Master N Chip Select 4 Table 564, “NCE Encoding Table,” on page 382	Output
			3	GPIO04	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	UART1RX	UART1 Receive	Input
			6	CT17	Timer/Counter Interface Signal 17 See “Implementing Counter/Timer Connections” on page 394.	Output
			7	MSPI2	MSPI Master Interface Signal 2 See “MSPI Connection” on page 393.	Input/ Output
E9	D2	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	GPIO05	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	RSV	Reserved	
			6	RSV	Reserved	
			7	CT8	Timer/Counter Interface Signal 8 See “Implementing Counter/Timer Connections” on page 397.	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E8	E2	6	0	M0SDAWIR3	I^2C Master 0 Data SPI Master 0 3 Wire Data	Bidirec-tional Open Drain
			1	M0MISO	SPI Master 0 Input Data	Input
			2	UA0CTS	UART0 Clear To Send (CTS)	Input
			3	GPIO06	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	CT10	Timer/Counter Interface Signal 10 See “Implementing Counter/Timer Connections” on page 394.	Output
			6	RSV	Reserved	
			7	I2SDAT	I2S Data	Output
F9	F2	7	0	NCE7	IO Master N Chip Select 7 Table 564, “NCE Encoding Table,” on page 382	Output
			1	M0MOSI	SPI Master 0 Output Data	Output
			2	CLKOUT	Programmable Output Clock	Output
			3	GPIO07	General Purpose I/O	Input/ Output
			4	TRIG0	ADC Trigger Input	Input
			5	UART0TX	UART0 Transmit	Output
			6	RSV	Reserved	
			7	CT19	Timer/Counter Interface Signal 19 See “Implementing Counter/Timer Connections” on page 397.	Output
F8	D1	8	0	M1SCL	I^2C Master 1 Clock	Open Drain Output
			1	M1SCK	SPI Master 1 Clock	Output
			2	NCE8	IO Master N Chip Select 8 Table 564, “NCE Encoding Table,” on page 382	Output
			3	GPIO08	General Purpose I/O	Input/ Output
			4	SCCCLK	Secure Card Controller Clock	Output
			5	RSV	Reserved	
			6	UART1TX	UART1 Transmit	Output
			7	RSV	Reserved	

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G7	F1	9	0	M1SDAWIR3	I^2C Master 1 Data SPI Master 1 3 Wire Data	Bidirectional Open Drain
			1	M1MISO	SPI Master 1 Input Data	Input
			2	NCE9	IO Master N Chip Select 9 Table 564, “NCE Encoding Table,” on page 382	Output
			3	GPIO09	General Purpose I/O	Input/ Output
			4	SCCIO	Secure Card Controller I/O	Input/ Output
			5	RSV	Reserved	
			6	UART1RX	UART1 Receive	Input
			7	RSV	Reserved	
G8	E1	10	0	UART1TX	UART1 Transmit	Output
			1	M1MOSI	SPI Master 1 Output Data	Output
			2	NCE10	IO Master N Chip Select 10 Table 564, “NCE Encoding Table,” on page 382	Output
			3	GPIO10	General Purpose I/O	Input/ Output
			4	PDMCLK	PDM Clock Output	Output
			5	UA1RTS	UART1 Request To Send	Output
			6	RSV	Reserved	
			7	RSV	Reserved	
B5	E4	11	0	ADCSE2	Analog to Digital Converter Single-Ended Input 2	Input
			1	NCE11	IO Master N Chip Select 11 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT31	Timer/Counter Interface Signal 31 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO11	General Purpose I/O	Input/ Output
			4	SLINT	Configurable Slave Interrupt	Output
			5	UA1CTS	UART1 Clear To Send	Input
			6	UART0RX	UART0 Receive	Input
			7	PDMDATA	PDM Data	Input

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A7	B4	12	0	ADCD0NSE9	Analog to Digital Converter Differential N Input 0 / Single-Ended Input 9	Input
			1	NCE12	IO Master N Chip Select 12 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT0	Timer/Counter Interface Signal 0 See “Implementing Counter/Timer Connections” on page 397.	Output
			3	GPIO12	General Purpose I/O	Input/ Output
			4	SLnCE	SPI Slave Chip Enable	Input
			5	PDMCLK	PDM Clock Output	Output
			6	UA0CTS	UART0 Clear To Send (CTS)	Input
			7	UART1TX	UART1 Transmit	Output
B7	B3	13	0	ADCD0PSE8	Analog to Digital Converter Differential P Input 0 / Single-Ended Input 9	Input
			1	NCE13	IO Master N Chip Select 13 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT2	Timer/Counter Interface Signal 2 See “Implementing Counter/Timer Connections” on page 397.	Output
			3	GPIO13	General Purpose I/O	Input/ Output
			4	I2SBCLK	I2S Bit Clock	Input
			5	RSV	Reserved	
			6	UA0RTS	UART0 Request To Send (RTS)	Output
			7	UART1RX	UART1 Receive	Input
D7	A3	14	0	ADCD1P	Analog to Digital Converter Differential P Input 1	Input
			1	NCE14	IO Master N Chip Select 14 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UART1TX	UART1 Transmit	Output
			3	GPIO14	General Purpose I/O	Input/ Output
			4	PDMCLK	PDM Output Clock	Output
			5	RSV	Reserved	
			6	SWDCK	Serial Wire Debug Clock	Input
			7	32kHzXT	32kHz Clock	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C7	C3	15	0	ADCD1N	Analog to Digital Converter Differential N Input 1	Input
			1	NCE15	IO Master N Chip Select 15 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UART1RX	UART1 Receive	Input
			3	GPIO15	General Purpose I/O	Input/ Output
			4	PDMDATA	PDM Data	Input
			5	RSV	Reserved	
			6	SWDIO	Serial Wire Debug I/O	Bidirec-tional 3-state
			7	SWO	Serial Wire Debug Output	Output
D4	A5	16	0	ADCSE0	Analog to Digital Converter Single-Ended Input 0	Input
			1	NCE16	IO Master N Chip Select 16 Table 564, “NCE Encoding Table,” on page 382	Output
			2	TRIG0	ADC Trigger Input 0	Input
			3	GPIO16	General Purpose I/O	Input/ Output
			4	SCCRST	Secure Card Controller Reset	Output
			5	CMPIN0	Voltage Comparator Input 0	Input
			6	UART0TX	UART0 Transmit	Output
			7	UA1RTS	UART1 Request To Send (RTS)	Output
D3	D4	17	0	CMPRF1	Voltage Comparator Reference 1	Input
			1	NCE17	IO Master N Chip Select 17 Table 564, “NCE Encoding Table,” on page 382	Output
			2	TRIG3	ADC Trigger Input 3	Input
			3	GPIO17	General Purpose I/O	Input/ Output
			4	SCCCLK	Secure Card Controller Clock	Output
			5	RSV	Reserved	
			6	UART0RX	UART0 Receive	Input
			7	UA1CTS	UART1 Clear To Send (CTS)	Input
C5	E5	18	0	CMPIN1	Voltage Comparator Input 1	Input
			1	NCE18	IO Master N Chip Select 18 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT4	Timer/Counter Interface Signal 4 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO18	General Purpose I/O	Input/ Output
			4	UA0RTS	UART0 Request To Send	Output
			5	RSV	Reserved	
			6	UART1TX	UART1 Transmit	Output
			7	SCCIO	Secure Card Controller I/O	Input/ Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C3	E6	19	0	CMPRF0	Comparator Reference 0	Input
			1	NCE19	IO Master N Chip Select 19 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT6	Timer/Counter Interface Signal 6 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO19	General Purpose I/O	Input/ Output
			4	SCCCLK	Secure Card Controller Clock	Output
			5	RSV	Reserved	
			6	UART1RX	UART1 Receive	Input
			7	I2SBCLK	I2S Bit Clock	Input
G6	C2	20	0	SWDCK	Software Debug Clock	Input
			1	NCE20	IO Master N Chip Select 20 Table 564, “NCE Encoding Table,” on page 382	Output
			2	RSV	Reserved	
			3	GPIO20	General Purpose I/O	Input/ Output
			4	UART0TX	UART0 Transmit	Output
			5	UART1TX	UART1 Transmit	Output
			6	I2SBCLK	I2S Bit Clock	Input
			7	UA1RTS	UART1 Request To Send (RTS)	Output
F7	C4	21	0	SWDIO	Software Data I/O	Bidirec-tional 3-state
			1	NCE21	IO Master N Chip Select 21 Table 564, “NCE Encoding Table,” on page 382	Output
			2	RSV	Reserved	Output
			3	GPIO21	General Purpose I/O	Input/ Output
			4	UART0RX	UART0 Receive	Input
			5	UART1RX	UART1 Receive	Input
			6	SCCRST	Secure Card Controller Reset	Output
			7	UA1CTS	UART1 Clear To Send (CTS)	Input

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C9	A1	22	0	UART0TX	UART0 Transmit	Output
			1	NCE22	IO Master N Chip Select 22 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT12	Timer/Counter Interface Signal 12 See “Implementing Counter/Timer Connections” on page 397.	Output
			3	GPIO22	General Purpose I/O	Input/ Output
			4	PDMCLK	PDM Output Clock	Output
			5	RSV	Reserved	
			6	MSPI0	MSPI Master Interface Signal 0 See “MSPI Connection” on page 393.	Input/ Output
			7	SWO	Serial Wire Debug Output	Output
D8	B1	23	0	UART0RX	UART0 Receive	Input
			1	NCE23	IO Master N Chip Select 23 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT14	Timer/Counter Interface Signal 14 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO23	General Purpose I/O	Input/ Output
			4	I2SWCLK	I2S Word Clock	Input
			5	CMPOUT	Voltage Comparator Output	Output
			6	MSPI3	MSPI Master Interface Signal 3 See “MSPI Connection” on page 393.	Input/ Output
			7	RSV	Reserved	
C8	B2	24	0	UART1TX	UART1 Transmit	Output
			1	NCE24	IO Master N Chip Select 24 Table 564, “NCE Encoding Table,” on page 382	Output
			2	MSPI8	MSPI Master Interface Signal 8 See “MSPI Connection” on page 393.	Input/ Output
			3	GPIO24	General Purpose I/O	Input/ Output
			4	UA0CTS	UART0 Clear To Send (CTS)	Input
			5	CT21	Timer/Counter Interface Signal 21 See “Implementing Counter/Timer Connections” on page 394.	Output
			6	32kHzXT	32kHz Clock Output	Output
			7	SWO	Serial Wire Debug Output	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A8	D3	25	0	UART1RX	UART1 Receive	Input
			1	NCE25	IO Master N Chip Select 25 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT1	Timer/Counter Interface Signal 1 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO25	General Purpose I/O	Input/ Output
			4	M2SDAWIR3	I2C Master 2 I/O Data SPI Master 2 3 Wire Data	Bidirec-tional Open-Drain
			5	M2MISO	SPI Master 2 Input Data	Input
			6	RSV	Reserved	
			7	RSV	Reserved	
B8	A2	26	0	RSV	Reserved	
			1	NCE26	IO Master N Chip Select 26 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT3	Timer/Counter Interface Signal 3 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO26	General Purpose I/O	Input/ Output
			4	SCCRST	Secure Card Controller Reset	Output
			5	MSPI1	MSPI Master Interface Signal 1 See “MSPI Connection” on page 393.	Input/ Output
			6	UART0TX	UART0 Transmit	Output
			7	UA1CTS	UART1 Clear To Send (CTS)	Input
B9	E3	27	0	UART0RX	UART0 Receive	Input
			1	NCE27	IO Master N Chip Select 27 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT5	Timer/Counter Interface Signal 5 See “Implementing Counter/Timer Connections” on page 397.	Output
			3	GPIO27	General Purpose I/O	Input/ Output
			4	M2SCL	I2C Master 2 Clock	Open Drain
			5	M2SCK	SPI Master 2 Clock	Output
			6	RSV	Reserved	
			7	RSV	Reserved	

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A9	F3	28	0	I2SWCLK	I2S Word Clock	Input
			1	NCE28	IO Master N Chip Select 28 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT7	Timer/Counter Interface Signal 7 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO28	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	M2MOSI	SPI Master 2 Output Data	Output
			6	UART0TX	UART0 Transmit	Output
			7	RSV	Reserved	
A5	B5	29	0	ADCSE1	Analog to Digital Converter Single-Ended Input 1	Input
			1	NCE29	IO Master N Chip Select 29 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT9	Timer/Counter Interface Signal 9 See “Implementing Counter/Timer Connections” on page 394.	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	UART0RX	UART0 Receive	Input
			7	PDMDATA	PDM Data	Input
F6	-	30	0	RSV	Reserved	
			1	NCE30	IO Master N Chip Select 30 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT11	Timer/Counter Interface Signal 11 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO30	General Purpose I/O	Input/ Output
			4	UART0TX	UART0 Transmit	Output
			5	UA1RTS	UART1 Request To Send (RTS)	Output
			6	BLEIF_SCK	BLE Interface SCK Observation	Output
			7	I2SDAT	I2S Data Output	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
D5	-	31	0	ADCSE3	Analog to Digital Converter Single-Ended Input 3	Input
			1	NCE31	IO Master N Chip Select 31 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT13	Timer/Counter Interface Signal 13 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO31	General Purpose I/O	Input/ Output
			4	UART0RX	UART0 Receive	Input
			5	SCCCLK	Secure Card Controller Clock	Output
			6	BLEIF_MISO	BLE Interface MISO Observation	Output
			7	UA1RTS	UART1 Request To Send (RTS)	Output
E6	-	32	0	ADCSE4	Analog to Digital Converter Single-Ended Input 4	Input
			1	NCE32	IO Master N Chip Select 32 Table 564, “NCE Encoding Table,” on page 385	Output
			2	CT15	Timer/Counter Interface Signal 15 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO32	General Purpose I/O	Input/ Output
			4	SCCIO	Secure Card Controller I/O	Input/ Output
			5	RSV	Reserved	
			6	BLEIF_MOSI	BLE Interface MOSI Observation	Output
			7	UA1CTS	UART1 Clear To Send (CTS)	Input
B6	-	33	0	ADCSE5	Analog to Digital Converter Single-Ended Input 5	Input
			1	NCE33	IO Master N Chip Select 33 Table 564, “NCE Encoding Table,” on page 382	Output
			2	32kHzXT	32kHz Clock Output	Output
			3	GPIO33	General Purpose I/O	Input/ Output
			4	BLEIF_CSN	BLE Interface Chip Select Observation	Output
			5	UA0CTS	UART0 Clear To Send (CTS)	Input
			6	CT23	Timer/Counter Interface Signal 23 See “Implementing Counter/Timer Connections” on page 394.	Output
			7	SWO	Serial Wire Debug Output	SWO

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C6	-	34	0	ADCSE6	Analog to Digital Converter Single-Ended Input 6	Input
			1	NCE34	IO Master N Chip Select 34 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UA1RTS	UART1 Request To Send (RTS)	Output
			3	GPIO34	General Purpose I/O	Input/ Output
			4	CMPRF2	Voltage Comparator Reference 2	Input
			5	UA0RTS	UART0 Request To Send (RTS)	Output
			6	UART0RX	UART0 Receive	Input
			7	PDMDATA	PDM Data	Input
D6	-	35	0	ADCSE7	Analog to Digital Converter Single-Ended Input 7	Input
			1	NCE35	IO Master N Chip Select 35 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UART1TX	UART1 Transmit	Output
			3	GPIO35	General Purpose I/O	Input/ Output
			4	I2SDAT	I2S Data	Output
			5	CT27	Timer/Counter Interface Signal 27 See “Implementing Counter/Timer Connections” on page 397.	Output
			6	UA0RTS	UART0 Request To Send (RTS)	Output
			7	BLEIF_STA-TUS	BLE Interface STATUS Observation	Output
H6	-	36	0	TRIG1	ADC Trigger Input 1	Input
			1	NCE36	IO Master N Chip Select 36 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UART1RX	UART1 Receive	Input
			3	GPIO36	General Purpose I/O	Input/ Output
			4	32kHzXT	32kHz Clock Output	Output
			5	UA1CTS	UART1 Clear To Send (CTS)	Input
			6	UA0CTS	UART0 Clear To Send (CTS)	Input
			7	PDMDATA	PDM Data	Input
H7	-	37	0	TRIG2	ADC Trigger Input 2	Input
			1	NCE37	IO Master N Chip Select 37 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UA0RTS	UART0 Request To Send (RTS)	Output
			3	GPIO37	General Purpose I/O	Input/ Output
			4	SCCIO	Secure Card Controller I/O	Input/ Output
			5	UART1TX	UART1 Transmit	Output
			6	PDMCLK	PDM Output Clock	Output
			7	CT29	Timer/Counter Interface Signal 29 See “Implementing Counter/Timer Connections” on page 397.	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
J6	-	38	0	TRIG3	ADC Trigger Input 3	Input
			1	NCE38	IO Master N Chip Select 38 Table 564, “NCE Encoding Table,” on page 382	Output
			2	UA0CTS	UART0 Clear To Send (CTS)	Input
			3	GPIO38	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	M3MOSI	SPI Master 3 Output Data	Output
			6	UART1RX	UART1 Receive	Input
			7	RSV	Reserved	
J8	H2	39	0	UART0TX	UART0 Transmit	Output
			1	UART1TX	UART1 Transmit	Output
			2	CT25	Timer/Counter Interface Signal 25 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO39	General Purpose I/O	Input/ Output
			4	M4SCL	I2C Master 4 Clock	Open Drain
			5	M4SCK	SPI Master 4 Clock	Output
			6	RSV	Reserved	
			7	RSV	Reserved	
J9	H1	40	0	UART0RX	UART0 Receive	Input
			1	UART1RX	UART1 Receive	Input
			2	TRIG0	ADC Trigger Input 0	Input
			3	GPIO40	General Purpose I/O	Input/ Output
			4	M4SDAWIR3	I2C Master 4 I/O Data SPI Master 4 3 Wire Data	Bidirec-tional Open Drain
			5	M4MISO	SPI Master 4 Data Input	Input
			6	RSV	Reserved	
			7	RSV	Reserved	
E5	H3	41	0	NCE41	IO Master N Chip Select 41 Table 564, “NCE Encoding Table,” on page 382	Output
			1	BLEIF_IRQ	BLE Interface IRQ Observation	Output
			2	SWO	Serial Wire Debug Output	Output
			3	GPIO41	General Purpose I/O	Input/ Output
			4	I2SWCLK	I2S Word Clock	Input
			5	UA1RTS	UART1 Request To Send (RTS)	Output
			6	UART0TX	UART0 Transmit	Output
			7	UA0RTS	UART0 Request To Send (RTS)	Output

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
H5	-	42	0	UART1TX	UART1 Transmit	Output
			1	NCE42	IO Master N Chip Select 42 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT16	See “Implementing Counter/Timer Connections” on page 397.	Output
			3	GPIO42	General Purpose I/O	Input/ Output
			4	M3SCL	I2C Master 3 Clock	Open Drain
			5	M3SCK	SPI Master 3 Clock	Output
			6	RSV	Reserved	
			7	RSV	Reserved	
J5	-	43	0	UART1RX	UART1 Receive	Input
			1	NCE43	IO Master N Chip Select 43 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT18	Timer/Counter Interface Signal 18 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO43	General Purpose I/O	Input/ Output
			4	M3SDAWIR3	I2C Master 3 I/O SPI Master 3 3 Wire Data	Bidirec-tional Open Drain
			5	M3MISO	SPI Master 3 Input Data	Input
			6	RSV	Reserved	
			7	RSV	Reserved	
J7	H4	44	0	UA1RTS	UART1 Request To Send (RTS)	Output
			1	NCE44	IO Master N Chip Select 44 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT20	Timer/Counter Interface Signal 20 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO44	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	M4MOSI	SPI Master 4 Output Data	Output
			6	UART0TX	UART0 Transmit	Output
			7	RSV	Reserved	

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F5	-	45	0	UA1CTS	UART1 Clear To Send (CTS)	Input
			1	NCE45	IO Master N Chip Select 45 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT22	Timer/Counter Interface Signal 22 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO45	General Purpose I/O	Input/ Output
			4	I2SDAT	I2S Data	Output
			5	PDMDATA	PDM Data	Input
			6	UART0RX	UART0 Receive	Input
			7	SWO	Serial Wire Debug Output	Output
E7	-	46	0	I2SBCLK	I2S Bit Clock	Input
			1	NCE46	IO Master N Chip Select 46 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT24	Timer/Counter Interface Signal 24 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO46	General Purpose I/O	Input/ Output
			4	SCCRST	Secure Card Controller Reset	Output
			5	PDMCLK	PDM Output Clock	Output
			6	UART1TX	UART1 Transmit	Output
			7	SWO	Serial Wire Debug Output	Output
H9	G2	47	0	32kHzXT	32kHz Clock Output	Output
			1	NCE47	IO Master N Chip Select 47 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT26	Timer/Counter Interface Signal 26 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO47	General Purpose I/O	Input/ Output
			4	RSV	Reserved	
			5	M5MOSI	SPI Master 5 Output Data	Output
			6	UART1RX	UART1 Receive	Input
			7	RSV	Reserved	

Table 1: Pin List and Function Table

BGA Pin Number	CSP Pin Number	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G9	G3	48	0	UART0TX	UART0 Transmit	Output
			1	NCE48	IO Master N Chip Select 48 Table 564, “NCE Encoding Table,” on page 382	Output
			2	CT28	Timer/Counter Interface Signal 28 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO48	General Purpose I/O	Input/ Output
			4	M5SCL	I2C Master 5 Clock	Open Drain
			5	M5SCK	SPI Master 5 Clock	Output
			6	RSV	Reserved	
			7	RSV	Reserved	
H8	G1	49	0	UART0RX	UART0 Receive	Input
			1	NCE49	IO Master N Chip Select 49 Table 564, “NCE Encoding Table,” on page 382	
			2	CT30	Timer/Counter Interface Signal 30 See “Implementing Counter/Timer Connections” on page 394.	Output
			3	GPIO49	General Purpose I/O	Input/ Output
			4	M5SDAWIR3	I2C Master 5 I/O Data SPI Master 5 3 Wire Data	Bidirec-tional Open Drain
			5	M5MISO	SPI Master 5 Input Data	Input
			6	RSV	Reserved	
			7	RSV	Reserved	

2. System Core

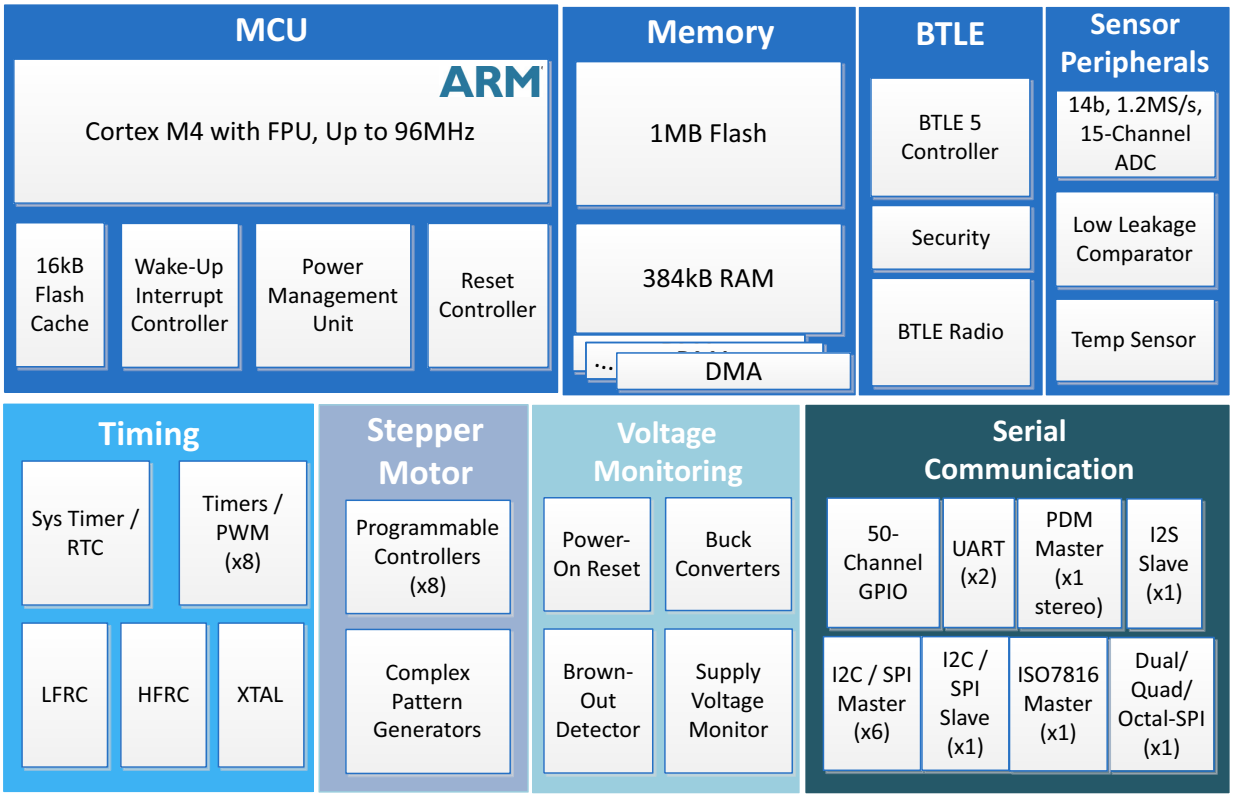


Figure 3. Block Diagram for the Ultra-Low Power Apollo3 Blue MCU

The ultra-low power Apollo3 Blue MCU, shown in Figure 3, is an ideal solution for battery-powered applications requiring sensor measurement and data analysis. In a typical system, the Apollo3 Blue MCU serves as an applications processor for one or more sensors and has a fully integrated BLE 5 radio. The Apollo3 Blue MCU can measure analog sensor outputs using an integrated 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 Apollo3 Blue MCU 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-to-market 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU enables the use of more complex sensor processing algorithms due to its extremely low active mode power of 6 µA/MHz. By using the Apollo3 Blue MCU, 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.

The Apollo3 Blue MCU provides support for higher performance operating modes through Ambiq's TurboSPOT technology. The TurboSPOT technology allows applications to meet critical timing as/when needed while still providing extremely high energy efficiency operation. The Apollo3 Blue MCU also supports secure boot using Ambiq's SecureSPOT technology enabling applications to establish and maintain a root of trust from boot to execution.

At the center of the Apollo3 Blue MCU is a 32-bit ARM Cortex-M4 processor with Floating Point Unit with several tightly coupled peripherals. The Ambiq Micro 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 384 KB Low Leakage RAM. The Wake-Up Interrupt Controller (WIC) coupled with the Cortex-M4 supports sophisticated and configurable sleep state transitions with a variety of interrupt sources.

An integrated Bluetooth low energy controller provides support for Bluetooth 5 at 3mA Rx/Tx. Apollo3 Blue MCU supports up to 4dBm transmit power with optional external power amplifier controls to enable even higher transmit range.

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 MCU intervention. In addition to integrated analog sensor peripherals, I^2C/SPI/PDM master ports and/or UART ports enables the MCU to communicate with external sensors and radios (such as Bluetooth transceivers) that have digital outputs. For higher bandwidth peripherals, the Apollo3 Blue MCU supports a multi-bit SPI (MSPI) controller for 1-bit, 2-bit, 4-bit and 8-bit data. For devices requiring secure communication, it supports an ISO7816 compliant master controller.

The Apollo3 Blue MCU 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 Apollo3 Blue MCU includes clock reliability functions 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.

Apollo3 supports highly optimized PWM pattern generation for complex, efficient stepper motor control operation. Up to 8 independent motors can be controlled from the MCU supporting several different operating modes.

As with any ARM-based MCU, the Apollo3 Blue MCU is supported by a complete suite of standard software development tools. Ambiq Micro 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).

3. MCU Core Details

At the center of the Apollo3 Blue MCU 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 MCU which allows access to all of the memory and peripheral devices of the MCU. 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 MCU core. Where multiple instances of a peripheral exist on Apollo3 Blue MCU (e.g., the I^2C/SPI master modules), base memory addresses for the registers are provided for each and noted as INSTANCE 0, INSTANCE 1, etc.

3.1 Interrupts

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

Below is the M4 Vector Table for Apollo3 Blue MCU:

Exception Number	IRQ Number	Offset	Vector	Peripheral/Description
255	239	0x03FC	IRQ239	
.	.	.	.	
.	.	.	.	
.	.	.	.	
.	.	0x00C0	IRQ31	Clock Control
.	.	0x00BC	IRQ23-30	Stimer Compare[0:7]
.	.	0x009C	IRQ22	Stimer Capture/Overflow
.	.	0x0098	IRQ21	SW INT
.	.	0x0094	IRQ20	MSPI
.	.	0x0090	IRQ19	PDM
.	.	0x008C	IRQ18	ADC
.	.	0x0088	IRQ17	SCARD
.	.	0x0084	IRQ16	UART1
.	.	0x0080	IRQ15	UART0
.	.	0x007C	IRQ14	Counter/Timers
.	.	0x0078	IRQ13	GPIO
.	.	0x0074	IRQ12	BLE
.	.	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	RTC
17	1	0x0048	IRQ1	Watchdog Timer
16	0	0x0044	IRQ0	Brownout Detection
15	-1	0x0040	Systick	
14	-2	0x003C	PendSV	
13		0x0038	Reserved	
12			Reserved for Debug	
11	-5		SVCall	
10		0x002C	Reserved	
9				
8				
7				
6	-10		Usage Fault	
5	-11	0x0018	Bus Fault	
4	-12	0x0014	Memory management Fault	
3	-13	0x0010	Hard fault	
2	-14	0x000C	NMI	Unused
1		0x0008	Reset	
		0x0004	Initial SP value	
		0x0000		

Figure 4. ARM Cortex-M4 Vector Table for Apollo3 Blue MCU

The Cortex-M4 allows the user to assign various interrupts to different priority levels based on the requirements of the application. In this MCU 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 MCU to the M4 NVIC as follows:

Table 2: MCU Interrupt Assignments

IRQ	Peripheral/Description
NMI	Unused
IRQ0	Brownout Detection
IRQ1	Watchdog Timer
IRQ2	RTC
IRQ3	Voltage Comparator
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	BLE
IRQ13	GPIO
IRQ14	Counter/Timers
IRQ15	UART0
IRQ16	UART1
IRQ17	SCARD
IRQ18	ADC
IRQ19	PDM
IRQ20	MSPI
IRQ21	SW INT
IRQ22	STimer Capture/Overflow
IRQ23-30	STimer Compare[0:7]
IRQ31	Clock Control

3.2 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 3: ARM Cortex-M4 Memory Map

Address	Name	Executable	Description
0x00000000 – 0x1FFFFFFF	Code	Y	ROM or 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
0xA0000000 – 0xDFFFFFFF	External Device	N	External device memory
0xE0000000 – 0xE00FFFFF	Private Peripheral Bus	N	NVIC, System timers, System Control Block
0xE0100000 – 0xFFFFFFFF	Vendor	N	Vendor Defined

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

Table 4: MCU System Memory Map

Address	Name	Executable	Description
0x00000000 – 0x000FFFFF	Flash	Y	Flash Memory
0x00100000 - 0x03FFFFFF	Reserved	X	No device at this address range
0x04000000 – 0x07FFFFFF	External MSPI Flash	Y	XIP Read-Only External MSPI Flash
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 – 0x1000FFFF	SRAM (TCM)	Y	Low-power / Low Latency SRAM (TCM)
0x10010000 – 0x1005FFFF	SRAM	Y	Main SRAM
0x10040000 – 0x3FFFFFFF	Reserved	X	No device at this address range
0x40000000 – 0x5FFFFFFF	Peripheral	N	Peripheral devices
0x60000000 – 0xDFFFFFFF	Reserved	X	No device at this address range
0xE0000000 – 0xE00FFFFF	PPB	N	NVIC, System timers, System Control Block
0xE0100000 – 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 5 shows the address mapping for the peripheral devices of the Base Platform.

Table 5: MCU Peripheral Device Memory Map

Address	Device
0x40000000 – 0x400003FF	Reset / BoD Control
0x40000400 – 0x40003FFF	Reserved
0x40004000 – 0x400041FF	Clock Generator
0x40004200 - 0x400043FF	RTC
0x40004400 – 0x40007FFF	Reserved
0x40008000 – 0x400083FF	Timers
0x40008400 – 0x4000BFFF	Reserved
0x4000C000 – 0x4000C3FF	Voltage Comparator
0x4000C400 – 0x4000FFFF	Reserved
0x40010000 – 0x400103FF	GPIO Control
0x40010400 – 0x40010FFF	Reserved
0x40011000 – 0x400113FF	Fast GPIO Control
0x40011400 – 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 – 0x4007FFFF	Reserved
0x40080000 - 0x400803FF	Secure Card
0x40080400 - 0x4FFFFFFF	Reserved
0x50000000 – 0x500003FF	I^2C / SPI Slave
0x50000400 – 0x50003FFF	Reserved
0x50004000 – 0x50004FFF	I^2C / SPI Master0
0x50005000 – 0x50005FFF	I^2C / SPI Master1
0x50006000 – 0x50006FFF	I^2C / SPI Master2

Table 5: MCU Peripheral Device Memory Map

Address	Device
0x50007000 – 0x50007FFF	I^2C / SPI Master3
0x50008000 – 0x50008FFF	I^2C / SPI Master4
0x50009000 – 0x50009FFF	I^2C / SPI Master5
0x5000A000 – 0x5000BFFF	Reserved
0x5000C000 – 0x5000CFFF	BLE
0x5000D000 – 0x5000FFFF	Reserved
0x50010000 – 0x500103FF	ADC
0x50010400 – 0x50010FFF	Reserved
0x50011000 – 0x500113FF	PDM
0x50011400 – 0x50013FFF	Reserved
0x50014000 – 0x500143FF	MSPI Master
0x50014400 – 0x5001FFFF	Reserved
0x50020000 – 0x5002FFFF	Flash OTP
0x50030000 – 0x5FFFFFFF	Reserved
0x51000000 – 0x51FFFFFF	XIP MM (Read/Write to External MSPI Device) *[Chip Rev B Only]*

3.3 Memory Protection Unit (MPU)

The Apollo3 Blue MCU 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.

You can use the MPU to:

- Enforce privilege rules.
- Separate processes.
- Enforce access rules.

3.4 System Busses

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 MCU.

The Apollo3 Blue MCU 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 MCU.

The peripherals of the Apollo3 Blue MCU 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.

3.5 Power Management

The Power Management Unit (PMU) is a finite-state machine that controls the transitions of the MCU 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 MCU. 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 Apollo3 Blue MCU power modes are described in the subsequent discussion along with the operation of the PMU.

3.5.1 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 Apollo3 Blue MCUwill 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.

3.5.1.1 Burst Mode

The Apollo3 Blue MCU supports the Ambiq TurboSPOT which enables a higher frequency operating mode (Burst Mode). In this mode, the M4 and all memory run at an elevated frequency. All of the non-debug ARM clocks (FCLK, HCLK) also operate at the elevated frequency level. All peripherals are maintained at the nominal frequency level during burst. This mode is entered and exited under software direction but transitions are completely handled in hardware.

NOTE

In Burst Mode on the Apollo3 Blue MCU, the SYSTICK increments at twice the normal (48 MHz) clock rate. Some RTOSes may use SYSTICK for scheduler timing by default, in which case scheduler event timing

will be wrong when using Burst Mode. It is recommended not to use SYSTICK and Burst Mode together unless proper compensation is made.

3.5.1.2 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 *ARMv7-M System Control Register* must be set to determine the mode to enter. See page B3-269 of the *ARMv7-M Architecture Reference Manual* for more details.

After the SCR is setup, code can enter the low-power states using one of the 3 following methods:

- Execute a Wait-For-Interrupt (WFI) instruction.
- Execute a Wait-For-Event (WFE) 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
- An event is received by the NVIC
- A Debug Event is received from the DAP

3.5.1.3 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.

3.5.1.4 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 MCU 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.

3.5.2 System Power Modes

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

3.5.2.1 SYS Active Burst (S_{ACTB})

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

3.5.2.2 SYS Active (S_{ACT})

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

3.5.2.3 SYS Sleep Mode 0 (S_{S0})

In SYS Sleep Mode 0, this is a low power state for the MCU. 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/I2C) 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.

3.5.2.4 SYS Sleep Mode 1 (S_{S1})

In SYS Sleep Mode 1, this is a low power state for the MCU. In this mode, all SRAM memory is retained (up to 384 KB), 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/I2C/MSPI/SCARD/BLE) 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.

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

In SYS Deep Sleep Mode 0, this is a deep low power state for the MCU. 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/I2C/MSPI/SCARD/BLE) 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.

Note: For easier notation, SRAM memory retention is defined as follows:

- 384KB: $S_{DS0\text{-}384RET}$
- 256KB: $S_{DS0\text{-}256RET}$
- 128KB: $S_{DS0\text{-}128RET}$
- 64KB: $S_{DS0\text{-}64RET}$
- 8KB: $S_{DS0\text{-}8RET}$
- 0KB: S_{DS0}

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

In SYS Deep Sleep Mode 1, this is a deep low power state for the MCU. 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.

Note: For easier notation, SRAM memory retention is defined as follows:

- 384KB: $S_{DS1\text{-}384RET}$
- 256KB: $S_{DS1\text{-}256RET}$
- 128KB: $S_{DS1\text{-}128RET}$
- 64KB: $S_{DS1\text{-}64RET}$
- 8KB: $S_{DS1\text{-}8RET}$
- 0KB: S_{DS1}

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

In SYS Deep Sleep Mode 2, this is the minimum power state that the MCU can resume normal operation. In this mode, minimal SRAM memory is retained as needed for software to resume (note that SRAM can have 0-384 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.

Note: For easier notation, SRAM memory retention is defined as follows:

- 384KB: $S_{DS2\text{-}384RET}$
- 256KB: $S_{DS2\text{-}256RET}$
- 128KB: $S_{DS2\text{-}128RET}$
- 64KB: $S_{DS2\text{-}64RET}$
- 8KB: $S_{DS2\text{-}8RET}$
- 0KB: S_{DS2}

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.

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

In SYS Deep Sleep Mode 3, this is a deep sleep power state for the MCU. 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.

3.5.2.9 SYS OFF Mode (S_{OFF})

In SYS OFF Mode, MCU 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 MCU by removing power to the device.

3.5.3 Power Control

Power control block provides control and status for the power state of all the power domains, 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 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 BLE/Burst and SIMO 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.

3.5.3.1 PWRCTRL Registers

PWR Controller Register Bank

INSTANCE 0 BASE ADDRESS:0x40021000

Power Controller register Bank - this is the place SW writes to.

3.5.3.1.1 Register Memory Map

Table 6: PWRCTRL Register Map

Address(s)	Register Name	Description
0x40021000	SUPPLYSRC	Voltage Regulator Select Register
0x40021004	SUPPLYSTATUS	Voltage Regulators status
0x40021008	DEVPWREN	Device Power Enables
0x4002100C	MEMPWDINSLEEP	Powerdown SRAM banks in Deep Sleep mode
0x40021010	MEMPWREN	Enables individual banks of the MEMORY array
0x40021014	MEMPWRSTATUS	Mem Power ON Status
0x40021018	DEVPWRSTATUS	Device Power ON Status
0x4002101C	SRAMCTRL	SRAM Control register
0x40021020	ADCSTATUS	Power Status Register for ADC Block
0x40021024	MISC	Power Optimization Control Bits
0x40021028	DEVPWREVENTEN	Event enable register to control which DEVPWRSTATUS bits are routed to event input of CPU.
0x4002102C	MEMPWREVENTEN	Event enable register to control which MEMPWRSTATUS bits are routed to event input of CPU.

3.5.3.1.2 PWRCTRL Registers

3.5.3.1.2.1SUPPLYSRC Register

Voltage Regulator Select Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40021000

This register controls the enable for BLE BUCK.

Table 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																															BLEBUCKEN

Table 8: SUPPLYSRC Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	BLEBUCKEN	0x0	RW	Enables and Selects the BLE Buck as the supply for the BLE power domain or for Burst LDO. It takes the initial value from Customer INFO space. Buck will be powered up only if there is an active request for BLEH domain or Burst mode and appropriate feature is allowed. EN = 0x1 - Enable the BLE Buck. DIS = 0x0 - Disable the BLE Buck.

3.5.3.1.2.2SUPPLYSTATUS Register

Voltage Regulators status

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40021004

Provides an indicator for the BLE BUCK and SIMO BUCK status. Once the SIMO BUCK is powered up MEM and CORE LDOs are disabled.

Table 9: SUPPLYSTATUS Register

31	30	29	28	27	26	25	24	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																														BLEBUCKON	SIMOBUCKON

Table 10: SUPPLYSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	BLEBUCKON	0x0	RO	Indicates whether the BLE (if supported) domain and burst (if supported) domain is supplied from the LDO or the Buck. Buck will be powered up only if there is an active request for BLEH domain or Burst mode and appropriate reature is allowed. LDO = 0x0 - Indicates the the LDO is supplying the BLE/Burst power domain BUCK = 0x1 - Indicates the the Buck is supplying the BLE/Burst power domain
0	SIMOBUCKON	0x0	RO	Indicates whether the Core/Mem low-voltage domains are supplied from the LDO or the Buck. OFF = 0x0 - Indicates the the SIMO Buck is OFF. ON = 0x1 - Indicates the the SIMO Buck is ON.

3.5.3.1.2.3DEVPWREN Register

Device Power Enables

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40021008

This enables various peripherals power domains.

Table 11: DEVPWREN Register

31	30	29	28	27	26	25	24	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																		PWRBLEL	PWRPDM	PWRMSPI	PWRSCARD	PWRADC	PWRUART1	PWRUART0	PWRIOM5	PWRIOM4	PWRIOM3	PWRIOM2	PWRIOM1	PWRIOM0	PWRIOS

Table 12: DEVPWREN Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	RESERVED.
13	PWRBLEL	0x0	RW	Power up BLE controller EN = 0x1 - Power up BLE controller DIS = 0x0 - Power down BLE controller
12	PWRPDM	0x0	RW	Power up PDM block EN = 0x1 - Power up PDM DIS = 0x0 - Power down PDM
11	PWRMSPI	0x0	RW	Power up MSPI Controller EN = 0x1 - Power up MSPI DIS = 0x0 - Power down MSPI
10	PWRSCARD	0x0	RW	Power up SCARD Controller EN = 0x1 - Power up SCARD DIS = 0x0 - Power down SCARD
9	PWRADC	0x0	RW	Power up ADC Digital Controller EN = 0x1 - Power up ADC DIS = 0x0 - Power Down ADC
8	PWRUART1	0x0	RW	Power up UART Controller 1 EN = 0x1 - Power up UART 1 DIS = 0x0 - Power down UART 1
7	PWRUART0	0x0	RW	Power up UART Controller 0 EN = 0x1 - Power up UART 0 DIS = 0x0 - Power down UART 0
6	PWRIOM5	0x0	RW	Power up IO Master 5 EN = 0x1 - Power up IO Master 5 DIS = 0x0 - Power down IO Master 5
5	PWRIOM4	0x0	RW	Power up IO Master 4 EN = 0x1 - Power up IO Master 4 DIS = 0x0 - Power down IO Master 4
4	PWRIOM3	0x0	RW	Power up IO Master 3 EN = 0x1 - Power up IO Master 3 DIS = 0x0 - Power down IO Master 3
3	PWRIOM2	0x0	RW	Power up IO Master 2 EN = 0x1 - Power up IO Master 2 DIS = 0x0 - Power down IO Master 2
2	PWRIOM1	0x0	RW	Power up IO Master 1 EN = 0x1 - Power up IO Master 1 DIS = 0x0 - Power down IO Master 1

Table 12: DEVPWREN Register Bits

Bit	Name	Reset	RW	Description
1	PWRIOM0	0x0	RW	Power up IO Master 0 EN = 0x1 - Power up IO Master 0 DIS = 0x0 - Power down IO Master 0
0	PWRIOS	0x0	RW	Power up IO Slave EN = 0x1 - Power up IO slave DIS = 0x0 - Power down IO slave

3.5.3.1.2.4MEMPWDINSLEEP Register

Powerdown SRAM banks in Deep Sleep mode

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4002100C

This controls the power down of the SRAM banks in deep sleep mode. If this is set, then the power for that SRAM bank will be gated when the core goes into deep sleep. Upon wake, the data within the SRAMs will be erased. If this is not set, retention voltage will be applied to the SRAM bank when the core goes into deep sleep. Upon wake, the data within the SRAMs are retained. Do not set this if the SRAM bank is used as the target for DMA transfer while CPU in deepsleep.

Table 13: MEMPWDINSLEEP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CACHEPWDSLP	RSVD																FLASH1PWDSLP	FLASH0PWDSLP	SRAMPWDSLP										DTCMPWDSLP		

Table 14: MEMPWDINSLEEP Register Bits

Bit	Name	Reset	RW	Description
31	CACHEPWD-SLP	0x0	RW	power down cache in deep sleep EN = 0x1 - Power down cache in deep sleep DIS = 0x0 - Retain cache in deep sleep
30:15	RSVD	0x0	RO	RESERVED.
14	FLASH1PWD-SLP	0x1	RW	Powerdown flash1 in deep sleep EN = 0x1 - Flash1 is powered down during deepsleep DIS = 0x0 - Flash1 is kept powered on during deepsleep

Table 14: MEMPWDINSLEEP Register Bits

Bit	Name	Reset	RW	Description
13	FLASH0PWD-SLP	0x1	RW	Powerdown flash0 in deep sleep EN = 0x1 - Flash0 is powered down during deepsleep DIS = 0x0 - Flash0 is kept powered on during deepsleep
12:3	SRAMPWDSLP	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 = 0x1 - SRAM GROUP0 powered down (64KB-96KB) GROUP1 = 0x2 - SRAM GROUP1 powered down (96KB-128KB) GROUP2 = 0x4 - SRAM GROUP2 powered down (128KB-160KB) GROUP3 = 0x8 - SRAM GROUP3 powered down (160KB-192KB) GROUP4 = 0x10 - SRAM GROUP4 powered down (192KB-224KB) GROUP5 = 0x20 - SRAM GROUP5 powered down (224KB-256KB) GROUP6 = 0x40 - SRAM GROUP6 powered down (256KB-288KB) GROUP7 = 0x80 - SRAM GROUP7 powered down (288KB-320KB) GROUP8 = 0x100 - SRAM GROUP8 powered down (320KB-352KB) GROUP9 = 0x200 - SRAM GROUP9 powered down (352KB-384KB) SRAM32K = 0x1 - Powerdown lower 32k SRAM (64KB-96KB) SRAM64K = 0x3 - Powerdown lower 64k SRAM (64KB-128KB) SRAM128K = 0xF - Powerdown lower 128k SRAM (64KB-192KB) ALLBUTLOWER32K = 0x3FE - All SRAM banks but lower 32k powered down (96KB-384KB). ALLBUTLOWER64K = 0x3FC - All banks but lower 64k powered down. ALLBUTLOWER128K = 0x3F0 - All banks but lower 128k powered down. ALL = 0x3FF - All banks powered down.
2:0	DTCMPWDSLP	0x0	RW	power down DTCM in deep sleep NONE = 0x0 - All DTCM retained GROUP0DTCM0 = 0x1 - Group0_DTCM0 powered down in deep sleep (0KB-8KB) GROUP0DTCM1 = 0x2 - Group0_DTCM1 powered down in deep sleep (8KB-32KB) GROUP0 = 0x3 - Both DTCMs in group0 are powered down in deep sleep (0KB-32KB) ALLBUTGROUP0DTCM0 = 0x6 - Group1 and Group0_DTCM1 are powered down in deep sleep (8KB-64KB) GROUP1 = 0x4 - Group1 DTCM powered down in deep sleep (32KB-64KB) ALL = 0x7 - All DTCMs powered down in deep sleep (0KB-64KB)

3.5.3.1.2.5MEMPWREN Register

Enables individual banks of the MEMORY array

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x40021010

This register enables the individual banks for the memories. When set, power will be enabled to the banks. This register works in conjection with the MEMPWDINSLEEP register. When this register is set, then the MEMPWRINSLEEP register will determine whether power is enabled to the SRAMs in deep sleep. If this register is not set, then power will always be disabled to the memory bank.

Table 15: MEMPWREN Register

31	30	29	28	27	26	25	24	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	CACHEB0	RSVD															FLASH1	FLASH0	SRAM										DTCM		

Table 16: MEMPWREN Register Bits

Bit	Name	Reset	RW	Description
31	CACHEB2	0x1	RW	Power up Cache Bank 2. This works in conjunction with Cache enable from flash_cache module. To power up cache bank2, cache has to be enabled and this bit has to be set. EN = 0x1 - Power up Cache Bank 2 DIS = 0x0 - Power down Cache Bank 2
30	CACHEB0	0x1	RW	Power up Cache Bank 0. This works in conjunction with Cache enable from flash_cache module. To power up cache bank0, cache has to be enabled and this bit has to be set. EN = 0x1 - Power up Cache Bank 0 DIS = 0x0 - Power down Cache Bank 0
29:15	RSVD	0x0	RO	RESERVED.
14	FLASH1	0x1	RW	Power up Flash1 EN = 0x1 - Power up Flash1 DIS = 0x0 - Power down Flash1
13	FLASH0	0x1	RW	Power up Flash0 EN = 0x1 - Power up Flash0 DIS = 0x0 - Power down Flash0
12:3	SRAM	0x3ff	RW	Power up SRAM groups NONE = 0x0 - Do not power ON any of the SRAM banks GROUP0 = 0x1 - Power ON only SRAM group0 (0KB-32KB) GROUP1 = 0x2 - Power ON only SRAM group1 (32KB-64KB) GROUP2 = 0x4 - Power ON only SRAM group2 (64KB-96KB) GROUP3 = 0x8 - Power ON only SRAM group3 (96KB-128KB) GROUP4 = 0x10 - Power ON only SRAM group4 (128KB-160KB) GROUP5 = 0x20 - Power ON only SRAM group5 (160KB-192KB) GROUP6 = 0x40 - Power ON only SRAM group6 (192KB-224KB) GROUP7 = 0x80 - Power ON only SRAM group7 (224KB-256KB) GROUP8 = 0x100 - Power ON only SRAM group8 (256KB-288KB) GROUP9 = 0x200 - Power ON only SRAM group9 (288KB-320KB) SRAM32K = 0x1 - Power ON only lower 32k SRAM64K = 0x3 - Power ON only lower 64k SRAM128K = 0xF - Power ON only lower 128k SRAM256K = 0xFF - Power ON only lower 256k ALL = 0x3FF - All SRAM banks (320K) powered ON

Table 16: MEMPWREN Register Bits

Bit	Name	Reset	RW	Description
2:0	DTCM	0x7	RW	Power up DTCM NONE = 0x0 - Do not enable power to any DTCMs GROUP0DTCM0 = 0x1 - Power ON only GROUP0_DTCM0 GROUP0DTCM1 = 0x2 - Power ON only GROUP0_DTCM1 GROUP0 = 0x3 - Power ON only DTCMs in group0 GROUP1 = 0x4 - Power ON only DTCMs in group1 ALL = 0x7 - Power ON all DTCMs

3.5.3.1.2.6MEMPWRSTATUS Register

Mem Power ON Status

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40021014

It provides the power status for all the memory banks including- caches, flash (0 and 1) and all the SRAM groups. The status here should reflect the enable provided by the MEMPWREN register. There may be a lag time between setting the bits in MEMPWREN register and MEMPWRSTATUS register, due to the need to cycle the power gate and isolation seqeunces to the memory banks.

Table 17: MEMPWRSTATUS Register

31	30	29	28	27	26	25	24	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															CACHEB2	CACHEB0	FLASH1	FLASH0	SRAM9	SRAM8	SRAM7	SRAM6	SRAM5	SRAM4	SRAM3	SRAM2	SRAM1	SRAM0	DTCM1	DTCM01	DTCM00

Table 18: MEMPWRSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	This bitfield is reserved for future use.
16	CACHEB2	0x0	RO	This bit is 1 if power is supplied to Cache Bank 2
15	CACHEB0	0x0	RO	This bit is 1 if power is supplied to Cache Bank 0
14	FLASH1	0x1	RO	This bit is 1 if power is supplied to FLASH 1
13	FLASH0	0x1	RO	This bit is 1 if power is supplied to FLASH 0
12	SRAM9	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP9
11	SRAM8	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP8

Table 18: MEMPWRSTATUS Register Bits

Bit	Name	Reset	RW	Description
10	SRAM7	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP7
9	SRAM6	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP6
8	SRAM5	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP5
7	SRAM4	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP4
6	SRAM3	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP3
5	SRAM2	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP2
4	SRAM1	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP1
3	SRAM0	0x1	RO	This bit is 1 if power is supplied to SRAM GROUP0
2	DTCM1	0x1	RO	This bit is 1 if power is supplied to DTCM GROUP1
1	DTCM01	0x1	RO	This bit is 1 if power is supplied to DTCM GROUP0_1
0	DTCM00	0x1	RO	This bit is 1 if power is supplied to DTCM GROUP0_0

3.5.3.1.2.7DEVPWRSTATUS Register

Device Power ON Status

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40021018

This provides the power status for the peripheral devices- BLEL, PDM, PDM, MSPI, SCARD, ADC, UART0 & 1, IOM5 to 0, IOSLAVE and MCUL (DMA and Fabrics) and MCUH (ARM core). The status here should reflect the enable provided by the DEVPWREN register. There may be a lag time between setting the bits in DEVPWREN register and DEVPWRSTATUS register, due to the need to cycle the power gate, isolation and reset seqeunces to the device power domains.

Table 19: DEVPWRSTATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SYSDEEPSLEEP	COREDEEPSLEEP	CORESLEEP	RSVD																			BLEH	BLEL	PWRPDM	PWRMSPI	PWRADC	HCPC	HCPB	HCPA	MCUH	MCUL

Table 20: DEVPWRSTATUS Register Bits

Bit	Name	Reset	RW	Description
31	SYSDEEPS-LEEP	0x0	RO	This bit is 1 if SYSTEM has been in Deep Sleep. Write '1' to this bit to clear it.
30	COREDEEPS-LEEP	0x0	RO	This bit is 1 if CORE has been in Deep Sleep. Write '1' to this bit to clear it.
29	CORESLEEP	0x0	RO	This bit is 1 if CORE has been in SLEEP State. Write '1' to this bit to clear it.
28:10	RSVD	0x0	RO	This bitfield is reserved for future use.
9	BLEH	0x0	RO	This bit is 1 if power is supplied to BLEH
8	BLEL	0x0	RO	This bit is 1 if power is supplied to BLEL
7	PWRPDM	0x0	RO	This bit is 1 if power is supplied to PDM
6	PWRMSPI	0x0	RO	This bit is 1 if power is supplied to MSPI
5	PWRADC	0x0	RO	This bit is 1 if power is supplied to ADC
4	HCPC	0x0	RO	This bit is 1 if power is supplied to HCPC domain (IO MASTER4, 5, 6)
3	HCPB	0x0	RO	This bit is 1 if power is supplied to HCPB domain (IO MASTER 0, 1, 2)
2	HCPA	0x0	RO	This bit is 1 if power is supplied to HCPA domain (IO SLAVE, UART0, UART1, SCARD)
1	MCUH	0x1	RO	This bit is 1 if power is supplied to MCUH
0	MCUL	0x1	RO	This bit is 1 if power is supplied to MCUL

3.5.3.1.2.8SRAMCTRL Register

SRAM Control register

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4002101C

This register provides additional fine-tune power management controls for the SRAMs and the SRAM controller. This includes enabling light sleep for the SRAM and TCM banks, and clock gating for reduced dynamic power.

Table 21: 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												SRAMLIGHTSLEEP												RSVD					SRAMMASTERCLKGATE	SRAMCLKGATE	RSVD

Table 22: SRAMCTRL Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bitfield is reserved for future use.
19:8	SRAMLIGHT-SLEEP	0x0	RW	Light Sleep enable for each TCM/SRAM bank. When 1, corresponding bank will be put into light sleep. For optimal power, banks should be put into light sleep while the system is active but the bank has minimal or no accesses. ALL = 0xFF - Enable LIGHT SLEEP for ALL SRAMs DIS = 0x0 - Disables LIGHT SLEEP for ALL SRAMs
7:3	RSVD	0x0	RO	This bitfield is reserved for future use.
2	SRAMMASTER-CLKGATE	0x0	RW	This bit is 1 when the master clock gate is enabled (top-level clock gate for entire SRAM block) EN = 0x1 - Enable Master SRAM Clock Gate DIS = 0x0 - Disables Master SRAM Clock Gating
1	SRAMCLKGATE	0x0	RW	This bit is 1 if clock gating is allowed for individual system SRAMs EN = 0x1 - Enable Individual SRAM Clock Gating DIS = 0x0 - Disables Individual SRAM Clock Gating
0	RSVD	0x0	RO	This bitfield is reserved for future use.

3.5.3.1.2.9ADCSTATUS Register

Power Status Register for ADC Block

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40021020

This provides the power status for various blocks within the ADC. These status comes directly from the ADC module and is captured through this interface.

Table 23: 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																										REFBUFPWD	REFKEEPPWD	VBATPWD	VPTATPWD	BGTPWD	ADCPWD

Table 24: ADCSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	REFBUFPWD	0x1	RO	This bit indicates that the ADC REFBUF is powered down
4	REFKEEPPWD	0x1	RO	This bit indicates that the ADC REFKEEP is powered down
3	VBATPWD	0x1	RO	This bit indicates that the ADC VBAT resistor divider is powered down
2	VPTATPWD	0x1	RO	This bit indicates that the ADC temperature sensor input buffer is powered down
1	BGTPWD	0x1	RO	This bit indicates that the ADC Band Gap is powered down
0	ADCPWD	0x1	RO	This bit indicates that the ADC is powered down

3.5.3.1.2.10MISC Register

Power Optimization Control Bits

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x40021024

This register includes additional debug control bits. This is an internal Ambiq-only register. Customers should not attempt to change this or else functionality cannot be guaranteed.

Table 25: MISC Register

31	30	29	28	27	26	25	24	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																								FORCEBLEBUCKACT	MEMVRLPBLE	FORCEMEMVRADC		FORCEMEMVRLPTIMERS	FORCECOREVRLPTIMERS	FORCECOREVRLPPDM	SIMOBUCKEN

Table 26: MISC Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7	FORCEBLE-BUCKACT	0x0	RW	Control Bit to enable BLE Buck to be in active state when BLE Buck is enabled. Default behavior is to be in active only when Burst or BLEH power on are requested.
6	MEMVRLPBLE	0x0	RW	Control Bit to let Mem VR go to lp mode in deep sleep even when BLEL or BLEH is powered on given none of the other domains require it. EN = 0x1 - Mem VR can go to lp mode even when BLE is powered on. DIS = 0x0 - Mem VR will stay in active mode when BLE is powered on.
5:4	FORCEMEM-VRADC	0x0	RW	Control Bit to force mem VR to LP or ACT mode in deep sleep when ADC is powered ON. 0x3 results in picking LP mode. ACT = 0x2 - In this mode if all the other domains but ADC are powered down, mem VR will stay in ACT mode. LP = 0x1 - In this mode if all the other domains but ADC are powered down, mem VR will stay in LP mode. DIS = 0x0 - In this mode if all the other domains but ADC are powered down, mem VR will duty cycle between active and LP modes depending on ADC sampling.
3	FORCEMEM-VRLPTIMERS	0x0	RW	Control Bit to force Mem VR to LP mode in deep sleep even when hfrc based ctimer or stimer is running.
2	FORCECOREV-RLPTIMERS	0x0	RW	Control Bit to force Core VR to LP mode in deep sleep even when hfrc based ctimer or stimer is running.
1	FORCECOREV-RLPPDM	0x0	RW	Control bit to enable the core VR to go into LP mode with HCPA/B/C/MSPI are powered off but PDM is powered on

Table 26: MISC Register Bits

Bit	Name	Reset	RW	Description
0	SIMOBUCKEN	0x0	RW	Enables and Selects the SIMO Buck as the supply for the low-voltage power domain. It takes the initial value from the bit set in Customer INFO space. EN = 0x1 - Enable the SIMO Buck DIS = 0x0 - Disable the SIMO Buck

3.5.3.1.2.11DEVPWREVENTEN Register

Event enable register to control which DEVPWRSTATUS bits are routed to event input of CPU.

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x40021028

This register controls which feature trigger will result in an event to the CPU. It includes all the power on status for the core domains, as well as the Burst event. If any bits are set, then if the domain is turned on, it will result in an event to the ARM core.

Table 27: DEVPWREVENTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BURSTEVEN	BURSTFEATUREEEVEN	BLEFEATUREEEVEN	RSVD																				BLELEVEN	PDMEVEN	MSPIEVEN	ADCEVEN	HCPCEVEN	HCPBEVEN	HCPAEVEN	MCUHEVEN	MCULEVEN

Table 28: DEVPWREVENTEN Register Bits

Bit	Name	Reset	RW	Description
31	BURSTEVEN	0x0	RW	Control BURST status event EN = 0x1 - Enable BURST status event DIS = 0x0 - Disable BURST status event
30	BURSTFEATU-REEEVEN	0x0	RW	Control BURSTFEATURE status event EN = 0x1 - Enable BURSTFEATURE status event DIS = 0x0 - Disable BURSTFEATURE status event
29	BLEFEATU-REEEVEN	0x0	RW	Control BLEFEATURE status event EN = 0x1 - Enable BLEFEATURE status event DIS = 0x0 - Disable BLEFEATURE status event
28:9	RSVD	0x0	RO	RESERVED.

Table 28: DEVPWREVENTEN Register Bits

Bit	Name	Reset	RW	Description
8	BLELEVEN	0x0	RW	Control BLE power-on status event EN = 0x1 - Enable BLE power-on status event DIS = 0x0 - Disable BLE power-on status event
7	PDMEVEN	0x0	RW	Control PDM power-on status event EN = 0x1 - Enable PDM power-on status event DIS = 0x0 - Disable PDM power-on status event
6	MSPIEVEN	0x0	RW	Control MSPI power-on status event EN = 0x1 - Enable MSPI power-on status event DIS = 0x0 - Disable MSPI power-on status event
5	ADCEVEN	0x0	RW	Control ADC power-on status event EN = 0x1 - Enable ADC power-on status event DIS = 0x0 - Disable ADC power-on status event
4	HCPCEVEN	0x0	RW	Control HCPC power-on status event EN = 0x1 - Enable HCPC power-on status event DIS = 0x0 - Disable HCPC power-on status event
3	HCPBEVEN	0x0	RW	Control HCPB power-on status event EN = 0x1 - Enable HCPB power-on status event DIS = 0x0 - Disable HCPB power-on status event
2	HCPAEVEN	0x0	RW	Control HCPA power-on status event EN = 0x1 - Enable HCPA power-on status event DIS = 0x0 - Disable HCPA power-on status event
1	MCUHEVEN	0x0	RW	Control MCUH power-on status event EN = 0x1 - Enable MCHU power-on status event DIS = 0x0 - Disable MCUH power-on status event
0	MCULEVEN	0x0	RW	Control MCUL power-on status event EN = 0x1 - Enable MCUL power-on status event DIS = 0x0 - Disable MCUL power-on status event

3.5.3.1.2.12MEMPWREVENTEN Register

Event enable register to control which MEMPWRSTATUS bits are routed to event input of CPU.

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4002102C

This register controls which power enable for the memories will result in an event to the CPU. It includes all the power on status for the memory domains. If any bits are set, then if the domain is turned on, it will result in an event to the ARM core.

Table 29: MEMPWREVENTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CACHEB2EN	CACHEB0EN	RSVD															FLASH1EN	FLASH0EN	SRAMEN										DTCMEN		

Table 30: MEMPWREVENTEN Register Bits

Bit	Name	Reset	RW	Description
31	CACHEB2EN	0x0	RW	Control CACHEB2 power-on status event EN = 0x1 - Enable CACHE BANK 2 status event DIS = 0x0 - Disable CACHE BANK 2 status event
30	CACHEB0EN	0x0	RW	Control CACHE BANK 0 power-on status event EN = 0x1 - Enable CACHE BANK 0 status event DIS = 0x0 - Disable CACHE BANK 0 status event
29:15	RSVD	0x0	RO	RESERVED.
14	FLASH1EN	0x0	RW	Control Flash power-on status event EN = 0x1 - Enable FLASH status event DIS = 0x0 - Disables FLASH status event
13	FLASH0EN	0x0	RW	Control Flash power-on status event EN = 0x1 - Enable FLASH status event DIS = 0x0 - Disables FLASH status event
12:3	SRAMEN	0x0	RW	Control SRAM power-on status event NONE = 0x0 - Disable SRAM power-on status event GROUP0EN = 0x1 - Enable SRAM group0 (0KB-32KB) power on status event GROUP1EN = 0x2 - Enable SRAM group1 (32KB-64KB) power on status event GROUP2EN = 0x4 - Enable SRAM group2 (64KB-96KB) power on status event GROUP3EN = 0x8 - Enable SRAM group3 (96KB-128KB) power on status event GROUP4EN = 0x10 - Enable SRAM group4 (128KB-160KB) power on status event GROUP5EN = 0x20 - Enable SRAM group5 (160KB-192KB) power on status event GROUP6EN = 0x40 - Enable SRAM group6 (192KB-224KB) power on status event GROUP7EN = 0x80 - Enable SRAM group7 (224KB-256KB) power on status event GROUP8EN = 0x100 - Enable SRAM group8 (256KB-288KB) power on status event GROUP9EN = 0x200 - Enable SRAM group9 (288KB-320KB) power on status event

Table 30: MEMPWREVENTEN Register Bits

Bit	Name	Reset	RW	Description
2:0	DTCMEN	0x0	RW	Enable DTCM power-on status event NONE = 0x0 - Do not enable DTCM power-on status event GROUP0DTCM0EN = 0x1 - Enable GROUP0_DTCM0 power on status event GROUP0DTCM1EN = 0x2 - Enable GROUP0_DTCM1 power on status event GROUP0EN = 0x3 - Enable DTCMs in group0 power on status event GROUP1EN = 0x4 - Enable DTCMs in group1 power on status event ALL = 0x7 - Enable all DTCM power on status event

3.6 Debug Interfaces

A number of useful debug facilities are provided in the Apollo3 Blue MCU.

3.6.1 Debugger Attachment

An external debugger can be connected to the MCU 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.

3.6.2 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.

3.6.3 Trace Port Interface Unit (TPIU)

The Apollo3 Blue MCU 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.

3.6.4 Faulting Address Trapping Hardware

The Apollo3 Blue MCU 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.

3.7 ITM Registers

ARM ITM Registers.

INSTANCE 0 BASE ADDRESS:0xE0000000

3.7.1 Register Memory Map

Table 31: ITM Register Map

Address(s)	Register 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
0xE0000E00	TER	Trace Enable Register.
0xE0000E40	TPR	Trace Privilege Register.
0xE0000E80	TCR	Trace Control Register.
0xE0000FB0	LOCKAREG	Lock Access Register

Table 31: ITM Register Map

Address(s)	Register Name	Description
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

3.7.2 ITM Registers

3.7.2.1 STIM0 Register

Stimulus Port Register 0

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0xE0000000

Stimulus Port Register 0

Table 32: 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 33: STIM0 Register Bits

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

3.7.2.2 STIM1 Register

Stimulus Port Register 1

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0xE0000004

Stimulus Port Register 1

Table 34: 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 35: STIM1 Register Bits

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

3.7.2.3 STIM2 Register

Stimulus Port Register 2

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0xE0000008

Stimulus Port Register 2

Table 36: 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 37: STIM2 Register Bits

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

3.7.2.4 STIM3 Register

Stimulus Port Register 3

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0xE000000C

Stimulus Port Register 3

Table 38: 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 39: STIM3 Register Bits

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

3.7.2.5 STIM4 Register

Stimulus Port Register 4

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0xE0000010

Stimulus Port Register 4

Table 40: 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 41: STIM4 Register Bits

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

3.7.2.6 STIM5 Register

Stimulus Port Register 5

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0xE0000014

Stimulus Port Register 5

Table 42: 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 43: STIM5 Register Bits

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

3.7.2.7 STIM6 Register

Stimulus Port Register 6

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0xE0000018

Stimulus Port Register 6

Table 44: 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 45: STIM6 Register Bits

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

3.7.2.8 STIM7 Register

Stimulus Port Register 7

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0xE000001C

Stimulus Port Register 7

Table 46: 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 47: STIM7 Register Bits

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

3.7.2.9 STIM8 Register

Stimulus Port Register 8

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0xE0000020

Stimulus Port Register 8

Table 48: 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 49: STIM8 Register Bits

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

3.7.2.10 STIM9 Register

Stimulus Port Register 9

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0xE0000024

Stimulus Port Register 9

Table 50: 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 51: STIM9 Register Bits

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

3.7.2.11 STIM10 Register

Stimulus Port Register 10

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0xE0000028

Stimulus Port Register 10

Table 52: 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 53: STIM10 Register Bits

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

3.7.2.12 STIM11 Register

Stimulus Port Register 11

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0xE000002C

Stimulus Port Register 11

Table 54: 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 55: STIM11 Register Bits

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

3.7.2.13 STIM12 Register

Stimulus Port Register 12

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0xE0000030

Stimulus Port Register 12

Table 56: 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 57: STIM12 Register Bits

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

3.7.2.14 STIM13 Register

Stimulus Port Register 13

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0xE0000034

Stimulus Port Register 13

Table 58: 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 59: STIM13 Register Bits

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

3.7.2.15 STIM14 Register

Stimulus Port Register 14

OFFSET: 0x00000038

INSTANCE 0 ADDRESS: 0xE0000038

Stimulus Port Register 14

Table 60: 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 61: STIM14 Register Bits

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

3.7.2.16 STIM15 Register

Stimulus Port Register 15

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0xE000003C

Stimulus Port Register 15

Table 62: 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 63: STIM15 Register Bits

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

3.7.2.17 STIM16 Register

Stimulus Port Register 16

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0xE0000040

Stimulus Port Register 16

Table 64: 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 65: STIM16 Register Bits

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

3.7.2.18 STIM17 Register

Stimulus Port Register 17

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0xE0000044

Stimulus Port Register 17

Table 66: 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 67: STIM17 Register Bits

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

3.7.2.19 STIM18 Register

Stimulus Port Register 18

OFFSET: 0x00000048

INSTANCE 0 ADDRESS: 0xE0000048

Stimulus Port Register 18

Table 68: 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 69: STIM18 Register Bits

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

3.7.2.20 STIM19 Register

Stimulus Port Register 19

OFFSET: 0x0000004C

INSTANCE 0 ADDRESS: 0xE000004C

Stimulus Port Register 19

Table 70: 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 71: STIM19 Register Bits

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

3.7.2.21 STIM20 Register

Stimulus Port Register 20

OFFSET: 0x00000050

INSTANCE 0 ADDRESS: 0xE0000050

Stimulus Port Register 20

Table 72: 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 73: STIM20 Register Bits

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

3.7.2.22 STIM21 Register

Stimulus Port Register 21

OFFSET: 0x00000054

INSTANCE 0 ADDRESS: 0xE0000054

Stimulus Port Register 21

Table 74: 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 75: STIM21 Register Bits

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

3.7.2.23 STIM22 Register

Stimulus Port Register 22

OFFSET: 0x00000058

INSTANCE 0 ADDRESS: 0xE0000058

Stimulus Port Register 22

Table 76: 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 77: STIM22 Register Bits

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

3.7.2.24 STIM23 Register

Stimulus Port Register 23

OFFSET: 0x0000005C

INSTANCE 0 ADDRESS: 0xE000005C

Stimulus Port Register 23

Table 78: 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 79: STIM23 Register Bits

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

3.7.2.25 STIM24 Register

Stimulus Port Register 24

OFFSET: 0x00000060

INSTANCE 0 ADDRESS: 0xE0000060

Stimulus Port Register 24

Table 80: 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 81: STIM24 Register Bits

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

3.7.2.26 STIM25 Register

Stimulus Port Register 25

OFFSET: 0x00000064

INSTANCE 0 ADDRESS: 0xE0000064

Stimulus Port Register 25

Table 82: 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 83: STIM25 Register Bits

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

3.7.2.27 STIM26 Register

Stimulus Port Register 26

OFFSET: 0x00000068

INSTANCE 0 ADDRESS: 0xE0000068

Stimulus Port Register 26

Table 84: 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 85: STIM26 Register Bits

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

3.7.2.28 STIM27 Register

Stimulus Port Register 27

OFFSET: 0x0000006C

INSTANCE 0 ADDRESS: 0xE000006C

Stimulus Port Register 27

Table 86: 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 87: STIM27 Register Bits

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

3.7.2.29 STIM28 Register

Stimulus Port Register 28

OFFSET: 0x00000070

INSTANCE 0 ADDRESS: 0xE0000070

Stimulus Port Register 28

Table 88: 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 89: STIM28 Register Bits

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

3.7.2.30 STIM29 Register

Stimulus Port Register 29

OFFSET: 0x00000074

INSTANCE 0 ADDRESS: 0xE0000074

Stimulus Port Register 29

Table 90: 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 91: STIM29 Register Bits

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

3.7.2.31 STIM30 Register

Stimulus Port Register 30

OFFSET: 0x00000078

INSTANCE 0 ADDRESS: 0xE0000078

Stimulus Port Register 30

Table 92: 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 93: STIM30 Register Bits

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

3.7.2.32 STIM31 Register

Stimulus Port Register 31

OFFSET: 0x0000007C

INSTANCE 0 ADDRESS: 0xE000007C

Stimulus Port Register 31

Table 94: 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 95: STIM31 Register Bits

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

3.7.2.33 TER Register

Trace Enable Register.

OFFSET: 0x00000E00

INSTANCE 0 ADDRESS: 0xE0000E00

Trace Enable Register.

Table 96: 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 97: 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..

3.7.2.34 TPR Register

Trace Privilege Register.

OFFSET: 0x00000E40

INSTANCE 0 ADDRESS: 0xE0000E40

Trace Privilege Register.

Table 98: 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 99: TPR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	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].

3.7.2.35 TCR Register

Trace Control Register.

OFFSET: 0x00000E80

INSTANCE 0 ADDRESS: 0xE0000E80

Trace Control Register.

Table 100: 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 101: 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.
4	SWV_ENABLE	0x0	RW	Enable SWV behavior – count on TPIUEMIT and TPIUBAUD.
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 timestamps 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.

3.7.2.36 LOCKAREG Register

Lock Access Register

OFFSET: 0x00000FB0

INSTANCE 0 ADDRESS: 0xE0000FB0

Lock Access Register

Table 102: 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 103: LOCKAREG Register Bits

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

3.7.2.37 LOCKSREG Register

Lock Status Register

OFFSET: 0x00000FB4

INSTANCE 0 ADDRESS: 0xE0000FB4

Lock Status Register

Table 104: 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 105: LOCKSREG Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	BYTEACC	0x0	RO	You cannot implement 8-bit lock accesses.

Table 105: LOCKSREG Register Bits

Bit	Name	Reset	RW	Description
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.

3.7.2.38 PID4 Register

Peripheral Identification Register 4

OFFSET: 0x00000FD0

INSTANCE 0 ADDRESS: 0xE0000FD0

Peripheral Identification Register 4

Table 106: 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 107: PID4 Register Bits

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

3.7.2.39 PID5 Register

Peripheral Identification Register 5

OFFSET: 0x00000FD4

INSTANCE 0 ADDRESS: 0xE0000FD4

Peripheral Identification Register 5

Table 108: 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 109: PID5 Register Bits

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

3.7.2.40 PID6 Register

Peripheral Identification Register 6

OFFSET: 0x00000FD8

INSTANCE 0 ADDRESS: 0xE0000FD8

Peripheral Identification Register 6

Table 110: 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 111: PID6 Register Bits

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

3.7.2.41 PID7 Register

Peripheral Identification Register 7

OFFSET: 0x00000FDC

INSTANCE 0 ADDRESS: 0xE0000FDC

Peripheral Identification Register 7

Table 112: 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 113: PID7 Register Bits

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

3.7.2.42 PID0 Register

Peripheral Identification Register 0

OFFSET: 0x00000FE0

INSTANCE 0 ADDRESS: 0xE0000FE0

Peripheral Identification Register 0

Table 114: PID0 Register

31	30	29	28	27	26	25	24	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 115: PID0 Register Bits

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

3.7.2.43 PID1 Register

Peripheral Identification Register 1

OFFSET: 0x00000FE4

INSTANCE 0 ADDRESS: 0xE0000FE4

Peripheral Identification Register 1

Table 116: 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 117: PID1 Register Bits

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

3.7.2.44 PID2 Register

Peripheral Identification Register 2

OFFSET: 0x00000FE8

INSTANCE 0 ADDRESS: 0xE0000FE8

Peripheral Identification Register 2

Table 118: 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 119: PID2 Register Bits

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

3.7.2.45 PID3 Register

Peripheral Identification Register 3

OFFSET: 0x00000FEC

INSTANCE 0 ADDRESS: 0xE0000FEC

Peripheral Identification Register 3

Table 120: 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 121: PID3 Register Bits

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

3.7.2.46 CID0 Register

Component Identification Register 1

OFFSET: 0x00000FF0

INSTANCE 0 ADDRESS: 0xE0000FF0

Component Identification Register 1

Table 122: 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 123: CID0 Register Bits

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

3.7.2.47 CID1 Register

Component Identification Register 1

OFFSET: 0x00000FF4

INSTANCE 0 ADDRESS: 0xE0000FF4

Component Identification Register 1

Table 124: 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 125: CID1 Register Bits

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

3.7.2.48 CID2 Register

Component Identification Register 2

OFFSET: 0x00000FF8

INSTANCE 0 ADDRESS: 0xE0000FF8

Component Identification Register 2

Table 126: 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 127: CID2 Register Bits

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

3.7.2.49 CID3 Register

Component Identification Register 3

OFFSET: 0x00000FFC

INSTANCE 0 ADDRESS: 0xE0000FFC

Component Identification Register 3

Table 128: 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 129: CID3 Register Bits

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

3.8 MCUCTRL Registers

MCU Miscellaneous Control Logic

INSTANCE 0 BASE ADDRESS:0x40020000

3.8.1 Register Memory Map

Table 130: MCUCTRL Register Map

Address(s)	Register Name	Description
0x40020000	CHIPPN	Chip Information Register
0x40020004	CHIPID0	Unique Chip ID 0
0x40020008	CHIPID1	Unique Chip ID 1
0x4002000C	CHIPREV	Chip Revision
0x40020010	VENDORID	Unique Vendor ID
0x40020014	SKU	Unique Chip SKU
0x40020018	FEATUREENABLE	Feature Enable on Burst and BLE
0x40020020	DEBUGGER	Debugger Control
0x40020100	BODCTRL	BOD control Register
0x40020104	ADCPWRDLY	ADC Power Up Delay Control
0x4002010C	ADCCAL	ADC Calibration Control
0x40020110	ADCBATTLOAD	ADC Battery Load Enable
0x40020118	ADCTRIM	ADC Trims
0x4002011C	ADCREFCOMP	ADC Referece Keeper and Comparator Control
0x40020120	XTALCTRL	XTAL Oscillator Control
0x40020124	XTALGENCTRL	XTAL Oscillator General Control
0x40020198	MISCCTRL	Miscellaneous control register.
0x400201A0	BOOTLOADER	Bootloader and secure boot functions
0x400201A4	SHADOWVALID	Register to indicate whether the shadow registers have been successfully loaded from the Flash Information Space.
0x400201B0	SCRATCH0	Scratch register that is not reset by any reset
0x400201B4	SCRATCH1	Scratch register that is not reset by any reset
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
0x40020200	DBGR1	Read-only debug register 1
0x40020204	DBGR2	Read-only debug register 2
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.

Table 130: MCUCTRL Register Map

Address(s)	Register Name	Description
0x40020264	OTAPOINTER	OTA (Over the Air) Update Pointer/Status. Reset only by POA
0x40020280	APBDMACTRL	DMA Control Register. Determines misc settings for DMA operation
0x40020284	SRAMMODE	SRAM Controller mode bits
0x40020348	KEXTCLKSEL	Key Register to enable the use of external clock selects via the EXTCLKSEL reg
0x4002035C	SIMOBUCK4	SIMO Buck Control Reg1
0x40020368	BLEBUCK2	BLEBUCK2 Control Reg
0x400203A0	FLASHWPROT0	Flash Write Protection Bits
0x400203A4	FLASHWPROT1	Flash Write Protection Bits
0x400203B0	FLASHRPROT0	Flash Read Protection Bits
0x400203B4	FLASHRPROT1	Flash Read Protection Bits
0x400203C0	DMASRAMWRITEPROTECT0	SRAM write-protection bits.
0x400203C4	DMASRAMWRITEPROTECT1	SRAM write-protection bits.
0x400203D0	DMASRAMREADPROTECT0	SRAM read-protection bits.
0x400203D4	DMASRAMREADPROTECT1	SRAM read-protection bits.

3.8.2 MCUCTRL Registers

3.8.2.1 CHIPPN Register

Chip Information Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40020000

Chip Information Register

Table 131: CHIPPN Register

31	30	29	28	27	26	25	24	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 132: CHIPPN Register Bits

Bit	Name	Reset	RW	Description
31:0	PARTNUM	0x4000000	RO	BCD part number. APOLLO3 = 0x6000000 - Apollo3 part number is 0x06xxxxxx. APOLLO2 = 0x3000000 - Apollo2 part number is 0x03xxxxxx. APOLLO = 0x1000000 - Apollo part number is 0x01xxxxxx. PN_M = 0xFF000000 - Mask for the part number field. PN_S = 0x18 - Bit position for the part number field. FLASHSIZE_M = 0xF00000 - Mask for the FLASH_SIZE field. FLASHSIZE_S = 0x14 - Bit position for the FLASH_SIZE field. SRAMSIZE_M = 0xF0000 - Mask for the SRAM_SIZE field. SRAMSIZE_S = 0x10 - Bit position for the SRAM_SIZE field. REV_M = 0xFF00 - Mask for the revision field. Bits [15:12] are major rev, [11:8] are minor rev. REV_S = 0x8 - Bit position for the revision field. PKG_M = 0xC0 - Mask for the package field. PKG_S = 0x6 - Bit position for the package field. PINS_M = 0x38 - Mask for the pins field. PINS_S = 0x3 - Bit position for the pins field. TEMP_M = 0x6 - Mask for the temperature field. TEMP_S = 0x1 - Bit position for the temperature field. QUAL_M = 0x1 - Mask for the qualified field. QUAL_S = 0x0 - Bit position for the qualified field.

3.8.2.2 CHIPID0 Register

Unique Chip ID 0

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40020004

Unique Chip ID 0

Table 133: 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
CHIPID0																															

Table 134: CHIPID0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHIPID0	0x0	RO	Unique chip ID 0. APOLLO3 = 0x0 - Apollo3 CHIPID0.

3.8.2.3 CHIPID1 Register

Unique Chip ID 1

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40020008

Unique Chip ID 1

Table 135: 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
CHIPID1																															

Table 136: CHIPID1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHIPID1	0x0	RO	Unique chip ID 1. APOLLO3 = 0x0 - Apollo3 CHIPID1.

3.8.2.4 CHIPREV Register

Chip Revision

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4002000C

Chip Revision

Table 137: 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												SIPART												REVMAJ				REVMIN			

Table 138: CHIPREV Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	RESERVED.
19:8	SIPART	0x0	RO	Silicon Part ID
7:4	REVMAJ	0x0	RO	Major Revision ID. B = 0x2 - Apollo3 revision B A = 0x1 - Apollo3 revision A
3:0	REVMIN	0x1	RO	Minor Revision ID. REV1 = 0x2 - Apollo3 minor rev 1. REV0 = 0x1 - Apollo3 minor rev 0. Minor revision value, succeeding minor revisions will increment from this value.

3.8.2.5 VENDORID Register

Unique Vendor ID

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x40020010

Unique Vendor ID

Table 139: 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
VENDORID																															

Table 140: VENDORID Register Bits

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

3.8.2.6 SKU Register

Unique Chip SKU

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40020014

Unique Chip SKU

Table 141: SKU Register

31	30	29	28	27	26	25	24	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																													SECBOOT	ALLOWBLE	ALLOW-

Table 142: SKU Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	SECBOOT	0x0	RO	Secure boot feature allowed
1	ALLOWBLE	0x0	RO	Allow BLE feature
0	ALLOWBURST	0x0	RO	Allow Burst feature

3.8.2.7 FEATUREENABLE Register

Feature Enable on Burst and BLE

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40020018

Feature Enable on Burst and BLE

Table 143: FEATUREENABLE Register

31	30	29	28	27	26	25	24	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																									BURSTAVAIL	BURSTACK	BURSTREQ	RSVD	BLEAVAIL	BLEACK	BLEREQ

Table 144: FEATUREENABLE Register Bits

Bit	Name	Reset	RW	Description
31:7	RSVD	0x0	RO	RESERVED.
6	BURSTAVAIL	0x0	RO	Availability of Burst functionality AVAIL = 0x1 - Burst functionality available NOTAVAIL = 0x0 - Burst functionality not available
5	BURSTACK	0x0	RO	ACK for BURSTREQ
4	BURSTREQ	0x0	RW	Controls the Burst functionality EN = 0x1 - Enable the Burst functionality DIS = 0x0 - Disable the Burst functionality
3	RSVD	0x0	RO	RESERVED.
2	BLEAVAIL	0x0	RO	AVAILABILITY of the BLE functionality AVAIL = 0x1 - BLE functionality available NOTAVAIL = 0x0 - BLE functionality not available
1	BLEACK	0x0	RO	ACK for BLEREQ
0	BLEREQ	0x1	RW	Controls the BLE functionality EN = 0x1 - Enable the BLE functionality DIS = 0x0 - Disable the BLE functionality

3.8.2.8 DEBUGGER Register

Debugger Control

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40020020

Debugger Control

Table 145: DEBUGGER Register

31	30	29	28	27	26	25	24	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																															LOCKOUT

Table 146: DEBUGGER Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	LOCKOUT	0x0	RW	Lockout of debugger (SWD).

3.8.2.9 BODCTRL Register

BOD control Register

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x40020100

BOD control Register

Table 147: BODCTRL Register

31	30	29	28	27	26	25	24	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																										BODHVREFSEL	BODLVREFSEL	BODFPWD	BODCPWD	BODHPWD	BODLPWD

Table 148: BODCTRL Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	BODHVREFSEL	0x0	RW	BODH External Reference Select. Note: the SWE mux select in PWRSEQ2SWE must be set for this to take effect.
4	BODLVREFSEL	0x0	RW	BODL External Reference Select. Note: the SWE mux select in PWRSEQ2SWE must be set for this to take effect.
3	BODFPWD	0x0	RW	BODF Power Down.
2	BODCPWD	0x0	RW	BODC Power Down.
1	BODHPWD	0x0	RW	BODH Power Down.
0	BODLPWD	0x0	RW	BODL Power Down.

3.8.2.10 ADCPWRDLY Register

ADC Power Up Delay Control

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x40020104

ADC Power Up Delay Control

Table 149: ADCPWRDLY Register

31	30	29	28	27	26	25	24	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																ADCPWR1								ADCPWR0							

Table 150: ADCPWRDLY Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED.
15:8	ADCPWR1	0x0	RW	ADC Reference Keeper enable delay in 16 ADC CLK increments for ADC_CLKSEL = 0x1, 8 ADC CLOCK increments for ADC_CLKSEL = 0x2.
7:0	ADCPWR0	0x0	RW	ADC Reference Buffer Power Enable delay in 64 ADC CLK increments for ADC_CLKSEL = 0x1, 32 ADC CLOCK increments for ADC_CLKSEL = 0x2.

3.8.2.11 ADCCAL Register

ADC Calibration Control

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x4002010C

ADC Calibration Control

Table 151: 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 152: ADCCAL Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	ADCCALI-BRATED	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

3.8.2.12 ADCBATTLOAD Register

ADC Battery Load Enable

OFFSET: 0x00000110

INSTANCE 0 ADDRESS: 0x40020110

ADC Battery Load Enable

Table 153: 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 154: 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

3.8.2.13 ADCTRIM Register

ADC Trims

OFFSET: 0x00000118

INSTANCE 0 ADDRESS: 0x40020118

ADC Trims

Table 155: ADCTRIM Register

31	30	29	28	27	26	25	24	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																			ADCRFBUFIBTRIM		ADCREFBUFTRIM					RSVD				ADCREFKEEPIBTRIM	

Table 156: ADCTRIM Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED.
12:11	ADCRFBUFIB-TRIM	0x0	RW	ADC reference buffer input bias trim
10:6	ADCREFBUF-TRIM	0x8	RW	ADC Reference buffer trim
5:2	RSVD	0x0	RO	RESERVED.
1:0	ADCREFKEEP-IBTRIM	0x0	RW	ADC Reference Ibias trim

3.8.2.14 ADCREFCOMP Register

ADC Referece Keeper and Comparator Control

OFFSET: 0x0000011C

INSTANCE 0 ADDRESS: 0x4002011C

ADC Referece Keeper and Comparator Control

Table 157: 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			ADCREFKEEPTRIM					RSVD							ADC_REFCOMP_OUT

Table 158: 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	ADCREFKEEP-TRIM	0x0	RW	ADC Reference Keeper Trim
7:1	RSVD	0x0	RO	RESERVED
0	ADC_REFCOM-P_OUT	0x0	RO	Output of the ADC reference comparator

3.8.2.15 XTALCTRL Register

XTAL Oscillator Control

OFFSET: 0x00000120

INSTANCE 0 ADDRESS: 0x40020120

XTAL Oscillator Control

Table 159: 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																						XTALICOMPTRIM		XTALIBUFTRIM		PWDBODXTAL	PDNBCMPRXTAL	PDNBCOREXTAL	BYPCMPRXTAL	FDBKDSBLXTAL	XTALSWE

Table 160: XTALCTRL Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED.
9:8	XTALI-COMPTRIM	0x3	RW	XTAL ICOMP trim
7:6	XTALIBUFTRIM	0x1	RW	XTAL IBUFF trim

Table 160: XTALCTRL Register Bits

Bit	Name	Reset	RW	Description
5	PWDBODXTAL	0x0	RW	XTAL Power down on brown out. PWRUPBOD = 0x0 - Power up xtal on BOD PWRDNBOD = 0x1 - Power down XTAL on BOD.
4	PDNBCMPRX-TAL	0x1	RW	XTAL Oscillator Power Down Comparator. PWRUPCOMP = 0x1 - Power up XTAL oscillator comparator. PWRDNCOMP = 0x0 - Power down XTAL oscillator comparator.
3	PDNBCOREX-TAL	0x1	RW	XTAL Oscillator Power Down Core. PWRUPCORE = 0x1 - Power up XTAL oscillator core. PWRDNCORE = 0x0 - 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	FDBKDSBLX-TAL	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. OVERRIDE_DIS = 0x0 - XTAL Software Override Disable. OVERRIDE_EN = 0x1 - XTAL Software Override Enable.

3.8.2.16 XTALGENCTRL Register

XTAL Oscillator General Control

OFFSET: 0x00000124

INSTANCE 0 ADDRESS: 0x40020124

XTAL Oscillator General Control

Table 161: 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																		XTALKSBIASTRIM						XTALBIASTRIM						ACWARMUP	

Table 162: XTALGENCTRL Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	RESERVED.
13:8	XTALKSBIAS-TRIM	0x1	RW	XTAL IBIAS Kick start trim. This trim value is used during the startup process to enable a faster lock.
7:2	XTALBIASTRIM	0x0	RW	XTAL BIAS trim
1:0	ACWARMUP	0x0	RW	Auto-calibration delay control SEC1 = 0x0 - Warmup period of 1-2 seconds SEC2 = 0x1 - Warmup period of 2-4 seconds SEC4 = 0x2 - Warmup period of 4-8 seconds SEC8 = 0x3 - Warmup period of 8-16 seconds

3.8.2.17 MISCCTRL Register

Miscellaneous control register.

OFFSET: 0x00000198

INSTANCE 0 ADDRESS: 0x40020198

Miscellaneous control register.

Table 163: MISCCTRL Register

31	30	29	28	27	26	25	24	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																										BLE_RESETN	RESERVED_RW_0				

Table 164: MISCCTRL Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	BLE_RESETN	0x0	RW	BLE reset signal.
4:0	RESERVED_RW_0	0x0	RW	Reserved bits, always leave unchanged. The MISCCTRL register must be modified via atomic RMW, leaving this bitfield completely unmodified. Failure to do so will result in unpredictable behavior.

3.8.2.18 BOOTLOADER Register

Bootloader and secure boot functions

OFFSET: 0x000001A0

INSTANCE 0 ADDRESS: 0x400201A0

Bootloader and secure boot functions

Table 165: BOOTLOADER Register

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

Table 166: BOOTLOADER Register Bits

Bit	Name	Reset	RW	Description
31:30	SEC-BOOTONRST	0x0	RO	Indicates whether the secure boot on warm reset is enabled DISABLED = 0x0 - Secure boot disabled ENABLED = 0x1 - Secure boot enabled ERROR = 0x2 - Error in secure boot configuration
29:28	SECBOOT	0x0	RO	Indicates whether the secure boot on cold reset is enabled DISABLED = 0x0 - Secure boot disabled ENABLED = 0x1 - Secure boot enabled ERROR = 0x2 - Error in secure boot configuration
27:26	SECBOOTFEA-TURE	0x0	RO	Indicates whether the secure boot feature is enabled. DISABLED = 0x0 - Secure boot disabled ENABLED = 0x1 - Secure boot enabled ERROR = 0x2 - Error in secure boot configuration
25:3	RSVD	0x0	RO	RESERVED.
2	PROTLOCK	0x1	RW	Flash protection lock. Always resets to 1, write 1 to clear. Enables writes to flash protection register set. LOCK = 0x1 - Enable the secure boot lock
1	SBLOCK	0x1	RW	Secure boot lock. Always resets to 1, write 1 to clear. Enables system visibility to bootloader until set. LOCK = 0x1 - Enable the secure boot lock

Table 166: BOOTLOADER Register Bits

Bit	Name	Reset	RW	Description
0	BOOTLOADER-LOW	0x1	RW	Determines whether the bootloader code is visible at address 0x00000000 or not. Resets to 1, write 1 to clear. ADDR0 = 0x1 - Bootloader code at 0x00000000.

3.8.2.19 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 167: 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																													INFO0_VALID	BLDSLEEP	VALID

Table 168: SHADOWVALID Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	INFO0_VALID	0x1	RO	Indicates whether info0 contains valid data VALID = 0x1 - Flash info0 (customer) space contains valid data.
1	BLDSLEEP	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.

3.8.2.20 SCRATCH0 Register

Scratch register that is not reset by any reset

OFFSET: 0x000001B0

INSTANCE 0 ADDRESS: 0x400201B0

Scratch register that is not reset by any reset

Table 169: SCRATCH0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SCRATCH0																															

Table 170: SCRATCH0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCRATCH0	0x0	RW	Scratch register 0.

3.8.2.21 SCRATCH1 Register

Scratch register that is not reset by any reset

OFFSET: 0x000001B4

INSTANCE 0 ADDRESS: 0x400201B4

Scratch register that is not reset by any reset

Table 171: SCRATCH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SCRATCH1																															

Table 172: SCRATCH1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCRATCH1	0x0	RW	Scratch register 1.

3.8.2.22 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 173: 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
ICODEFAULTADDR																															

Table 174: ICODEFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	ICODE-FAULTADDR	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.

3.8.2.23 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 175: 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
DCODEFAULTADDR																															

Table 176: DCODEFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	DCODE-FAULTADDR	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.

3.8.2.24 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 177: 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
SYSFAULTADDR																															

Table 178: SYSFAULTADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	SYS-FAULTADDR	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.

3.8.2.25 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 179: 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																													SYSFAULT	DCODEFAULT	ICODEFAULT

Table 180: FAULTSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	SYSFAULT	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.

Table 180: FAULTSTATUS Register Bits

Bit	Name	Reset	RW	Description
1	DCODEFAULT	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	ICODEFAULT	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.

3.8.2.26 FAULTCAPTUREEN Register

Enable the fault capture registers

OFFSET: 0x000001D0

INSTANCE 0 ADDRESS: 0x400201D0

Enable the fault capture registers

Table 181: 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																															FAULTCAPTUREEN

Table 182: FAULTCAPTUREEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	FAULTCAPTU-REEN	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.

3.8.2.27 DBGR1 Register

Read-only debug register 1

OFFSET: 0x00000200

INSTANCE 0 ADDRESS: 0x40020200

Read-only debug register 1

Table 183: DBGR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ONETO8																															

Table 184: DBGR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	ONETO8	0x12345678	RO	Read-only register for communication validation

3.8.2.28 DBGR2 Register

Read-only debug register 2

OFFSET: 0x00000204

INSTANCE 0 ADDRESS: 0x40020204

Read-only debug register 2

Table 185: DBGR2 Register

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

Table 186: DBGR2 Register Bits

Bit	Name	Reset	RW	Description
31:0	COOLCODE	0xc001c0de	RO	Read-only register for communication validation

3.8.2.29 PMUENABLE Register

Control bit to enable/disable the PMU

OFFSET: 0x00000220

INSTANCE 0 ADDRESS: 0x40020220

Control bit to enable/disable the PMU

Table 187: 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 188: PMUENABLE Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.
0	ENABLE	0x1	RW	PMU Enable Control bit. When set, the MCU's PMU will place the MCU 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 MCU power management. EN = 0x1 - Enable MCU power management.

3.8.2.30 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 189: 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 190: TPIUCTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED.

Table 190: TPIUCTRL Register Bits

Bit	Name	Reset	RW	Description
10:8	CLKSEL	0x0	RW	This field selects the frequency of the ARM M4 TPIU port. LOWPWR = 0x0 - Low power state. HFRCDIV2 = 0x1 - Selects HFRC divided by 2 as the source TPIU clk HFRCDIV8 = 0x2 - Selects HFRC divided by 8 as the source TPIU clk HFRCDIV16 = 0x3 - Selects HFRC divided by 16 as the source TPIU clk HFRCDIV32 = 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.

3.8.2.31 OTAPOINTER Register

OTA (Over the Air) Update Pointer/Status. Reset only by POA

OFFSET: 0x00000264

INSTANCE 0 ADDRESS: 0x40020264

OTA (Over the Air) Update Pointer/Status. Reset only by POA

Table 191: OTAPOINTER Register

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

Table 192: OTAPOINTER Register Bits

Bit	Name	Reset	RW	Description
31:2	OTAPOINTER	0x0	RW	Flash page pointer with updated OTA image
1	OTASBLUP-DATE	0x0	RW	Indicates that the sbl_init has been updated

Table 192: OTAPOINTER Register Bits

Bit	Name	Reset	RW	Description
0	OTAVALID	0x0	RW	Indicates that an OTA update is valid

3.8.2.32 APBDMACTRL Register

DMA Control Register. Determines misc settings for DMA operation

OFFSET: 0x00000280

INSTANCE 0 ADDRESS: 0x40020280

DMA Control Register. Determines misc settings for DMA operation

Table 193: APBDMACTRL Register

31	30	29	28	27	26	25	24	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																HYSTERESIS								RSVD						DECODEABORT	DMA_ENABLE

Table 194: APBDMACTRL Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED.
15:8	HYSTERESIS	0x2	RW	This field determines how long the DMA will remain active during deep sleep before shutting down and returning the system to full deep sleep. Values are based on a 94KHz clock and are roughly 10us increments for a range of ~10us to 2.55ms
7:2	RSVD	0x0	RO	RESERVED.
1	DECODEABORT	0x1	RW	APB Decode Abort. When set, the APB bridge will issue a data abort (bus fault) on transactions to peripherals that are powered down. When set to 0, writes are quietly discarded and reads return 0. DISABLE = 0x0 - Bus operations to powered down peripherals are quietly discarded ENABLE = 0x1 - Bus operations to powered down peripherals result in a bus fault.
0	DMA_ENABLE	0x1	RW	Enable the DMA controller. When disabled, DMA requests will be ignored by the controller DISABLE = 0x0 - DMA operations disabled ENABLE = 0x1 - DMA operations enabled

3.8.2.33 SRAMMODE Register

SRAM Controller mode bits

OFFSET: 0x00000284

INSTANCE 0 ADDRESS: 0x40020284

SRAM Controller mode bits

Table 195: SRAMMODE Register

31	30	29	28	27	26	25	24	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		DPREFETCH_CACHE	DPREFETCH	RSVD		IPREFETCH_CACHE	IPREFETCH

Table 196: SRAMMODE Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:6	RSVD	0x0	RO	RESERVED.
5	DPREFETCH_-CACHE	0x0	RW	Secondary prefetch feature that will cache prefetched data across bus wait-states (requires DPREFETCH to be set).
4	DPREFETCH	0x0	RW	When set, data bus accesses to the SRAM banks will be prefetched (normally 2 cycle read access). Use of this mode bit is only recommended if the work flow has a large number of sequential accesses.
3:2	RSVD	0x0	RO	RESERVED.
1	IPREFETCH_-CACHE	0x0	RW	Secondary prefetch feature that will cache prefetched data across bus wait-states (requires IPREFETCH to be set).
0	IPREFETCH	0x0	RW	When set, instruction accesses to the SRAM banks will be prefetched (normally 2 cycle read access). Generally, this mode bit should be set for improved performance when executing instructions from SRAM.

3.8.2.34 KEXTCLKSEL Register

Key Register to enable the use of external clock selects via the EXTCLKSEL reg

OFFSET: 0x00000348

INSTANCE 0 ADDRESS: 0x40020348

Key Register to enable the use of external clock selects via the EXTCLKSEL reg

Table 197: KEXTCLKSEL Register

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

Table 198: KEXTCLKSEL Register Bits

Bit	Name	Reset	RW	Description
31:0	KEXTCLKSEL	0x0	RW	Key register value. Key = 0x53 - Key

3.8.2.35 SIMOBUCK4 Register

SIMO Buck Control Reg1

OFFSET: 0x0000035C

INSTANCE 0 ADDRESS: 0x4002035C

SIMO Buck Control Reg1

Table 199: SIMOBUCK4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SIMOBUCKIBIASTRIM				SIMOBUCKUVLOMODE		SIMOBUCKPRIORITYSEL	SIMOBUCKCOMP2TIMEOUTEN	SIMOBUCKCOMP2LPEN	SIMOBUCKCLKDIVSEL		SIMOBUCKEXTCLKSEL	SIMOBUCKUVLODRVSTRTRIM			SIMOBUCKUVLOCNTRTRIM			SIMOBUCKZXTRIM				SIMOBUCKMEMLEAKAGETRIM		SIMOBUCKMEMLPDRVSTRTRIM		SIMOBUCKMEMACTDRVSTRTRIM		SIMOBUCKMEMLPLOWTONTRIM			

Table 200: SIMOBUCK4 Register Bits

Bit	Name	Reset	RW	Description
31:28	SIMOBUCKIBI-ASTRIM	0x3	RW	simobuck_bias_trim

Table 200: SIMOBUCK4 Register Bits

Bit	Name	Reset	RW	Description
27:26	SIMOBUCKU-VLOMODE	0x3	RW	simobuck_uvlo_mode. In B0, these bits are used as SIMOBUCK mode bits. uvlo_mode[0] enables use of tonclk_lp for all operations and uvlo_mode[1] controls core_low/mem_low synchronization. USE_LP_CLOCK = 0x1 - LP clock is used for simobuck in both active and low-power mode. X_LOW_NOSYNC = 0x2 - No synchronization is applied to core_low/mem_low inputs (A1 behavior) X_LOW_SYNC = 0x0 - Synchronization is applied to core_low/mem_low inputs
25	SIMOBUCKPRI-ORITYSEL	0x0	RW	simobuck_priority_sel
24	SIMOBUCK-COMP2TIME-OUTEN	0x0	RW	simobuck_comp2_timeout_en
23	SIMOBUCK-COMP2LPEN	0x1	RW	simobuck_comp2_lp_en
22:21	SIMOBUCK-CLKDIVSEL	0x0	RW	simobuck_clkdiv_sel
20	SIMOBUCKEX-TCLKSEL	0x0	RW	simobuck_extclk_sel
19:17	SIMOBUCKU-VLODRVSTR-TRIM	0x6	RW	simobuck_uvlo_drvstr_trim
16:14	SIMOBUCKU-VLOCNTRTRIM	0x6	RW	For B0, this register has been redefined as mode bits for the Simobuck. Each bit is independent: [0]=always enable LP clock [1]=enable priority_state [2]=enable zx_comp reset removal fix ENABLE_LP_CLK = 0x1 - When set to 1, the LP clock will always be activated. When 0, the logic will request the clock when needed DISABLE_PRIORITY_STATE = 0x2 - (Inverse polarity mode bit) When set to 1, the priority state logic will be disabled and when set to 0, priority_state will enforce that both core and mem bucks get equal priority. ENABLE_ZXCOMP_SYNC = 0x4 - When set to 1, ZXCOMP will be routed through a flop and removal synchronized to the internal clock. When set to 0, logic will act like A1 logic and will be asynchronous.
13:10	SIMOBUCKZX-TRIM	0x0	RW	simobuck_zx_trim
9:8	SIMOBUCK-MEMLEAKAG-ETRIM	0x0	RW	simobuck_mem_leakage_trim
7:6	SIMOBUCK-MEMLP-DRVSTRTRIM	0x2	RW	simobuck_mem_lp_drvstr_trim
5:4	SIMOBUCKME-MACTDRVSTR-TRIM	0x2	RW	simobuck_mem_act_drvstr_trim
3:0	SIMOBUCK-MEMLPLOW-TONTRIM	0xa	RW	simobuck_mem_lp_low_ton_trim

3.8.2.36 BLEBUCK2 Register

BLEBUCK2 Control Reg

OFFSET: 0x00000368

INSTANCE 0 ADDRESS: 0x40020368

BLEBUCK2 Control Reg

Table 201: BLEBUCK2 Register

31	30	29	28	27	26	25	24	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														BLEBUCKTOND2ATRIM						BLEBUCKTONHITRIM						BLEBUCKTONLOWTRIM					

Table 202: BLEBUCK2 Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED.
17:12	BLEBUCK-TOND2ATRIM	0x0	RO	blebuck_ton_trim
11:6	BLEBUCKTON-HITRIM	0x1	RW	blebuck_ton_hi_trim
5:0	BLEBUCKTON-LOWTRIM	0xe	RW	blebuck_ton_low_trim

3.8.2.37 FLASHWPROT0 Register

Flash Write Protection Bits

OFFSET: 0x000003A0

INSTANCE 0 ADDRESS: 0x400203A0

These bits write-protect flash in 16KB chunks.

Table 203: FLASHWPROT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FW0BITS																															

Table 204: FLASHWPROT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	FW0BITS	0x0	RW	Write protect flash 0x00000000 - 0x0007FFFF. Each bit provides write protection for 16KB chunks of flash data space. Bits are cleared by writing a 1 to the bit. When read, 0 indicates the region is protected. Bits are sticky (can be set when PROTLOCK is 1, but only cleared by reset)

3.8.2.38 FLASHWPROT1 Register

Flash Write Protection Bits

OFFSET: 0x000003A4

INSTANCE 0 ADDRESS: 0x400203A4

These bits write-protect flash in 16KB chunks.

Table 205: FLASHWPROT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FW1BITS																															

Table 206: FLASHWPROT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	FW1BITS	0x0	RW	Write protect flash 0x00080000 - 0x000FFFFF. Each bit provides write protection for 16KB chunks of flash data space. Bits are cleared by writing a 1 to the bit. When read, 0 indicates the region is protected. Bits are sticky (can be set when PROTLOCK is 1, but only cleared by reset)

3.8.2.39 FLASHRPROT0 Register

Flash Read Protection Bits

OFFSET: 0x000003B0

INSTANCE 0 ADDRESS: 0x400203B0

These bits read-protect flash in 16KB chunks.

Table 207: FLASHRPROT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FR0BITS																															

Table 208: FLASHRPROT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	FR0BITS	0x0	RW	Copy (read) protect flash 0x00000000 - 0x0007FFFF. Each bit provides read protection for 16KB chunks of flash. Bits are cleared by writing a 1 to the bit. When read, 0 indicates the region is protected. Bits are sticky (can be set when PROTLOCK is 1, but only cleared by reset)

3.8.2.40 FLASHRPROT1 Register

Flash Read Protection Bits

OFFSET: 0x000003B4

INSTANCE 0 ADDRESS: 0x400203B4

These bits read-protect flash in 16KB chunks.

Table 209: FLASHRPROT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FR1BITS																															

Table 210: FLASHRPROT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	FR1BITS	0x0	RW	Copy (read) protect flash 0x00080000 - 0x000FFFFF. Each bit provides read protection for 16KB chunks of flash. Bits are cleared by writing a 1 to the bit. When read, 0 indicates the region is protected. Bits are sticky (can be set when PROTLOCK is 1, but only cleared by reset)

3.8.2.41 DMASRAMWRITEPROTECT0 Register

SRAM write-protection bits.

OFFSET: 0x000003C0

INSTANCE 0 ADDRESS: 0x400203C0

These bits write-protect system SRAM from DMA operations in 8KB chunks.

Table 211: DMASRAMWRITEPROTECT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DMA_WPROT0																															

Table 212: DMASRAMWRITEPROTECT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	DMA_WPROT0	0x0	RW	Write protect SRAM from DMA. Each bit provides write protection for an 8KB region of memory. When set to 1, the region will be protected from DMA writes, when set to 0, DMA may write the region.

3.8.2.42 DMASRAMWRITEPROTECT1 Register

SRAM write-protection bits.

OFFSET: 0x000003C4

INSTANCE 0 ADDRESS: 0x400203C4

These bits write-protect system SRAM from DMA operations in 8KB chunks.

Table 213: DMASRAMWRITEPROTECT1 Register

31	30	29	28	27	26	25	24	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																DMA_WPROT1															

Table 214: DMASRAMWRITEPROTECT1 Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED.
15:0	DMA_WPROT1	0x0	RW	Write protect SRAM from DMA. Each bit provides write protection for an 8KB region of memory. When set to 1, the region will be protected from DMA writes, when set to 0, DMA may write the region.

3.8.2.43 DMASRAMREADPROTECT0 Register

SRAM read-protection bits.

OFFSET: 0x000003D0

INSTANCE 0 ADDRESS: 0x400203D0

These bits read-protect system SRAM from DMA operations in 8KB chunks.

Table 215: DMASRAMREADPROTECT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DMA_RPROT0																															

Table 216: DMASRAMREADPROTECT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	DMA_RPROT0	0x0	RW	Read protect SRAM from DMA. Each bit provides write protection for an 8KB region of memory. When set to 1, the region will be protected from DMA reads, when set to 0, DMA may read the region.

3.8.2.44 DMASRAMREADPROTECT1 Register

SRAM read-protection bits.

OFFSET: 0x000003D4

INSTANCE 0 ADDRESS: 0x400203D4

These bits read-protect system SRAM from DMA operations in 8KB chunks.

Table 217: DMASRAMREADPROTECT1 Register

31	30	29	28	27	26	25	24	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																DMA_RPROT1															

Table 218: DMASRAMREADPROTECT1 Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED.
15:0	DMA_RPROT1	0x0	RW	Read protect SRAM from DMA. Each bit provides write protection for an 8KB region of memory. When set to 1, the region will be protected from DMA reads, when set to 0, DMA may read the region.

3.9 Memory Subsystem

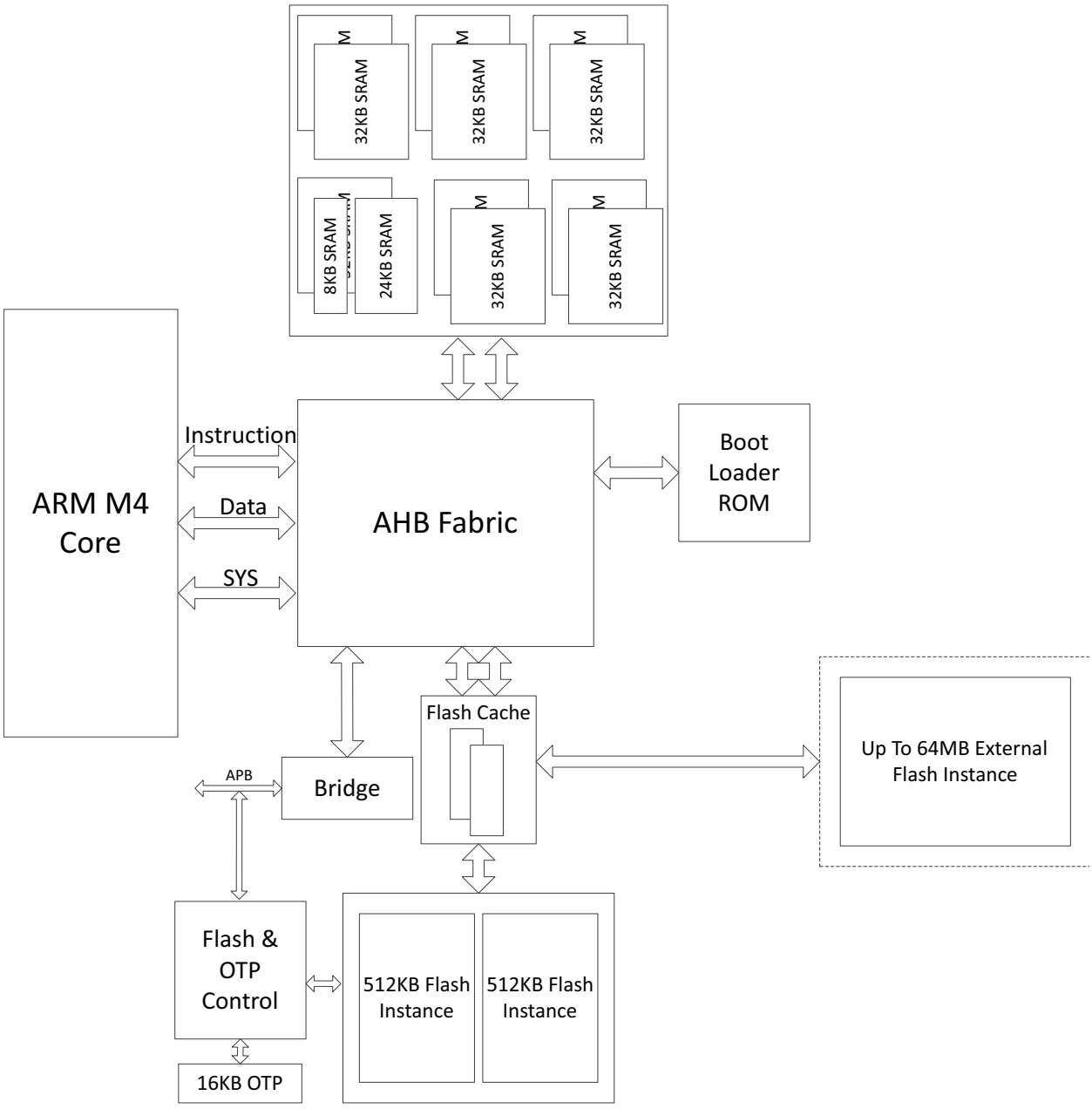


Figure 5. Block Diagram for Flash and OTP Memory Subsystem

3.9.1 Features

The Apollo3 Blue MCU integrates four kinds of memory as shown in Figure 5:

- SRAM
- Integrated Flash / External Flash via MSPI (with Flash cache)
- Boot Loader ROM
- One Time Programmable (OTP) memory

Key features include:

- 384KB SRAM

- 2 instances of 512 KB flash Memory (up to 1 MB total)
- 16 KB Flash cache (2-way set-associative/Direct Mapped, 512 entry, 128b linesize)
- 16 KBytes OTP
  - 8 KBytes contain factory preset per chip trim values.
  - 8 KBytes 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
- External Flash with XiP (via MSPI) with cache support (up to 64MB)

3.9.2 Functional Overview

The Apollo3 Blue MCU Integrates up to 1024 KBytes of on-board flash memory and 16 Kbytes of one time programmable memory. These two memories are managed by the APB flash controller for write operations.

During normal MCU 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.

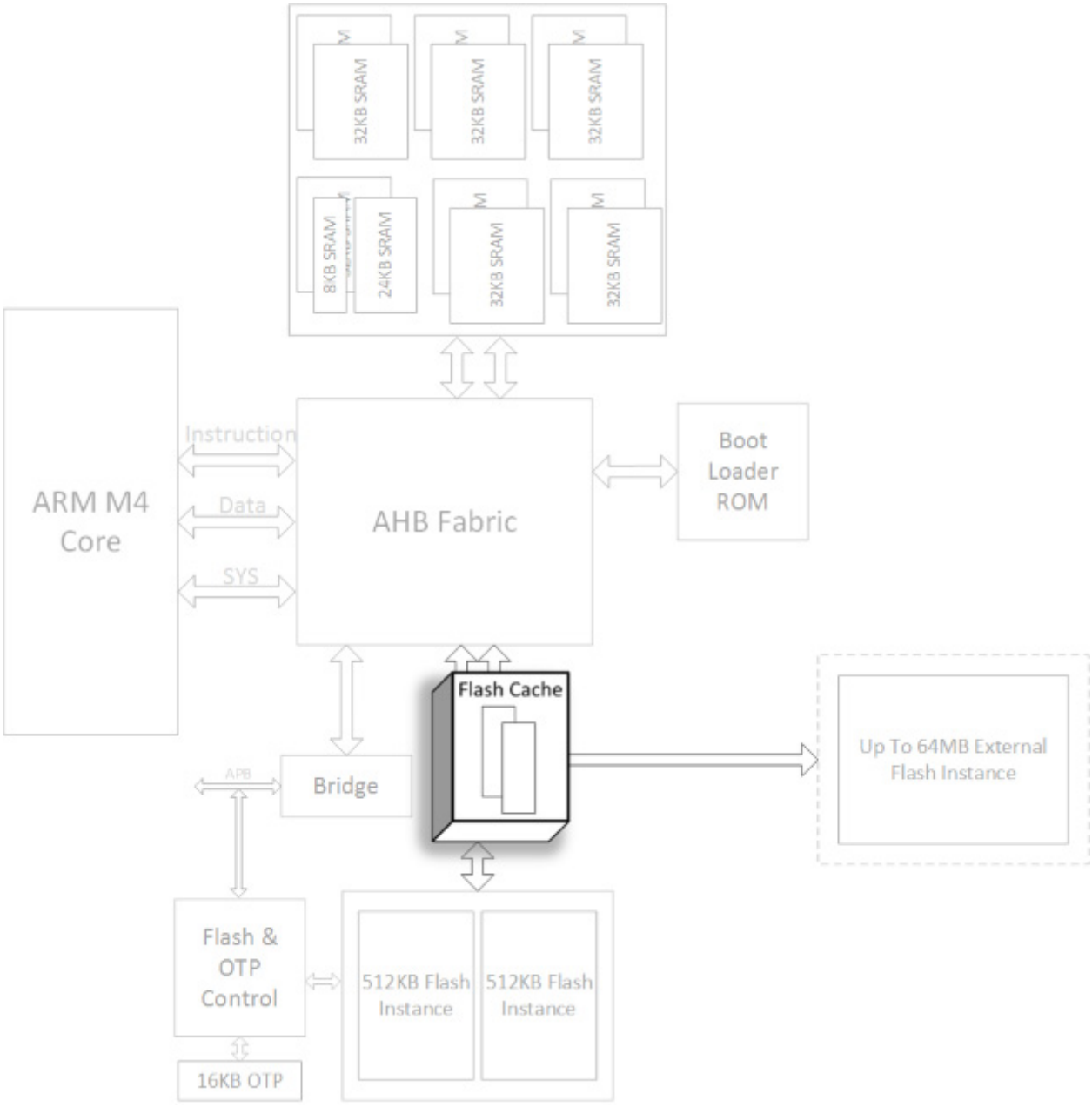
The boot loader ROM contains approximately 4 KB of instructions that are executed upon power up of the processor. Once a valid reset vector is establish at offset zero in the flash memory, the boot loader transfers control to users application by issuing a POR type reset which causes the core to enter the reset vector in flash.

The Apollo3 Blue MCU supports secure boot leveraging the SecureSPOT technology. The root of trust for the secure boot is the boot ROM and the Ambiq secure boot loader. Secure boot, if enabled, will be invoked on each boot and reset cycle. Some secure boot functionality is conditionally supported on reset leveraging the SECBOOTONRST configuration in OTP. More details on the Apollo3 Blue MCU security features are described in “Security” on page 173 and also in the Ambiq Apollo3 Blue MCU Security Whitepaper.

3.9.3 Flash Cache

3.9.3.1 Functional Overview

Figure 6. Block Diagram for Apollo3 Blue MCU with Flash Cache



Apollo3 Blue MCU 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). Caching is supported for the entire 1MB internal Flash and the 64MB external Flash aperture (via MSPI).

The cache is configurable 2-way set associative or direct mapped, 512 or 256 entry and 128b linesize.

3.9.3.1.1 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 supercedes 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.

3.9.3.1.2 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 waitstates added to accesses, so the elapsed time should be used to infer this value (waitstates are added as a result of cache misses or contention for the tag lookup if both busses require a simultaneous lookup).

3.9.3.2 CACHECTRL Registers

Flash Cache Controller

INSTANCE 0 BASE ADDRESS:0x40018000

3.9.3.2.1 Register Memory Map

Table 219: CACHECTRL Register Map

Address(s)	Register Name	Description
0x40018000	CACHECFG	Flash Cache Control Register
0x40018004	FLASHCFG	Flash Control Register
0x40018008	CTRL	Cache Control
0x40018010	NCR0START	Flash Cache Noncachable Region 0 Start
0x40018014	NCR0END	Flash Cache Noncachable Region 0 End
0x40018018	NCR1START	Flash Cache Noncachable Region 1 Start
0x4001801C	NCR1END	Flash Cache Noncachable Region 1 End
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

3.9.3.2.2 CACHECTRL Registers

3.9.3.2.2.1 CACHECFG Register

Flash Cache Control Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40018000

Flash Cache Control Register

Table 220: 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	RSVD								CACHE_LS	CACHE_CLKGATE	DCACHE_ENABLE	ICACHE_ENABLE	CONFIG				ENABLE_NC1	ENABLE_NC0	LRU	ENABLE

Table 221: CACHECFG Register Bits

Bit	Name	Reset	RW	Description
31:25	RSVD	0x0	RO	This bitfield is reserved for future use.
24	ENABLE_MONITOR	0x0	RW	Enable Cache Monitoring Stats. Cache monitoring consumes additional power and should only be enabled when profiling code and counters will increment when this bit is set. Counter values will be retained when this is set to 0, allowing software to enable/disable counting for multiple code segments.
23:21	RSVD	0x0	RO	This bitfield is reserved for future use.
20	DATA_CLKGATE	0x1	RW	Enable aggressive clock gating of entire data array. This bit should be set to 1 for optimal power efficiency.
19:12	RSVD	0x0	RO	This bitfield is reserved for future use.
11	CACHE_LS	0x1	RW	Enable LS (light sleep) of cache RAMs. Software should DISABLE this bit since cache activity is too high to benefit from LS usage.
10	CACHE_CLKGATE	0x1	RW	Enable clock gating of cache TAG RAM. Software should enable this bit for optimal power efficiency.
9	DCACHE_ENABLE	0x0	RW	Enable Flash Data Caching.
8	ICACHE_ENABLE	0x0	RW	Enable Flash Instruction Caching

Table 221: CACHECFG Register Bits

Bit	Name	Reset	RW	Description
7:4	CONFIG	0x5	RW	Sets the cache configuration W1_128B_512E = 0x4 - Direct mapped, 128-bit linesize, 512 entries (4 SRAMs active) W2_128B_512E = 0x5 - Two-way set associative, 128-bit linesize, 512 entries (8 SRAMs active) W1_128B_1024E = 0x8 - Direct mapped, 128-bit linesize, 1024 entries (8 SRAMs active)
3	ENABLE_NC1	0x0	RW	Enable Non-cacheable region 1. See NCR1 registers to define the region.
2	ENABLE_NC0	0x0	RW	Enable Non-cacheable region 0. See NCR0 registers to define the region.
1	LRU	0x0	RW	Sets the cache repleacment policy. 0=LRR (least recently replaced), 1=LRU (least recently used). LRR minimizes writes to the TAG SRAM.
0	ENABLE	0x0	RW	Enables the flash cache controller and enables power to the cache SRAMs. The ICACHE_ENABLE and DCACHE_ENABLE should be set to enable caching for each type of access.

3.9.3.2.2.2FLASHCFG Register

Flash Control Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40018004

Flash Control Register

Table 222: 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																		LPMMODE		LPM_RD_WAIT				RSVD	SEDELAY			RD_WAIT			

Table 223: FLASHCFG Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	This bitfield is reserved for future use.

Table 223: FLASHCFG Register Bits

Bit	Name	Reset	RW	Description
13:12	LPMMODE	0x0	RW	Controls flash low power modes (control of LPM pin). NEVER = 0x0 - High power mode (LPM not used). STANDBY = 0x1 - Fast Standby mode. LPM deasserted for read operations, but asserted while flash IDLE. ALWAYS = 0x2 - Low Power mode. LPM always asserted for reads. LPM_RD_WAIT must be programmed to accomodate longer read access times.
11:8	LPM_RD_WAIT	0x8	RW	Sets flash waitstates when in LPM Mode 2 (RD_WAIT in LPM mode 2 only)
7	RSVD	0x0	RO	This bitfield is reserved for future use.
6:4	SEDELAY	0x7	RW	Sets SE delay (flash address setup). A value of 5 is recommended.
3:0	RD_WAIT	0x3	RW	Sets read waitstates for normal (fast) operation. A value of 1 is recommended.

3.9.3.2.2.3CTRL Register

Cache Control

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40018008

Cache Control

Table 224: CTRL Register

31	30	29	28	27	26	25	24	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 225: CTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield is reserved for future use.

Table 225: CTRL Register Bits

Bit	Name	Reset	RW	Description
10	FLASH1_SLM_ENABLE	0x0	WO	Enable Flash Sleep Mode. Write to 1 to put flash 1 into sleep mode. NOTE: there is a 5us latency after waking flash until the first access will be returned.
9	FLASH1_SLM_-DISABLE	0x0	WO	Disable Flash Sleep Mode. Write 1 to wake flash1 from sleep mode (reading the array will also automatically wake it).
8	FLASH1_SLM_STATUS	0x0	RO	Flash Sleep Mode Status. 1 indicates that flash1 is in sleep mode, 0 indicates flash1 is in normal mode.
7	RSVD	0x0	RO	This bitfield is reserved for future use.
6	FLASH0_SLM_ENABLE	0x0	WO	Enable Flash Sleep Mode. Write to 1 to put flash 0 into sleep mode. NOTE: there is a 5us latency after waking flash until the first access will be returned.
5	FLASH0_SLM_-DISABLE	0x0	WO	Disable Flash Sleep Mode. Write 1 to wake flash0 from sleep mode (reading the array will also automatically wake it).
4	FLASH0_SLM_STATUS	0x0	RO	Flash Sleep Mode Status. 1 indicates that flash0 is in sleep mode, 0 indicates flash0 is in normal mode.
3	RSVD	0x0	RO	This bitfield is reserved for future use.
2	CACHE_READY	0x0	RO	Cache Ready Status (enabled and not processing an invalidate operation)
1	RESET_STAT	0x0	WO	Reset Cache Statistics. When written to a 1, the cache monitor counters will be cleared. The monitor counters can be reset only when the CACHECFG.ENABLE_MONITOR bit is set. CLEAR = 0x1 - Clear Cache Stats
0	INVALIDATE	0x0	WO	Writing a 1 to this bitfield invalidates the flash cache contents.

3.9.3.2.2.4NCR0START Register

Flash Cache Noncachable Region 0 Start

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x40018010

Flash Cache Noncachable Region 0 Start

Table 226: 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 227: NCR0START Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bitfield is reserved for future use.
26:4	ADDR	0x0	RW	Start address for non-cacheable region 0
3:0	RSVD	0x0	RO	This bitfield is reserved for future use.

3.9.3.2.2.5NCR0END Register

Flash Cache Noncachable Region 0 End

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40018014

Flash Cache Noncachable Region 0 End

Table 228: 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 229: NCR0END Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bitfield is reserved for future use.
26:4	ADDR	0x0	RW	End address for non-cacheable region 0
3:0	RSVD	0x0	RO	This bitfield is reserved for future use.

3.9.3.2.2.6NCR1START Register

Flash Cache Noncachable Region 1 Start

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40018018

Flash Cache Noncachable Region 1 Start

Table 230: 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 231: NCR1START Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bitfield is reserved for future use.
26:4	ADDR	0x0	RW	Start address for non-cacheable region 1
3:0	RSVD	0x0	RO	This bitfield is reserved for future use.

3.9.3.2.2.7NCR1END Register

Flash Cache Noncachable Region 1 End

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4001801C

Flash Cache Noncachable Region 1 End

Table 232: 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 233: NCR1END Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bitfield is reserved for future use.
26:4	ADDR	0x0	RW	End address for non-cacheable region 1
3:0	RSVD	0x0	RO	This bitfield is reserved for future use.

3.9.3.2.2.8DMON0 Register

Data Cache Total Accesses

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x40018040

Data Cache Total Accesses

Table 234: 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 235: DMON0 Register Bits

Bit	Name	Reset	RW	Description
31:0	DAC-CESS_COUNT	0x0	RO	Total accesses to data cache. All performance metrics should be relative to the number of accesses performed.

3.9.3.2.2.9DMON1 Register

Data Cache Tag Lookups

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0x40018044

Data Cache Tag Lookups

Table 236: 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 237: DMON1 Register Bits

Bit	Name	Reset	RW	Description
31:0	DLOOK-UP_COUNT	0x0	RO	Total tag lookups from data cache.

3.9.3.2.2.10DMON2 Register

Data Cache Hits

OFFSET: 0x00000048

INSTANCE 0 ADDRESS: 0x40018048

Data Cache Hits

Table 238: 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 239: DMON2 Register Bits

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

3.9.3.2.2.11DMON3 Register

Data Cache Line Hits

OFFSET: 0x0000004C

INSTANCE 0 ADDRESS: 0x4001804C

Data Cache Line Hits

Table 240: 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 241: DMON3 Register Bits

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

3.9.3.2.2.12IMON0 Register

Instruction Cache Total Accesses

OFFSET: 0x00000050

INSTANCE 0 ADDRESS: 0x40018050

Instruction Cache Total Accesses

Table 242: 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 243: IMON0 Register Bits

Bit	Name	Reset	RW	Description
31:0	IAC-CESS_COUNT	0x0	RO	Total accesses to Instruction cache

3.9.3.2.2.13IMON1 Register

Instruction Cache Tag Lookups

OFFSET: 0x00000054

INSTANCE 0 ADDRESS: 0x40018054

Instruction Cache Tag Lookups

Table 244: 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 245: IMON1 Register Bits

Bit	Name	Reset	RW	Description
31:0	ILOOK-UP_COUNT	0x0	RO	Total tag lookups from Instruction cache

3.9.3.2.2.14IMON2 Register

Instruction Cache Hits

OFFSET: 0x00000058

INSTANCE 0 ADDRESS: 0x40018058

Instruction Cache Hits

Table 246: 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 247: IMON2 Register Bits

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

3.9.3.2.2.15IMON3 Register

Instruction Cache Line Hits

OFFSET: 0x0000005C

INSTANCE 0 ADDRESS: 0x4001805C

Instruction Cache Line Hits

Table 248: IMON3 Register

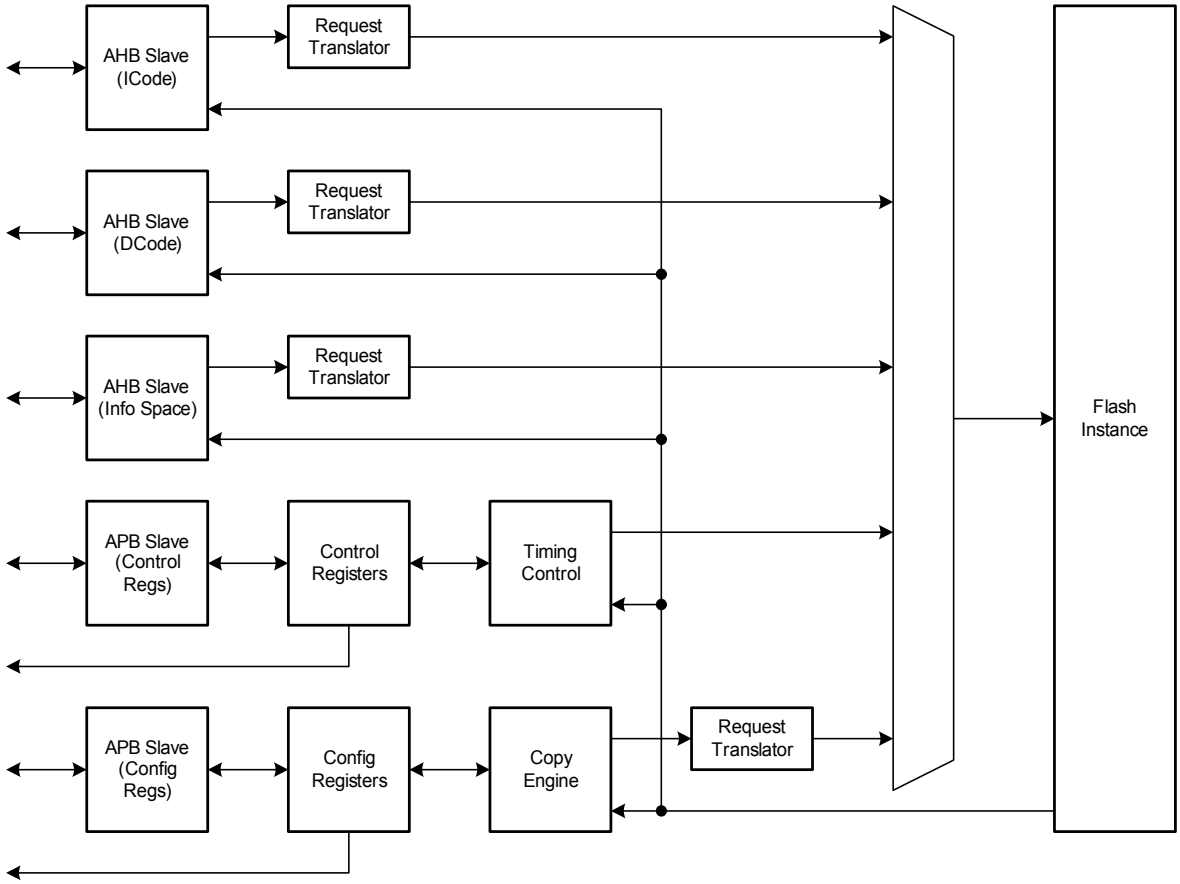
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ILINE_COUNT																															

Table 249: IMON3 Register Bits

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

3.9.3.3 Flash Memory Controller

Figure 7. Block diagram for the Flash Memory Controller



3.9.3.3.1 Functional Overview

During normal MCU 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 MCU 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.

Note: Programming the same bit cannot be reprogrammed to a ‘0’ before an erase cycle. Doing so may cause data corruption/retention issues within the word line. Ambiq provides a “modify” function to help with this.

Note: The number of program cycles performed per word line (512 bytes) should be limited to no more than 160 between erase cycles. Alternatively, 1 full line program plus 100 additional program cycles to the

same line should not be exceeded. Doing more than the restricted number of program cycles to the same line between erase operations may cause data corruption/retention issues within the word line.

3.9.4 SRAM Interface

3.9.4.1 Functional Overview

Figure 8. Block diagram for the SRAM Interface

The SRAM Interface translates requests from the CPU core and DMA controllers 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. On Apollo3, the DTCM banks are guaranteed to be zero wait-state unless there is contention for that specific memory array with another requestor (CPU I/D Bus or DMA Bus). The Main SRAM banks are zero wait-state for sequential accesses or 1-wait state for non-sequential accesses for I/D Bus accesses unless there is contention for that specific memory array with another requestor (CPU I/D Bus or DMA Bus). DMA accesses to Main SRAM are always 0-wait state unless there is contention for that specific memory array. Prefetching is used on the I/D Bus accesses to Main SRAM to minimize/eliminate wait-state bubbles. Prefetching can be enabled/disabled for I and/or D Bus accesses.

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 8 shows a logical block diagram of the SRAM Interface.

4. Security

4.1 Functional Overview

The Apollo3 Blue MCU includes the following security features:

- Secure Boot
- Secure OTA
- Secure Key Storage
- Key Revocation
- AES128, SHA256
- CRC32
- External Flash Inline Encryption/Decryption

More details on the Apollo3 Blue MCU security features are described in the Ambiq Apollo3 Blue MCU Security Whitepaper.

4.2 Secure Boot

The Secure Boot feature on the Apollo3 Blue MCU provides a secure foundation for customer firmware. The secure boot loader provides authentication, decryption and integrity validation for customer firmware on installation and boot/reset. Secure boot loader provides firmware recovery and OTA update support.

Secure boot is configurable leveraging OTP to direct the secure boot loader based on the customer security requirements.

The secure boot flow is illustrated in Figure 9.

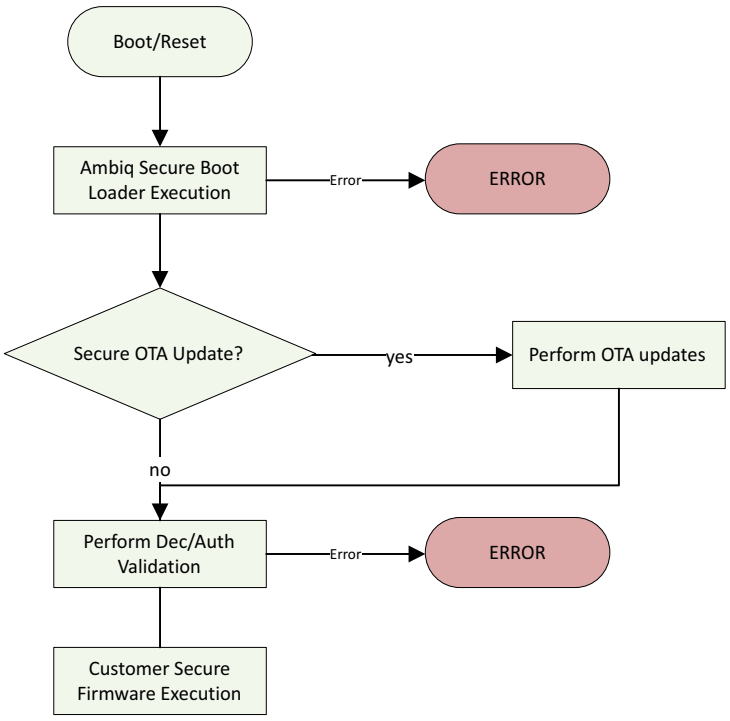


Figure 9. Secure Boot Flow

4.3 Secure OTA

Apollo3 Blue MCU supports secure OTA leveraging the Ambiq secure boot loader. Customers can update any firmware component securely as directed via the security policy configuration in OTP.

The basic flow is shown in Figure 10.

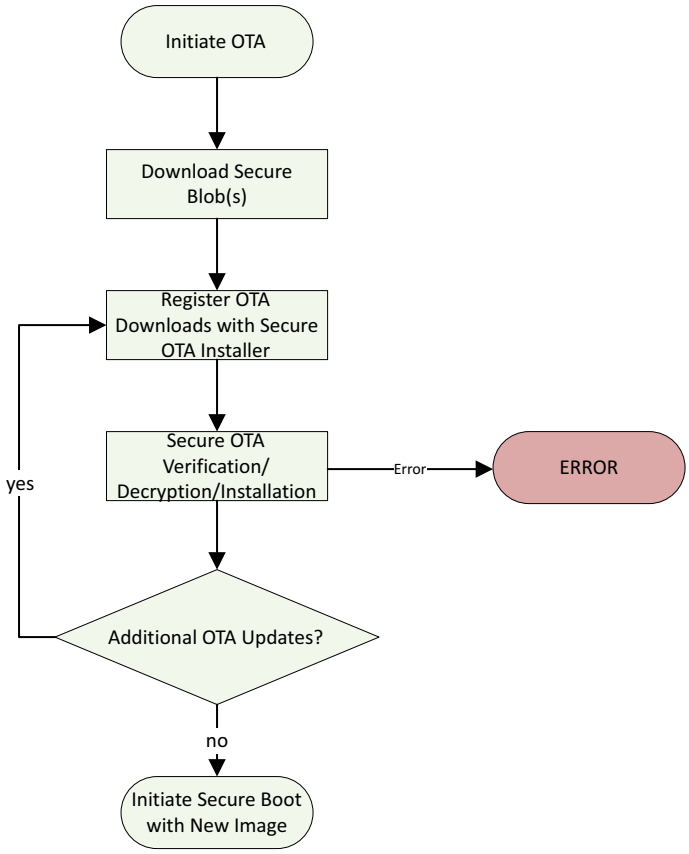


Figure 10. Secure OTA Flow

4.4 Secure Key Storage

Key Storage is managed in hardware and provides secure access to keys as needed to support secure boot as well as any other runtime security operation. A customer key bank is provided and can be provisioned and allocated as required to support various configurations.

It may also be necessary to initiate key revocation in the event a particular key is compromised, stale or needing to be refreshed. The Apollo3 Blue MCU provides key revocation for customer key bank keys as needed. The number of key revocations supported is dependent on the length of keys required and the partitioning/allocation of the key bank.

4.5 External Flash Inline Encrypt/Decrypt

External flash is supported on Apollo3 Blue MCU via the MSPI controller interface. The MSPI controller supports inline encrypt/decrypt to enable customers to securely store firmware or any other secure image data in external flash without concern of the firmware/data confidentiality being compromised.

The Ambiq secure inline encrypt/decrypt provides robust, high performance and extremely low power protection for external flash contents. Ambiq's inline encrypt/decrypt enables truly inline capability that does not degrade performance when asking external flash.

For more details on the inline support, See "MSPI Master Module" on page 216.

5. DMA

5.1 Functional Overview

The Apollo3 Blue MCU supports DMA capability for the following peripheral controllers:

- SPI Master
- I2C Master
- PDM
- ADC
- MSPI
- BLE
- Security

DMA is supported from peripheral to SRAM and SRAM/Flash to peripheral. DMA transactions to/from SRAM occur concurrently to CPU instruction/data accesses as long as the accesses are to different physical banks of memory. Accesses to the same physical bank are arbitrated in hardware. Similarly, accesses to Flash occur concurrently to other DMA transactions to SRAM. CPU accesses (via cache miss or uncacheable access) are arbitrated with DMA accesses in hardware. There is hardware support to manage DMA request arbitration, physical memory resource arbitration, clocking and power management.

DMA configuration is programmed via the respective peripheral controller interface. Each peripheral has the same DMA capability with some minor exceptions where peripheral specific behavior is required. For example, for configuring the DMA transaction trigger, there are different trigger options for each peripheral depending on the mode of operation.

5.1.1 General Usage

The DMA controller is enabled at reset and no chip-level initialization is required (DMA may be disabled by clearing the DMA_ENABLE bit in the APBDMACTRL register in the CONTROL register block). The DMA controller automatically manages byte-aligned addresses in memory and non-word transfer lengths. While peripherals have the ability to DMA large blocks of data to/from memory, individual DMA transfers are performed at a granularity of 1-16 bytes per transfer.

To utilize DMA, software should program the peripheral's DMA control registers to enable data transfer to/from the FIFOs that would normally have been done by software. DMA-capable peripherals have been updated with additional interrupts to notify software of transfer and DMA completion events. Each peripheral also has the following common registers:

- "DMATARGADDR: Specifies the SRAM or flash address for the start of the transfer. As the transfer proceeds, the peripheral will update this address to track the current DMA location in memory.
- "DMATOTCOUNT: Specifies the total number of bytes to be transferred to/from memory. This value will also decrement throughout the transfer.
- "DMABCOUNT: Specifies the DMA "burst" size or number of bytes to be transferred each time a DMA access is triggered in the peripheral. For most optimal efficiency, this should be set to 16 or 32 bytes which would correspond to one or two actual transactions to memory.

Each peripheral also has registers to control when DMA transfers are initiated. Upon reaching the DMA threshhold, the device will request a DMA transfer and the DMA engine will perform the required number of read/write operations to move the data to/from SRAM or flash memory. Peripherals will typically issue multiple DMA read/write operations to complete a longer DMA transfer and each peripheral has a few configuration options to help software manage the flow of data. For instance, a peripheral with a 16-word FIFO might be configured to transfer 4 words each time the FIFO reaches 4 entries while the CPU is awake in order to flush data as quickly as possible while during periods of deep-sleep, the peripheral may be configured to transfer 8-12 words at a time once the FIFO reaches 12-16 entries in order to minimize the wake time of the SRAM banks.

5.1.2 Auto Power Down

The DMA-capable peripherals can be configured to automatically power down the respective peripheral device once the total DMA transaction is complete. This feature is particularly useful in cases where a device transaction can be queued up allowing the CPU to go into deep sleep while the transaction completes which could take a long time depending on the data rate of the device and/or the trigger conditions for sending/receiving data.

The auto power down mode is fully autonomous where not only is the peripheral device powered down but any associated memory is also replaced back into its lowest power state as applicable. The auto power down mode is enabled in the DMA_CFG register of the respective peripheral device register space.

5.1.3 Priority

Each DMA agent can be assigned a high priority or a “best effort” priority. This allows software to ensure a certain quality of service as required for the particular peripheral depending on the use case requirements. The peripheral also has safeguards to auto promote priority if its corresponding trigger levels are approaching critical levels. This is to ensure the respective peripheral does not overflow/underflow.

The priority settings as well as the auto promote feature are enabled in the DMA_CFG register of the respective peripheral device register space.

5.1.4 Hardware Handshake / Hardware Triggering

The IOM, BLE and MSPI peripherals include handshaking to allow coordination of data flow between the peripherals and system memory without CPU involvement by using the command queuing support in the peripheral. See the respective peripheral sections for details regarding command queuing and hardware triggering.

6. BLE Module

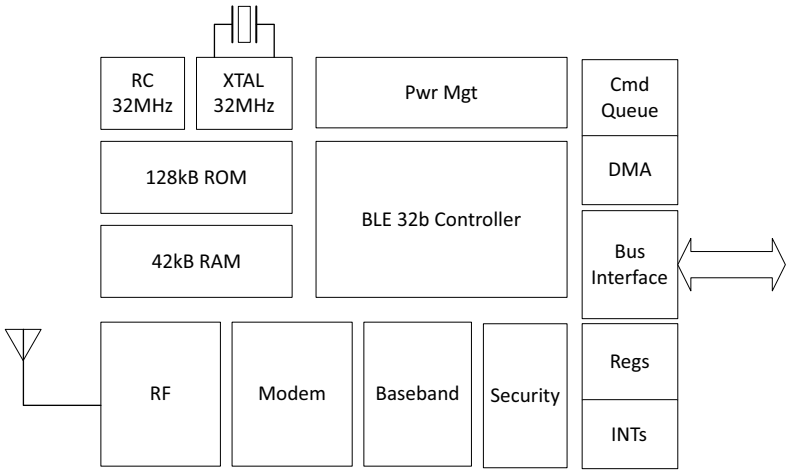


Figure 11. Block Diagram for the BLE Module

6.1 Functional Overview

6.1.1 Introduction

The Apollo3 Blue MCU includes a low power Bluetooth low energy subsystem. The BLE controller and host can be configured to support up to eight simultaneous connections. Secure connections and extended packet length are also supported.

The BLE subsystem contains a 2.4 GHz RF transceiver, modem, baseband and 32-bit processor. It supports an external 32 MHz crystal clock source as well as an internal 32 MHz RC oscillator clock source. The 32 MHz crystal is required as the frequency reference for the radio and also as the main clock source for the controller blocks. The internal 32 MHz RC can be used as a clock source for the RF processor if the requirements allow for lower precision and lower power operation. Driving an active clock into BLE crystal pins is not supported, as the crystal pins do not support active components.

The BLE subsystem provides a Host Controller Interface (HCI) to the host.

6.1.2 Main Features

The highlighted features of the BLE are as follows:

Bluetooth 5 Low Energy Technology

- Full on-chip HCI Transport Layer
- Up to eight (8) simultaneous connections supported
- Extended PDU length and enhanced security
- AES-128 Hardware Encryption Engine

Secure Firmware Over-the-Air Updating

- Per application, function or configuration

High Performance RF

- -94 dBm Bluetooth low energy transceiver sensitivity, selectivity and blocking performance
- -20 to +4 dBm transmitter output power range
- TX: 3 mA @0dBm, RX: 3 mA

- External Power Amplifier support
- Integrated Balun and antenna matching network

6.2 Functional Description

The BLE subsystem is a fully integrated system providing autonomous clock and power management. The subsystem is accessed via the BLE interface block. Software leverages the fully HCI compliant interface for Bluetooth operation. A series of proprietary HCI commands are also leveraged to provide additional performance and low power operation.

The BLE subsystem must first be enabled by issuing an enable to the BLE feature enable register (Section 3.8.2.7 on page 128). The device is then enabled by setting the BLEL controller device enable field in the power controller device register (DEVPWREN Register in System Core chapter). Once the BLEL domain is powered up through the BLEL enable, software can enable the power state machine within the BLE interface module to allow the BLEH power domain to be activated.

Communication between the BLE core and the MCU is done through the BLE interface (BLEIF) module. This module uses a similar interface as the IOM module. This module will facilitate the data transfer to and from the BLE core and supports direct and DMA data transfer mechanisms. The module also contains the power sequencing logic to control the power domains used for the BLE Core. This logic will control the initial power on, as well as power down of domains during sleep mode automatically.

The BLEIF contains flow control mechanisms that allow write transactions under control of the BSTATUS signal from the BLE core, and will similarly gate read transactions using the BLEIRQ signal from the BLE Core. These are enabled via the BLEIF_MSPICFG register fields of RDFC and WTFC.

The BLE subsystem will automatically enter into a low power sleep mode when no active commands are issued and no active Tx/Rx events.

6.2.1 Data Transfers

Data transfers to and from the BLE core are done using HCI packets. The HCI packet structure is used for both data input and output, For event frames read from the BLE Core, an optional mode is available to prepend a 2 byte length to the start of the packet. This mode is enabled with a vendor specific command.

The HCI commands and packet formats are detailed in the Bluetooth specification, version 4.2, volume 2, part E, sections 7.8.1 through 7.8.46 (LE Controller commands). Additional vendor specific commands are also available for operations such as setting the frame mode, setting sleep mode and other BLE Core specific commands.

6.2.1.1 DMA data transfers

DMA transfers are enabled by configuring the DMA related registers, enabling the DMA channel, and then issuing the command. The command will automatically fetch and store the data associated with the command without MCU intervention. The DMA channel is enabled via the DMAEN field in the REG_BLEIF_DMACFG register.P2M DMA operations transfer data from peripheral to memory, and are used in BLEIF READ operations. M2P DMA operations transfer data from memory to peripheral, and are used in BLEIF write operations. DMA transfer size is programmed into the REG_BLEIF_DMATOTCOUNT register and supports up to 4095 bytes of data transfer. The DMA transfer size is independent from the transaction size, and allows a single DMA setting to be used across multiple commands. The direction of DMA data transfer must match the command. The DMAEN field in the REG_BLEIF_DMACFG register enables/disables the DMA transfer capability and must be set last when configuring the DMA, generally prior to sending the command.

The DMA engine within the module will initiate a transfer of data when a trigger event occurs. There are 2 type of triggers available, threshold (THR) and command completion (CMDCMP). The THR trigger will activate when the threshold programmed into the FIFOWTHR or FIFORTHR in the REG_BLEIF_FIFOTHR

register meets the data criteria. Because the MCU access to the interface is 32b wide, only the word count of the selected THR is used, and the low order bits of the FIFOWTHR or FIFORTHR are ignored.

During the transfer, the TOTCOUNT register is decremented to reflect the number of bytes transferred.

For BLE write operations (data written from BLEIF into the BLE Core), the THR trigger will activate when the write FIFO contains FIFOWTHR[5:2] free words. If the remaining DMA transfer size is less than this, only the needed number of words are transferred.

For BLE read operations (data read from BLE Core from the BLEIF), the THR trigger will activate when the read FIFO contains FIFORTHR[5:2] words of valid data. If the remaining DMA transfer size is less than the RTHR words, then the CMDCMP trigger can be enabled to transfer the remaining data. If the CMDCMP trigger is disabled, and the number of bytes in the read FIFO is greater to or equal to the current TOTCOUNT, a DMA transfer of TOTCOUNT will be done to complete the DMA operation. Note that this mode requires that the THR trigger be enabled as well.

The CMDCMP trigger activates when the command is complete, and will transfer the lesser of the TOTCOUNT or the number of bytes in the read FIFO. Note, this trigger is not needed for write operations, and the THR trigger should be used in this case. If a read operation is done, and the THR trigger is disabled, and only the CMDCMP trigger is enabled, and the transaction size is greater than the FIFO size (32 bytes), the module will hang, as there is not trigger to cause a DMA operation, and the logic will pause the interface until there is room within the read FIFO to store data.

If DMA transfer size is matched to the BLEIF transaction size, it is recommended to program both the FIFORTHR and FIFOWTHR to 0x10 (16 bytes) and only enable the THR trigger.

6.2.1.2 Command Queue

The BLEIF module can also fetch register write data from SRAM or FLASH, and update the registers as if the write was performed via the MCU. Register data is stored as a doublet of 2 words. The first word is the register address offset, word aligned. The second word is the write data value. Once enabled, the command queue (CQ) will fetch the address, and perform a write to the register If no command is started by the register write, the next doublet will be fetched by the CQ. If a command is started, the transaction will run, and the CQ will continue fetching when the module is idle. No pre-fetching is done via the CQ, and the register write operations are performed in series with the transactions. This allows a predictable path for execution of commands.

6.3 BLEIF Registers

BLE Interface

INSTANCE 0 BASE ADDRESS:0x5000C000

Registers associated with the BLE Core interface module. The BLEIF module is used to interface with the embedded BLE Core module and supports read and write transactions to the BLE Core. It also contains the power sequencing control which will switch the BLEH power to the BLE core when needed. The registers control the speed of the interface, mode of operation and other parameters for the transaction. It is recommended to run at 16MHz with a mode of 3 on the SPI interface. Prior to use and access, the BLE module domain must be powered up through registers within the power control module. Once powered, the power state machine must be enabled to allow power control of the BLE Core module.

6.3.1 Register Memory Map

Table 250: BLEIF Register Map

Address(es)	Register Name	Description
0x5000C000	FIFO	FIFO Access Port
0x5000C100	FIFOPTR	FIFO size and remaining slots open values
0x5000C104	FIFOTHR	FIFO Threshold Configuration
0x5000C108	FIFOPOP	FIFO POP register
0x5000C10C	FIFOPUSH	FIFO PUSH register
0x5000C110	FIFOCTRL	FIFO Control Register
0x5000C114	FIFOLOC	FIFO Pointers
0x5000C200	CLKCFG	I/O Clock Configuration
0x5000C20C	CMD	Command and offset Register
0x5000C210	CMDRPT	Command Repeat Register
0x5000C214	OFFSETHI	High order offset bytes
0x5000C218	CMDSTAT	Command status
0x5000C220	INTEN	IO Master Interrupts: Enable
0x5000C224	INTSTAT	IO Master Interrupts: Status
0x5000C228	INTCLR	IO Master Interrupts: Clear
0x5000C22C	INTSET	IO Master Interrupts: Set
0x5000C230	DMATRIGEN	DMA Trigger Enable Register
0x5000C234	DMATRIGSTAT	DMA Trigger Status Register
0x5000C238	DMACFG	DMA Configuration Register
0x5000C23C	DMATOTCOUNT	DMA Total Transfer Count
0x5000C240	DMATARGADDR	DMA Target Address Register
0x5000C244	DMASTAT	DMA Status Register
0x5000C248	CQCFG	Command Queue Configuration Register
0x5000C24C	CQADDR	CQ Target Read Address Register
0x5000C250	CQSTAT	Command Queue Status Register
0x5000C254	CQFLAGS	Command Queue Flag Register
0x5000C258	CQSETCLEAR	Command Queue Flag Set/Clear Register
0x5000C25C	CQPAUSEEN	Command Queue Pause Enable Register
0x5000C260	CQCURIDX	IOM Command Queue current index value. Compared to the CQENDIDX reg contents to generate the IDXEQ Pause event for command queue
0x5000C264	CQENDIDX	IOM Command Queue current index value. Compared to the CQCURIDX reg contents to generate the IDXEQ Pause event for command queue
0x5000C268	STATUS	IOM Module Status Register
0x5000C300	MSPICFG	SPI module master configuration
0x5000C304	BLECFG	BLE Core Control

Table 250: BLEIF Register Map

Address(es)	Register Name	Description
0x5000C308	PWRCMD	BLE Power command interface
0x5000C30C	BSTATUS	BLE Core status
0x5000C410	BLEDBG	BLEIF Master Debug Register

6.3.2 BLEIF Registers

6.3.2.1 FIFO Register

FIFO Access Port

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x5000C000

Provides direct random access to both input and output fifos. The state of the FIFO is not disturbed by reading these locations (i.e., no POP will be done). FIFO0 is accessible from addresses 0x0 - 0x1C, and is used for data output from the IOM to external devices. These FIFO locations can be read and written directly.

Table 251: 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 252: FIFO Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFO	0x0	RW	FIFO direct access. Only locations 0 - 3F will return valid information.

6.3.2.2 FIFOPTR Register

FIFO size and remaining slots open values

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x5000C100

Provides the current valid byte count of data within the FIFO as seen from the internal state machines. FIFO0 is dedicated to outgoing transactions and FIFO1 is dedicated to incoming transactions. All counts are specified in units of bytes.

Table 253: 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
FIFO1REM								FIFO1SIZ								FIFO0REM								FIFO0SIZ							

Table 254: FIFOPTR Register Bits

Bit	Name	Reset	RW	Description
31:24	FIFO1REM	0x0	RO	The number of remaining data bytes slots currently in FIFO 1 (written by interface, read by MCU)
23:16	FIFO1SIZ	0x0	RO	The number of valid data bytes currently in FIFO 1 (written by interface, read by MCU)
15:8	FIFO0REM	0x0	RO	The number of remaining data bytes slots currently in FIFO 0 (written by MCU, read by interface)
7:0	FIFO0SIZ	0x0	RO	The number of valid data bytes currently in the FIFO 0 (written by MCU, read by interface)

6.3.2.3 FIFOTHR Register

FIFO Threshold Configuration

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x5000C104

Sets the threshold values for incoming and outgoing transactions. The threshold values are used to assert the interrupt if enabled, and also used during DMA to set the transfer size as a result of DMATHR trigger.

Table 255: 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 256: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	RESERVED
13:8	FIFOWTHR	0x0	RW	FIFO write threshold in bytes. A value of 0 will disable the write FIFO level from activating the threshold interrupt. If this field is non-zero, it will trigger a threshold interrupt when the write fifo contains FIFOWTHR free bytes, as indicated by the FIFO0REM field. This is intended to signal when a transfer of FIFOWTHR bytes can be done from the host to the IOM write fifo to support large IOM write operations.
7:6	RSVD	0x0	RO	RESERVED

Table 256: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
5:0	FIFORTHR	0x0	RW	FIFO read threshold in bytes. A value of 0 will disable the read FIFO level from activating the threshold interrupt. If this field is non-zero, it will trigger a threshold interrupt when the read fifo contains FIFORTHR valid bytes of data, as indicated by the FIFO1SIZ field. This is intended to signal when a data transfer of FIFORTHR bytes can be done from the IOM module to the host via the read fifo to support large IOM read operations.

6.3.2.4 FIFOPOP Register

FIFO POP register

OFFSET: 0x00000108

INSTANCE 0 ADDRESS: 0x5000C108

Will advance the internal read pointer of the incoming FIFO (FIFO1) when read, if POPWR is not active. If POPWR is active, a write to this register is needed to advance the internal FIFO pointer.

Table 257: FIFOPOP Register

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

Table 258: FIFOPOP Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFODOUT	0x0	RW	This register will return the read data indicated by the current read pointer on reads. If the POPWR control bit in the FIFOCTRL register is reset (0), the fifo read pointer will be advanced by one word as a result of the read.

6.3.2.5 FIFOPUSH Register

FIFO PUSH register

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x5000C10C

Will write new data into the outgoing FIFO and advance the internal write pointer.

Table 259: FIFOPUSH Register

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

Table 260: FIFOPUSH Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFODIN	0x0	RW	This register is used to write the FIFORAM in FIFO mode and will cause a push event to occur to the next open slot within the FIFORAM. Writing to this register will cause the write point to increment by 1 word(4 bytes).

6.3.2.6 FIFOCTRL Register

FIFO Control Register

OFFSET: 0x00000110

INSTANCE 0 ADDRESS: 0x5000C110

Provides controls for the operation of the internal FIFOs. Contains fields used to control the operation of the POP register, and also controls to reset the internal pointers of the FIFOs.

Table 261: FIFOCTRL Register

31	30	29	28	27	26	25	24	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																														FIFORSTN	POPWR

Table 262: FIFOCTRL Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	FIFORSTN	0x1	RW	Active low manual reset of the fifo. Write to 0 to reset fifo, and then write to 1 to remove the reset.
0	POPWR	0x0	RW	Selects the mode in which 'pop' events are done for the fifo read operations. A value of '1' will prevent a pop event on a read operation, and will require a write to the FIFOPOP register to create a pop event.

6.3.2.7 FIFOLOC Register

FIFO Pointers

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x5000C114

Provides a read only value of the current read and write pointers. This register is read only and can be used along with the FIFO direct access method to determine the next data to be used for input and output functions.

Table 263: FIFOLOC Register

31	30	29	28	27	26	25	24	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																				FIFORPTR				RSVD				FIFOWPTR			

Table 264: FIFOLOC Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	Reserved
11:8	FIFORPTR	0x0	RW	Current FIFO read pointer. Used to index into the incoming FIFO (FIFO1), which is used to store read data returned from external devices during a read operation.
7:4	RSVD	0x0	RO	Reserved
3:0	FIFOWPTR	0x0	RW	Current FIFO write pointer. Value is the index into the outgoing FIFO (FIFO0), which is used during write operations to external devices.

6.3.2.8 CLKCFG Register

I/O Clock Configuration

OFFSET: 0x00000200

INSTANCE 0 ADDRESS: 0x5000C200

Provides clock related controls used internal to the BLEIF module, and enablement of 32KHz clock to the BLE Core module. The internal clock sourced is selected via the FSEL and can be further divided by 3 using the DIV3 control.

Table 265: 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
RSVD																			DIV3	CLK32KEN	FSEL			RSVD							IOCLKEN

Table 266: CLKCFG Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	DIV3	0x0	RW	Enable of the divide by 3 of the source IOCLK.
11	CLK32KEN	0x0	RW	Enable for the 32Khz clock to the BLE module
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 IOM 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.
7:1	RSVD	0x0	RO	RESERVED
0	IOCLKEN	0x0	RW	Enable for the interface clock. Must be enabled prior to executing any IO operations.

6.3.2.9 CMD Register

Command and offset Register

OFFSET: 0x0000020C

INSTANCE 0 ADDRESS: 0x5000C20C

Writes to this register will start an IO transaction, as well as set various parameters for the command itself. Reads will return the command value written to the CMD register.

Table 267: 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
OFFSETLO								RSRVD54		CMDSEL		TSIZE												CONT	OFFSETCNT		CMD				

Table 268: CMD Register Bits

Bit	Name	Reset	RW	Description
31:24	OFFSETLO	0x0	RW	This register holds the low order byte of offset to be used in the transaction. The number of offset bytes to use is set with bits 1:0 of the command. Offset bytes are transferred starting from the highest byte first.
23:22	RSRVD54	0x0	RO	Reserved
21:20	CMDSEL	0x0	RW	Command Specific selection information
19:8	TSIZE	0x0	RW	Defines the transaction size in bytes. The offset transfer is not included in this size.
7	CONT	0x0	RW	Continue to hold the bus after the current transaction if set to a 1 with a new command issued.
6:5	OFFSETCNT	0x0	RW	Number of offset bytes to use for the command - 0, 1, 2, 3 are valid selections. The second (byte 1) and third byte (byte 2) are read from the OFFSETHI register, and the low order byte is pulled from this register in the OFFSETLO field.
4:0	CMD	0x0	RW	Command for submodule. WRITE = 0x1 - Write command using count of offset bytes specified in the OFFSETCNT field READ = 0x2 - Read command using count of offset bytes specified in the OFFSETCNT field

6.3.2.10 CMDRPT Register

Command Repeat Register

OFFSET: 0x00000210

INSTANCE 0 ADDRESS: 0x5000C210

Will repeat the next command for CMDRPT number of times. If CMDRPT is set to 1, the next command will be done 2 times in series. A repeat count of up to 31 is possible. Each command will be done as a separate command, but the data will

Table 269: CMDRPT Register

31	30	29	28	27	26	25	24	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																											CMDRPT				

Table 270: CMDRPT Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4:0	CMDRPT	0x0	RW	Count of number of times to repeat the next command.

6.3.2.11 OFFSETHI Register

High order offset bytes

OFFSET: 0x00000214

INSTANCE 0 ADDRESS: 0x5000C214

Provides the high order bytes of 2 or 3 byte offset transactions of the current command. Usage of these bytes is dependent on the offsetcnt field in the CMD register. If the offsetcnt == 3, the data located at OFFSETHI[15:0] will first be transmitted,

Table 271: OFFSETHI Register

31	30	29	28	27	26	25	24	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																OFFSETHI															

Table 272: OFFSETHI Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved
15:0	OFFSETHI	0x0	RW	Holds the high order bytes of the 2 or 3 byte offset phase of a transaction.

6.3.2.12 CMDSTAT Register

Command status

OFFSET: 0x00000218

INSTANCE 0 ADDRESS: 0x5000C218

Provides status on the execution of the command currently in progress. The fields in this register will reflect the real time status of the internal state machines and data transfers within the IOM.

Table 273: CMDSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD0												CTSIZE												CMDSTAT			CCMD				

Table 274: CMDSTAT Register Bits

Bit	Name	Reset	RW	Description
31:20	RSRVD0	0x0	RO	Reserved
19:8	CTSIZE	0x0	RO	The current number of bytes still to be transferred with this command. This field will count down to zero.
7:5	CMDSTAT	0x0	RO	The current status of the command execution. ERR = 0x1 - Error encountered with command ACTIVE = 0x2 - Actively processing command IDLE = 0x4 - Idle state, no active command, no error WAIT = 0x6 - Command in progress, but waiting on data from host
4:0	CCMD	0x0	RO	current command that is being executed

6.3.2.13 INTEN Register

IO Master Interrupts: Enable

OFFSET: 0x00000220

INSTANCE 0 ADDRESS: 0x5000C220

Set bits in this register to allow this module to generate the corresponding interrupt.

Table 275: 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															B2MSHUTDN	B2MACTIVE	B2MSLEEP	CQERR	CQUPD	CQPAUSED	DERR	DCMP	BLECSSTAT	BLECIRQ	ICMD	IACC	B2MST	FOVFL	FUNDFL	THR	CMDCMP

Table 276: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	RESERVED
16	B2MSHUTDN	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into shutdown state. Revision B: Falling BLE Core Status signal. Asserted when the BLE_STATUS signal from the BLE Core is de-asserted (1 -> 0)
15	B2MACTIVE	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into the active state. Revision B: Falling BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is de-asserted (1 -> 0)
14	B2MSLEEP	0x0	RW	The B2M_STATE from the BLE Core transitioned into the sleep state
13	CQERR	0x0	RW	Command queue error during processing. When an error occurs, the system will stop processing and halt operations to allow software to take recovery actions
12	CQUPD	0x0	RW	Command queue write operation executed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
11	CQPAUSED	0x0	RW	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
10	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
9	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
8	BLECSSTAT	0x0	RW	BLE Core SPI Status interrupt. Asserted when the SPI_STATUS signal from the BLE Core is asserted, indicating that SPI writes can be done to the BLE Core.
7	BLECIRQ	0x0	RW	BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is asserted, indicating the availability of read data from the BLE Core.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event

Table 276: INTEN Register Bits

Bit	Name	Reset	RW	Description
4	B2MST	0x0	RW	B2M State change interrupt. Asserted on any change in the B2M_STATE signal from the BLE Core.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. Asserted when a pop operation is done to a empty read FIFO.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

6.3.2.14 INTSTAT Register

IO Master Interrupts: Status

OFFSET: 0x00000224

INSTANCE 0 ADDRESS: 0x5000C224

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

Table 277: 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															B2MSHUTDN	B2MACTIVE	B2MSLEEP	CQERR	CQUPD	CQPAUSED	DERR	DCMP	BLECSSTAT	BLECIRQ	ICMD	IACC	B2MST	FOVFL	FUNDFL	THR	CMDCMP

Table 278: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	RESERVED
16	B2MSHUTDN	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into shutdown state. Revision B: Falling BLE Core Status signal. Asserted when the BLE_STATUS signal from the BLE Core is de-asserted (1 -> 0)

Table 278: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
15	B2MACTIVE	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into the active state. Revision B: Falling BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is de-asserted (1 -> 0)
14	B2MSLEEP	0x0	RW	The B2M_STATE from the BLE Core transitioned into the sleep state
13	CQERR	0x0	RW	Command queue error during processing. When an error occurs, the system will stop processing and halt operations to allow software to take recovery actions
12	CQUPD	0x0	RW	Command queue write operation executed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
11	CQPAUSED	0x0	RW	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
10	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
9	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
8	BLECSSTAT	0x0	RW	BLE Core SPI Status interrupt. Asserted when the SPI_STATUS signal from the BLE Core is asserted, indicating that SPI writes can be done to the BLE Core.
7	BLECIRQ	0x0	RW	BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is asserted, indicating the availability of read data from the BLE Core.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	B2MST	0x0	RW	B2M State change interrupt. Asserted on any change in the B2M_STATE signal from the BLE Core.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. Asserted when a pop operation is done to a empty read FIFO.

Table 278: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

6.3.2.15 INTCLR Register

IO Master Interrupts: Clear

OFFSET: 0x00000228

INSTANCE 0 ADDRESS: 0x5000C228

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 279: 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															B2MSHUTDN	B2MACTIVE	B2MSLEEP	CQERR	CQUPD	CQPAUSED	DERR	DCMP	BLECSSTAT	BLECIRQ	ICMD	IACC	B2MST	FOVFL	FUNDFL	THR	CMDCMP

Table 280: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	RESERVED
16	B2MSHUTDN	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into shutdown state. Revision B: Falling BLE Core Status signal. Asserted when the BLE_STATUS signal from the BLE Core is de-asserted (1 -> 0)
15	B2MACTIVE	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into the active state. Revision B: Falling BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is de-asserted (1 -> 0)
14	B2MSLEEP	0x0	RW	The B2M_STATE from the BLE Core transitioned into the sleep state
13	CQERR	0x0	RW	Command queue error during processing. When an error occurs, the system will stop processing and halt operations to allow software to take recovery actions

Table 280: INTCLR Register Bits

Bit	Name	Reset	RW	Description
12	CQUPD	0x0	RW	Command queue write operation executed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
11	CQPAUSED	0x0	RW	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
10	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
9	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
8	BLECSSTAT	0x0	RW	BLE Core SPI Status interrupt. Asserted when the SPI_STATUS signal from the BLE Core is asserted, indicating that SPI writes can be done to the BLE Core.
7	BLECIRQ	0x0	RW	BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is asserted, indicating the availability of read data from the BLE Core.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	B2MST	0x0	RW	B2M State change interrupt. Asserted on any change in the B2M_STATE signal from the BLE Core.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. Asserted when a pop operation is done to a empty read FIFO.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

6.3.2.16 INTSET Register

IO Master Interrupts: Set

OFFSET: 0x0000022C

INSTANCE 0 ADDRESS: 0x5000C22C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 281: 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															B2MSHUTDN	B2MACTIVE	B2MSLEEP	CQERR	CQUPD	CQPAUSED	DERR	DCMP	BLECSSTAT	BLECIRQ	ICMD	IACC	B2MST	FOVFL	FUNDFL	THR	CMDCMP

Table 282: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:17	RSVD	0x0	RO	RESERVED
16	B2MSHUTDN	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into shutdown state Revision B: Falling BLE Core Status signal. Asserted when the BLE_STATUS signal from the BLE Core is de-asserted (1 -> 0)
15	B2MACTIVE	0x0	RW	Revision A: The B2M_STATE from the BLE Core transitioned into the active state. Revision B: Falling BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is de-asserted (1 -> 0)
14	B2MSLEEP	0x0	RW	The B2M_STATE from the BLE Core transitioned into the sleep state
13	CQERR	0x0	RW	Command queue error during processing. When an error occurs, the system will stop processing and halt operations to allow software to take recovery actions
12	CQUPD	0x0	RW	Command queue write operation executed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
11	CQPAUSED	0x0	RW	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
10	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.

Table 282: INTSET Register Bits

Bit	Name	Reset	RW	Description
9	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
8	BLECSSTAT	0x0	RW	BLE Core SPI Status interrupt. Asserted when the SPI_STATUS signal from the BLE Core is asserted, indicating that SPI writes can be done to the BLE Core.
7	BLECIRQ	0x0	RW	BLE Core IRQ signal. Asserted when the BLE_IRQ signal from the BLE Core is asserted, indicating the availability of read data from the BLE Core.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	B2MST	0x0	RW	B2M State change interrupt. Asserted on any change in the B2M_STATE signal from the BLE Core.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. Asserted when a pop operation is done to a empty read FIFO.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

6.3.2.17 DMATRIGEN Register

DMA Trigger Enable Register

OFFSET: 0x00000230

INSTANCE 0 ADDRESS: 0x5000C230

Provides control on which event will trigger the DMA transfer after the DMA operation is setup and enabled. The trigger event will cause a number of bytes (depending on trigger event) to be

Table 283: DMATRIGEN Register

31	30	29	28	27	26	25	24	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																														DTHREN	DCMDCMPEN

Table 284: DMATRIGEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DTHREN	0x0	RW	Trigger DMA upon THR level reached. For M2P DMA operations (IOM writes), the trigger will assert when the write FIFO has (WTHR/4) number of words free in the write FIFO, and will transfer (WTHR/4) number of words
0	DCMDCMPEN	0x0	RW	Trigger DMA upon command complete. Enables the trigger of the DMA when a command is completed. When this event is triggered, the number of words transferred will be the lesser of the remaining TOTCOUNT bytes, or the number of bytes in the FIFO when the command completed. If this is disabled, and the number of bytes in the FIFO is equal or greater than the TOTCOUNT bytes, a transfer of TOTCOUNT bytes will be done to ensure read data is stored when the DMA is completed.

6.3.2.18 DMATRIGSTAT Register

DMA Trigger Status Register

OFFSET: 0x00000234

INSTANCE 0 ADDRESS: 0x5000C234

Provides the status of trigger events that have occurred for the transaction. Some of the bits are read only and some can be reset via a write of 0.

Table 285: DMATRIGSTAT Register

31	30	29	28	27	26	25	24	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																													DTOTCMP	DTHR	DCMDCMP

Table 286: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	DTOTCMP	0x0	RO	DMA triggered when DCMDCMP = 0, and the amount of data in the FIFO was enough to complete the DMA operation (greater than or equal to current TOTCOUNT) when the command completed. This trigger is default active when the DCMDCMP trigger is
1	DTHR	0x0	RO	Triggered DMA from THR event. Bit is read only and can be cleared by disabling the DTHR trigger enable or by disabling DMA.
0	DCMDCMP	0x0	RO	Triggered DMA from Command complete event. Bit is read only and can be cleared by disabling the DCMDCMP trigger enable or by disabling DMA.

6.3.2.19 DMACFG Register

DMA Configuration Register

OFFSET: 0x00000238

INSTANCE 0 ADDRESS: 0x5000C238

Configuration control of the DMA process, including the direction of DMA, and enablement of DMA

Table 287: DMACFG Register

31	30	29	28	27	26	25	24	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																						DPWROFF	DMAPRI	RSVD						DMADIR	DMAEN

Table 288: DMACFG Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED.
9	DPWROFF	0x0	RW	Power off module after DMA is complete. If this bit is active, the module will request to power off the supply it is attached to. If there are other units still requiring power from the same domain, power down will not be performed. DIS = 0x0 - Power off disabled EN = 0x1 - Power off enabled
8	DMAPRI	0x0	RW	Sets the Priority of the DMA request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)

Table 288: DMACFG Register Bits

Bit	Name	Reset	RW	Description
7:2	RSVD	0x0	RO	RESERVED.
1	DMADIR	0x0	RW	Direction P2M = 0x0 - Peripheral to Memory (SRAM) transaction. To be set when doing IOM read operations, i.e., reading data from external devices. M2P = 0x1 - Memory to Peripheral transaction. To be set when doing IOM write operations, i.e., writing data to external devices.
0	DMAEN	0x0	RW	DMA Enable. Setting this bit to EN will start the DMA operation. This should be the last DMA related register set prior to issuing the command DIS = 0x0 - Disable DMA Function EN = 0x1 - Enable DMA Function

6.3.2.20 DMATOTCOUNT Register

DMA Total Transfer Count

OFFSET: 0x0000023C

INSTANCE 0 ADDRESS: 0x5000C23C

Contains the number of bytes to be transferred for this DMA transaction. This register is decremented as the data is transferred, and will be 0 at the completion of the DMA operation.

Table 289: DMATOTCOUNT Register

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

Table 290: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:12	RSRVDD	0x0	RO	Reserved
11:0	TOTCOUNT	0x0	RW	Triggered DMA from Command complete event occurred. Bit is read only and can be cleared by disabling the DTHR trigger enable or by disabling DMA.

6.3.2.21 DMATARGADDR Register

DMA Target Address Register

OFFSET: 0x00000240

INSTANCE 0 ADDRESS: 0x5000C240

The source or destination address internal the SRAM for the DMA data. For write operations, this can only be SRAM data (ADDR bit 28 = 1); For read operations, this can be either SRAM or FLASH (ADDR bit 28 = 0)

Table 291: DMATARGADDR Register

31	30	29	28	27	26	25	24	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			TARGADDR28	RSVD								TARGADDR																			

Table 292: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	Reserved
28	TARGADDR28	0x0	RW	Bit 28 of the target byte address for source of DMA (either read or write). In cases of non-word aligned addresses, the DMA logic will take care for ensuring only the target bytes are read/written.
27:20	RSVD	0x0	RO	Reserved
19:0	TARGADDR	0x0	RW	Bits [19:0] of the target byte address for source of DMA (either read or write). The address can be any byte alignment, and does not have to be word aligned. In cases of non-word aligned addresses, the DMA logic will take care for ensuring only the target bytes are read/written.

6.3.2.22 DMASTAT Register

DMA Status Register

OFFSET: 0x00000244

INSTANCE 0 ADDRESS: 0x5000C244

Status of the DMA operation currently in progress.

Table 293: DMASTAT Register

31	30	29	28	27	26	25	24	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																													DMAERR	DMACPL	DMATIP

Table 294: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	DMAERR	0x0	RW	DMA Error. This active high bit signals that an error was encountered during the DMA operation.
1	DMACPL	0x0	RW	DMA Transfer Complete. This signals the end of the DMA operation. This bit can be cleared by writing to 0.
0	DMATIP	0x0	RO	DMA Transfer In Progress indicator. 1 will indicate that a DMA transfer is active. The DMA transfer may be waiting on data, transferring data, or waiting for priority.

6.3.2.23 CQCFG Register

Command Queue Configuration Register

OFFSET: 0x00000248

INSTANCE 0 ADDRESS: 0x5000C248

Controls parameters and options for execution of the command queue operation. To enable command queue, create this in memory, set the address, and enable it with a write to CQEN

Table 295: CQCFG Register

31	30	29	28	27	26	25	24	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																														CQPRI	CQEN

Table 296: CQCFG Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	CQPRI	0x0	RW	Sets the Priority of the command queue dma request. LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
0	CQEN	0x0	RW	Command queue enable. When set, will enable the processing of the command queue and fetches of address/data pairs will proceed from the word address within the CQADDR register. Can be disabled DIS = 0x0 - Disable CQ Function EN = 0x1 - Enable CQ Function

6.3.2.24 CQADDR Register

CQ Target Read Address Register

OFFSET: 0x0000024C

INSTANCE 0 ADDRESS: 0x5000C24C

The SRAM address which will be fetched next execution of the CQ operation. This register is updated as the CQ operation progresses, and is the live version of the register. The register can also be

Table 297: CQADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD2			CQADDR28	RSRVD1								CQADDR																		RSRVD0	

Table 298: CQADDR Register Bits

Bit	Name	Reset	RW	Description
31:29	RSRVD2	0x0	RO	Reserved
28	CQADDR28	0x0	RW	Bit 28 of target byte address for source of CQ. Used to denote Flash (0) or SRAM (1) access
27:20	RSRVD1	0x0	RO	Reserved
19:2	CQADDR	0x0	RW	Bits 19:2 of target byte address for source of CQ. The buffer must be aligned on a word boundary
1:0	RSRVD0	0x0	RO	Reserved

6.3.2.25 CQSTAT Register

Command Queue Status Register

OFFSET: 0x00000250

INSTANCE 0 ADDRESS: 0x5000C250

Provides the status of the command queue operation. If the command queue is disabled, these bits will be cleared. The bits are read only

Table 299: CQSTAT Register

31	30	29	28	27	26	25	24	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																													CQERR	CQPAUSED	CQTIP

Table 300: CQSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	CQERR	0x0	RW	Command queue processing error. This active high bit signals that an error was encountered during the CQ operation.
1	CQPAUSED	0x0	RO	Command queue operation is currently paused.
0	CQTIP	0x0	RO	Command queue Transfer In Progress indicator. 1 will indicate that a CQ transfer is active and this will remain active even when paused waiting for external event.

6.3.2.26 CQFLAGS Register

Command Queue Flag Register

OFFSET: 0x00000254

INSTANCE 0 ADDRESS: 0x5000C254

Provides the current status of the SWFLAGS (bits 7:0) and the hardware generated flags (15:8). A '1' will pause the CQ operation if it the same bit is enabled in the CQPAUSEEN register

Table 301: CQFLAGS Register

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

Table 302: CQFLAGS Register Bits

Bit	Name	Reset	RW	Description
31:16	CQIRQMASK	0x0	RW	Provides for a per-bit mask of the flags used to invoke an interrupt. A '1' in the bit position will enable the pause event to trigger the interrupt, if the CQWT_int interrupt is enabled.
15:0	CQFLAGS	0x0	RO	Current flag status (read-only). Bits [7:0] are software controllable and bits [15:8] are hardware status.

6.3.2.27 CQSETCLEAR Register

Command Queue Flag Set/Clear Register

OFFSET: 0x00000258

INSTANCE 0 ADDRESS: 0x5000C258

Set/Clear the command queue software pause flags on a per-bit basis. Contains 3 fields, allowing for setting, clearing or toggling the value in the software flags. Priority when the same bit

Table 303: CQSETCLEAR Register

31	30	29	28	27	26	25	24	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								CQFCLR								CQFTGL								CQFSET							

Table 304: CQSETCLEAR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	Reserved
23:16	CQFCLR	0x0	WO	Clear CQFlag status bits. Will clear to 0 any SWFLAG with a '1' in the corresponding bit position of this field
15:8	CQFTGL	0x0	WO	Toggle the indicated bit. Will toggle the value of any SWFLAG with a '1' in the corresponding bit position of this field
7:0	CQFSET	0x0	WO	Set CQFlag status bits. Will set to 1 the value of any SWFLAG with a '1' in the corresponding bit position of this field

6.3.2.28 CQPAUSEEN Register

Command Queue Pause Enable Register

OFFSET: 0x0000025C

INSTANCE 0 ADDRESS: 0x5000C25C

Enables a flag to pause an active command queue operation. If a bit is '1' and the corresponding bit in the CQFLAG register is '1', CQ processing will halt until either value is changed to '0'.

Table 305: CQPAUSEEN Register

31	30	29	28	27	26	25	24	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																CQPEN															

Table 306: CQPAUSEEN Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved
15:0	CQPEN	0x0	RW	Enables the specified event to pause command processing when active CNTEQ = 0x8000 - Pauses command queue processing when HWCNT matches SWCNT BLEXOREN = 0x4000 - Pause command queue when input BLE bit XORed with SWFLAG4 is '1' IOMXOREN = 0x2000 - Pause command queue when input IOM bit XORed with SWFLAG3 is '1' GPIOXOREN = 0x1000 - Pause command queue when input GPIO irq_bit XORed with SWFLAG2 is '1' MSPI1XNOREN = 0x800 - Pause command queue when input MSPI1 bit XNORed with SWFLAG1 is '1' MSPI0XNOREN = 0x400 - Pause command queue when input MSPI0 bit XNORed with SWFLAG0 is '1' MSPI1XOREN = 0x200 - Pause command queue when input MSPI1 bit XORed with SWFLAG1 is '1' MSPI0XOREN = 0x100 - Pause command queue when input MSPI0 bit XORed with SWFLAG0 is '1' SWFLAGEN7 = 0x80 - Pause the command queue when software flag bit 7 is '1'. SWFLAGEN6 = 0x40 - Pause the command queue when software flag bit 7 is '1' SWFLAGEN5 = 0x20 - Pause the command queue when software flag bit 7 is '1' SWFLAGEN4 = 0x10 - Pause the command queue when software flag bit 7 is '1' SWFLAGEN3 = 0x8 - Pause the command queue when software flag bit 7 is '1' SWFLAGEN2 = 0x4 - Pause the command queue when software flag bit 7 is '1' SWFLAGEN1 = 0x2 - Pause the command queue when software flag bit 7 is '1' SWFLGEN0 = 0x1 - Pause the command queue when software flag bit 7 is '1'

6.3.2.29 CQCURIDX Register

IOM Command Queue current index value. Compared to the CQENDIDX reg contents to generate the IDXEQ Pause event for command queue

OFFSET: 0x00000260

INSTANCE 0 ADDRESS: 0x5000C260

Current index value, targeted to be written by register write operations within the command queue. This is compared to the CQENDIDX and will stop the CQ operation if bit 15 of the CQPAUSEEN is '1' and

Table 307: CQCURIDX Register

31	30	29	28	27	26	25	24	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																								CQCURIDX							

Table 308: CQCURIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:0	CQCURIDX	0x0	RW	Holds 8 bits of data that will be compared with the CQENDIX register field. If the values match, the IDXEQ pause event will be activated, which will cause the pausing of command queue operation if the IDXEQ bit is enabled in CQPAUSEEN.

6.3.2.30 CQENDIDX Register

IOM Command Queue current index value. Compared to the CQCURIDX reg contents to generate the IDXEQ Pause event for command queue

OFFSET: 0x00000264

INSTANCE 0 ADDRESS: 0x5000C264

End index value, targeted to be written by software to indicate the last valid register pair contained within the command queue. register write operations within the command queue.

Table 309: CQENDIDX Register

31	30	29	28	27	26	25	24	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																								CQENDIDX							

Table 310: CQENDIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED

Table 310: CQENDIDX Register Bits

Bit	Name	Reset	RW	Description
7:0	CQENDIDX	0x0	RW	Holds 8 bits of data that will be compared with the CQCURIX register field. If the values match, the IDXEQ pause event will be activated, which will cause the pausing of command queue operation if the IDXEQ bit is enabled in CQPAUSEEN.

6.3.2.31 STATUS Register

IOM Module Status Register

OFFSET: 0x00000268

INSTANCE 0 ADDRESS: 0x5000C268

General status of the IOM module command execution.

Table 311: 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 312: STATUS Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	IDLEST	0x0	RO	indicates if the active I/O state machine is IDLE. Note - The state machine could be in idle state due to hold-offs from data availability, or as the command gets propagated into the logic from the registers. IDLE = 0x1 - The I/O state machine is in the idle state.
1	CMDACT	0x0	RO	Indicates if the active I/O Command is currently processing a transaction, or command is complete, but the FIFO pointers are still synchronizing internally. This bit will go high at ACTIVE = 0x1 - An I/O command is active. Indicates the active module has an active command and is processing this. De-asserted when the command is completed.
0	ERR	0x0	RO	Bit has been deprecated. Please refer to the other error indicators. This will always return 0. ERROR = 0x1 - Bit has been deprecated and will always return 0.

6.3.2.32 MSPICFG Register

SPI module master configuration

OFFSET: 0x00000300

INSTANCE 0 ADDRESS: 0x5000C300

Controls the configuration of the SPI master module, including POL/PHA, LSB, flow control, and delays for MISO and MOSI

Table 313: MSPICFG Register

31	30	29	28	27	26	25	24	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	MSPIRST	DOUTDLY			DINDLY			SPILSB	RDFCPOL	WTF-	RSVD			RDFC	WTFC	RSVD													FULLDUP	SPHA	SPOL

Table 314: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	MSPIRST	0x1	RW	Bit is deprecated. setting it will have no effect.
29:27	DOUTDLY	0x0	RW	Delay tap to use for the output signal (MOSI). This give more hold time on the output data.
26:24	DINDLY	0x0	RW	Delay tap to use for the input signal (MISO). This gives more hold time on the input data.
23	SPILSB	0x0	RW	Selects data transfer as MSB first (0) or LSB first (1) for the data portion of the SPI transaction. The offset bytes are always transmitted MSB first. MSB = 0x0 - Send and receive MSB bit first LSB = 0x1 - Send and receive LSB bit first
22	RDFCPOL	0x0	RW	Selects the read flow control signal polarity. When set, the clock will be held low until the flow control is de-asserted. NORMAL = 0x0 - SPI_STATUS signal from BLE Core high(1) creates flow control and new read spi transactions will not be started until the signal goes low.(default) INVERTED = 0x1 - SPI_STATUS signal from BLE Core low(0) creates flow control and new read spi transactions will not be started until the signal goes high.
21	WTFCPOL	0x0	RW	Selects the write flow control signal polarity. The transfers are halted when the selected flow control signal is OPPOSITE polarity of this bit. (For example: WTFCPOL = 0 will allow a SPI_STATUS=1 to pause transfers). NORMAL = 0x0 - SPI_STATUS signal from BLE Core high(1) creates flow control and new write spi transactions will not be started until the signal goes low.(default) INVERTED = 0x1 - SPI_STATUS signal from BLE Core high(1) creates low(0) control and new write spi transactions will not be started until the signal goes high.
20:18	RSVD	0x0	R0	Reserved

Table 314: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
17	RDFC	0x0	RW	Enables flow control of new read transactions based on the SPI_STATUS signal from the BLE Core. DIS = 0x0 - Read mode flow control disabled. EN = 0x1 - Read mode flow control enabled.
16	WTFC	0x0	RW	Enables flow control of new write transactions based on the SPI_STATUS signal from the BLE Core. DIS = 0x0 - Write mode flow control disabled. EN = 0x1 - Write mode flow control enabled.
15:3	RSVD	0x0	R0	Reserved
2	FULLDUP	0x0	RW	Full Duplex mode. Capture read data during writes operations
1	SPHA	0x0	RW	Selects the SPI phase; When 1, will shift the sampling edge by 1/2 clock. SAMPLE_LEADING_EDGE = 0x0 - Sample on the leading (first) clock edge, rising or falling dependent on the value of SPOL SAMPLE_TRAILING_EDGE = 0x1 - Sample on the trailing (second) clock edge, rising of falling dependent on the value of SPOL
0	SPOL	0x0	RW	This bit selects SPI polarity. CLK_BASE_0 = 0x0 - The initial value of the clock is 0. CLK_BASE_1 = 0x1 - The initial value of the clock is 1.

6.3.2.33 BLECFG Register

BLE Core Control

OFFSET: 0x00000304

INSTANCE 0 ADDRESS: 0x5000C304

Provides control of isolation and IO signals between the interface module and the BLE Core.

Table 315: BLECFG Register

31	30	29	28	27	26	25	24	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																SPIISOCTL		PWRISOCTL		STAYASLEEP	FRCCLK	MCUFRCSLP	WT4ACTOFF	BLEHREQCTL		DCDCFLGCTL		WAKEUPCTL		BLERSTN	PWRSMEN

Table 316: BLECFG Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED

Table 316: BLECFG Register Bits

Bit	Name	Reset	RW	Description
15:14	SPIISOCTL	0x0	RW	Configuration of BLEH isolation controls for SPI related signals. ON = 0x3 - SPI signals from BLE Core to/from MCU Core are isolated. OFF = 0x2 - SPI signals from BLE Core to/from MCU Core are not isolated. AUTO = 0x0 - SPI signals from BLE Core to/from MCU Core are automatically isolated by the logic
13:12	PWRISOCTL	0x0	RW	Configuration of BLEH isolation control for power related signals. ON = 0x3 - BLEH power signal isolation to on (isolated). OFF = 0x2 - BLEH power signal isolation to off (not isolated). AUTO = 0x0 - BLEH Power signal isolation is controlled automatically through the interface logic
11	STAYASLEEP	0x0	RW	Set to prevent the BLE power control module from waking up the BLE Core after going into power down. To be used for graceful shutdown, set by software prior to powering off and will allow assertion of reset from sleep state.
10	FRCCLK	0x0	RW	Force the clock in the BLEIF to be always running
9	MCUFRCSLP	0x0	RW	Force power state machine to go to the sleep state. Intended for debug only. Has no effect on the actual BLE Core state, only the state of the BLEIF interface state machine.
8	WT4ACTOFF	0x0	RW	Debug control of BLEIF power state machine. Allows transition into the active state in the BLEIF state without waiting for dcdc req from BLE Core.
7:6	BLEHREQCTL	0x0	RW	BLEH power on request override. The value of this field will be sent to the BLE Core when the PWRSM is off. Otherwise, the value is supplied from internal logic. ON = 0x3 - BLEH Power-on reg signal is set to on (1). OFF = 0x2 - BLEH Power-on signal is set to off (0). AUTO = 0x0 - BLEH Power-on signal is controlled by the PWRSM logic and automatically controlled
5:4	DCDCFLGCTL	0x0	RW	DCDCFLG signal override. The value of this field will be sent to the BLE Core when the PWRSM is off. Otherwise, the value is supplied from internal logic. ON = 0x3 - DCDC Flag signal is set to on (1). OFF = 0x2 - DCDC Flag signal is set to off (0). AUTO = 0x0 - DCDC Flag signal is controlled by the PWRSM logic and automatically controlled
3:2	WAKEUPCTL	0x0	RW	WAKE signal override. Controls the source of the WAKE signal to the BLE Core. ON = 0x3 - Wake signal is set to on (1). OFF = 0x2 - Wake signal is set to off (0). AUTO = 0x0 - Wake signal is controlled by the PWRSM logic and automatically controlled
1	BLERSTN	0x0	RW	Reset line to the BLE Core. This will reset the BLE core when asserted ('0') and must be written to '1' prior to performing any BTLE related operations to the core. ACTIVE = 0x1 - The reset signal is active (0) INACTIVE = 0x0 - The reset signal is inactive (1)

Table 316: BLECFG Register Bits

Bit	Name	Reset	RW	Description
0	PWRSMEN	0x0	RW	Enable the power state machine for automatic sequencing and control of power states of the BLE Core module. ON = 0x1 - Internal power state machine is enabled and will sequence the BLEH power domain as indicated in the design document. Overrides for the power signals are not enabled. OFF = 0x0 - Internal power state machine is disabled and will not sequence the BLEH power domain. The values of the overrides will be used to drive the output sequencing signals

6.3.2.34 PWRCMD Register

BLE Power command interface

OFFSET: 0x00000308

INSTANCE 0 ADDRESS: 0x5000C308

Sends power related commands to the power state machine in the BLE IF module.

Table 317: PWRCMD Register

31	30	29	28	27	26	25	24	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																														RESTART	WAKEREQ

Table 318: PWRCMD Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	Reserved
1	RESTART	0x0	WO	Restart the BLE Core after going into the shutdown state. Only valid when in the shutdown state.
0	WAKEREQ	0x0	WO	Wake request from the MCU. When asserted (1), the BLE Interface logic will assert the wakeup request signal to the BLE Core. Only recognized when in the sleep state

6.3.2.35 BSTATUS Register

BLE Core status

OFFSET: 0x0000030C

INSTANCE 0 ADDRESS: 0x5000C30C

Status of the BLE Core interface signals

Table 319: BSTATUS Register

31	30	29	28	27	26	25	24	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																			BLEHREQ	BLEHACK	PWRST			BLEIRQ	WAKEUP	DCDCFLAG	DCDCREQ	SPISTATUS	B2MSTATE		

Table 320: BSTATUS Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	BLEHREQ	0x0	RO	Value of the BLEHREQ signal to the power control unit. The BLEHREQ signal is sent from the BLEIF module to the power control module to request the BLEH power up. When the BLEHACK signal is asserted,
11	BLEHACK	0x0	RO	Value of the BLEHACK signal from the power control unit. If the signal is '1', the BLEH power is active and ready for use.
10:8	PWRST	0x0	RO	Current status of the power state machine OFF = 0x0 - Internal power state machine is disabled and will not sequence the BLEH power domain. The values of the overrides will be used to drive the output sequencing signals INIT = 0x1 - Initialization state. BLEH not powered PWRON = 0x2 - Waiting for the power-up of the BLEH ACTIVE = 0x3 - The BLE Core is powered and active SLEEP = 0x6 - The BLE Core has entered sleep mode and the power request is inactive SHUTDOWN = 0x4 - The BLE Core is in shutdown mode
7	BLEIRQ	0x0	RO	Status of the BLEIRQ signal from the BLE Core. A value of 1 indicates that read data is available in the core and a read operation needs to be performed.
6	WAKEUP	0x0	RO	Value of the WAKEUP signal to the BLE Core. The WAKEUP signals is sent from the BLEIF to the BLECORE to request the BLE Core transition from sleep state to active state.
5	DCDCFLAG	0x0	RO	Value of the DCDCFLAG signal to the BLE Core. The DCDCFLAG is a signal to the BLE Core indicating that the BLEH power is active.
4	DCDCREQ	0x0	RO	Value of the DCDCREQ signal from the BLE Core. The DCDCREQ signal is sent from the core to the BLEIF module when the BLE core requires BLEH power to be active. When activated, this is
3	SPISTATUS	0x0	RO	Value of the SPISTATUS signal from the BLE Core. The signal is asserted when the BLE Core is able to accept write data via the SPI interface. Data should be transmitted to the

Table 320: BSTATUS Register Bits

Bit	Name	Reset	RW	Description
2:0	B2MSTATE	0x0	RO	State of the BLE Core logic. RESET = 0x0 - Reset State Shutdown = 0x0 - Shutdown state Sleep = 0x1 - Sleep state. Standby = 0x2 - Standby State Idle = 0x3 - Idle state Active = 0x4 - Active state.

6.3.2.36 BLEDBG Register

BLEIF Master Debug Register

OFFSET: 0x00000410

INSTANCE 0 ADDRESS: 0x5000C410

Debug control

Table 321: BLEDBG Register

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

Table 322: BLEDBG Register Bits

Bit	Name	Reset	RW	Description
31:3	DBGDATA	0x0	RW	Debug data
2	APBCLKON	0x0	RW	APBCLK debug clock control. Enable APB_CLK to be active when this bit is '1'. Otherwise, the clock is controlled with gating from the logic as needed.
1	IOCLKON	0x0	RW	IOCLK debug clock control. Enable IO_CLK to be active when this bit is '1'. Otherwise, the clock is controlled with gating from the logic as needed.
0	DBGEN	0x0	RW	Debug Enable. Setting this bit will enable the update of data within this register, otherwise it is clock gated for power savings

7. MSPI Master Module

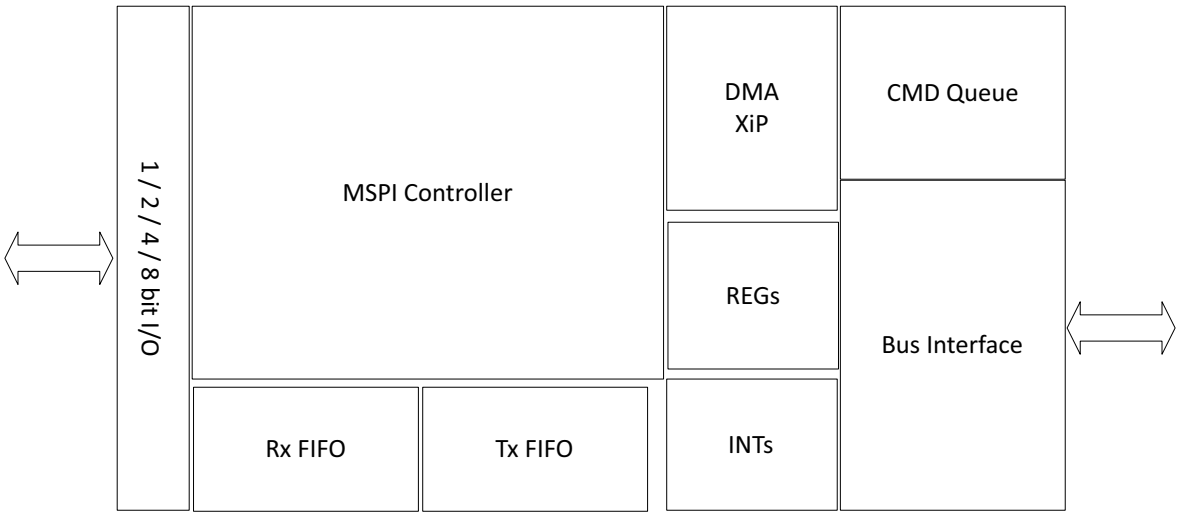


Figure 12. Block Diagram for the MSPI Master Module

7.1 Functional Overview

The Apollo3 MCU includes a Multi-bit SPI (MSPI) module which can be used to connect to external serial memory devices. It supports operation up to 24 MHz, all four SPI CPOL/CPHA modes, and can transfer in serial, dual, quad, and octal modes (with a single octal device or a pair of quad devices). The MSPI module has a unified 16-entry FIFO (32 bits wide) that is used for both transmit and receive data. To ensure that transactions are not dropped because of system or software latency, the MSPI controller will pause the clock (and thus the transfer on the bus) if the TX FIFO empties or the RX FIFO fills during an operation. It will automatically resume once the FIFO condition has cleared.

MSPI transfers generally consist of transmitting a 1 byte instruction, a 1-4 byte address (optional), and 1 byte to 64KB of write or read data (with an optional number of turnaround clock cycles between address and RX data). Access to flash devices are supported through PIO operations (primarily for configuration operations), through DMA operations to automatically transfer data blocks to/from the flash, and through an XIP mode, where instructions/data can be accessed in the external flash memory through an aperture in the integrated flash cache. The MSPI module also supports data scrambling of external accesses within an address window having boundaries aligned to 64K address blocks.

Once the external devices are configured, the MSPI supports a simple DMA model, where software can program the internal (SRAM or flash) address and external device address, transfer direction, and transfer size. Once enabled, the MSPI DMA interface will move data between the system and external flash and interrupt when complete. The MSPI also supports a higher-level command queuing (CQ) protocol, where software can construct a buffer of operations in SRAM (or internal flash memory) and the MSPI will execute the series of operations autonomously. The MSPI can also power itself down at the end of DMA or CQ operations.

While the MSPI module can be used as a generic SPI device (with two chip enables), it is primarily designed to support serial NAND/NOR flash memory and is intended to be used to initialize the external memory devices and then configured with the parameters matching the flash access characteristics. Devices can then be accessed through DMA or XIP operations with minimal software overhead.

7.2 Configuration

The MSPI module should be configured to match the transfer characteristics of the external device(s) on the bus. Generally, the configuration sequence would proceed as follows:

- Configure MSPI clock divider (MSPICFG register). The MSPI's reference frequency is 48MHz, so the resulting clock frequency is 48/CLKDIV value.
- Configure MSPI transfer characteristics (CFG register) to initialize the device (usually mode 0, serial transfers)
- Configure MSPI PADOUTEN to enable the desired bits on the MSPI bus (clock plus relevant data bits). NOTE: Enabling unused data lines will impact the values present on those pads even if the GPIO function select is not set to MSPI.
- Program external flash device to the appropriate mode, enable dual/quad/octal modes
- Update CFG register to new settings (in cases of a transfer mode or addressing change)
- Write FLASH register to set read/write instructions and transfer characteristics for DMA/XIP operations (and optionally enable XIP mode).

The MSPI's CFG register contains the controller's settings when communicating with any given device and it is expected that these values will be static after initial configuration of the external memory devices. The DEVCFG field specifies both the transfer mode (serial, dual, quad, etc) as well as which chip enable is used to access the device. The ISIZE and ASIZE fields indicate the number of bytes transmitted for the instruction and address phases, but individual operations can select whether to transmit these or not. The TURNAROUND field indicates the number of cycles between the TX of instruction/address and reception of the first RX byte (the flash device must be programmed to use the same count). Finally, the CPOL and CPHA fields indicate the settings for the clock polarity and clock phase settings, which are often referenced in literature as SPI modes 0-3. Most memory devices utilize mode 0 (CPOL=0, CPHA=0).

7.3 PIO Operations

Software can issue general PIO operations to devices on the MSPI bus using the INSTR, ADDR, and CTRL registers. Software should first write the instruction to be sent to the INSTR register and the address to be sent to the ADDR register (if required) followed by a write to the CTRL register to start the transfer. The TXRX bit indicates whether data should flow to or from the device and XFERBYTES indicates the number of bytes to transfer. SENDI and SENDA can be used to enable or disable the instruction or address phases and the ENTURN is used to enable the turnaround phase. The transfer will only commence if the START bit is set. Software may read the BUSY and STATUS fields to check on transaction status, otherwise the CMDCPL interrupt can be used to indicate completion.

```
AM_REG(MSPI,INSTR) = instr;
AM_REG(MSPI,CTRL) = AM_REG_MSPI_CTRL_XFERBYTES(bytes) |
                    AM_REG_MSPI_CTRL_SENDI_M  |
                    AM_REG_MSPI_CTRL_TXRX(1)  |
                    AM_REG_MSPI_CTRL_START_M;
```

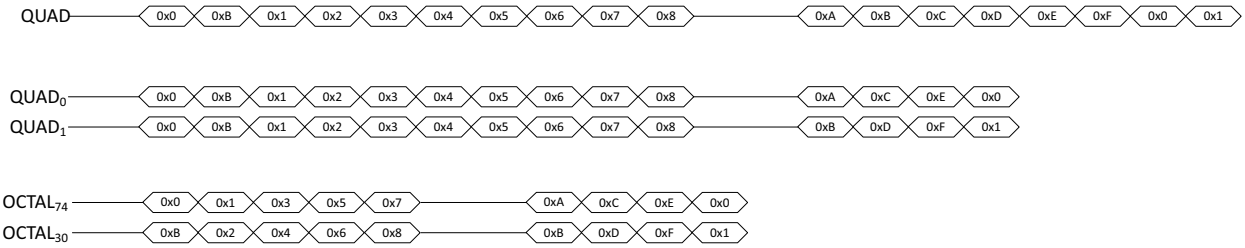
For write (TX) operations, data should be written to the TXFIFO after the transaction has been started. Software should read TXENTRIES before writing to ensure that space is available in the FIFO before writing new TX data. For read operations, software should read the RXENTRIES to determine the number of words available and then read the data from the RXFIFO register.

```
// Example TX data write loop
for (i = 0; i < count; ) {
    temp1=AM_REG(MSPI,TXENTRIES);
    for(;(temp1<16) && (i<count);temp1++,i++) {
      AM_REG(MSPI,TXFIFO) = data[i];
  }
 }
```

7.3.1 Paired-Quad Device Operation (QUADCMD)

Using a single serial, dual, quad, or octal device is fairly straightforward since all data and commands sent to the device are transmitted and received as a string of serialized bytes. On the surface, using a pair of quad devices would appear to work like a single octal device, but in actuality, it is a bit more complex since instruction and address phase bytes must be replicated as nibbles to each device while data phases of the transfer are split across devices. To simplify the use of paired-quad devices by software, the MSPI controller automatically manages the nibble replication as shown in the following sequences (instruction=0x0B (read), address=0x12345678, data=0xABCDEF01 with 2 turnaround cycles).

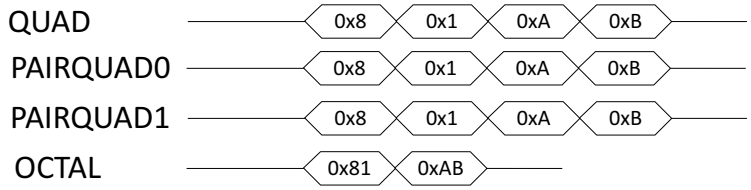
Note that the address phase of the paired-quad looks identical to the quad device while the data phase



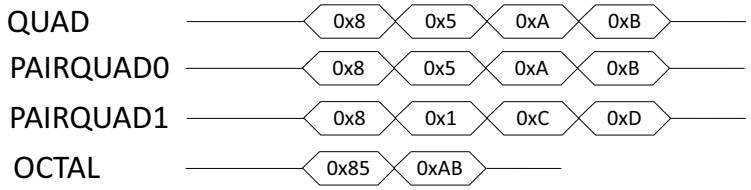
looks identical to the octal device. The MSPI module handles this automatically based on the DEVCFG field setting instead of requiring software to configure each part individually (each part could be configured by first programming the lower lane using SERIAL0 mode, the upper lane using SERIAL1 mode, then switch the MSPI interface into QUADPAIR mode).

However, register operations to a pair of quad devices must be handled as if writing both devices in parallel (data is also replicated to each device). To accomplish this, software should also set the QUADCMD field in the CTRL register when writing/reading registers (versus memory) in a paired-quad configuration, which lets the controller know that the operation is a register operation that requires the full instruction and data to be replicated to both devices.

Consider the following operations to write an 8-bit register and read the 8-bit device status for an individual quad, an octal part, and a pair of quad devices (the write sequence is 0x81 0xAB which is a write volatile configuration register on a Micron part):



Notice that each of the devices receives the same sequence as an individual quad part. Likewise, when reading status from a pair of quad devices, the status from each device must be read instead of a single 8-bit status:



Again, software must set the QUADCMD bit in the CTRL register to ensure that read data from each device is captured independently. In addition, for read operations the controller will de-interleave the read bytes and return them in the RX FIFO as 0xCDAB (device 0 in byte 0, device 1 in byte 1) and to maintain similarities when running with quad/octal devices, only a single byte write/read operation should be issued.

7.4 DMA Operations

The MSPI controller tightly integrates the DMA controller with the transfer interface and automatically handles sequencing of instructions and address to serial flash device and the subsequent transfer of data to/from system memory. Before starting DMA operations, software should have already configured the CFG register (to specify device configuration) and the FLASH register (to specify the template used for DMA operations). Software should first set up the static DMA parameters which specify the DMA burst parameters:

```
AM_REG(MSPI,DMATHRESH)=8;   // Issue new DMA at FIFO half empty/full condition
AM_REG(MSPI,DMABCOUNT)=32;  // burst count=32 bytes (8 words)
```

The MSPI implements a single FIFO for both TX and RX transfers as well as a single threshold value for RX/TX operations. In most cases, the DMATHRESH should be set at 8 to indicate that a TX DMA (read from SRAM) will be triggered when the FIFO drops below eight entries and will trigger an RX DMA (write to SRAM) when the FIFO level reaches eight entries. The BCOUNT indicates the number of words that will be transferred each time that DMA is triggered. The DMA will also trigger automatically to flush or fill the FIFOs at the end of transfer if the total count is not a multiple of 32 bytes.

To initiate a DMA transfer, software should issue the following register operations:

```
AM_REG(MSPI,DMADEVADDR)=(uint32_t) addr;   // set device address
AM_REG(MSPI,DMATARGADDR)=(uint32_t) data;  // set address in system memory
AM_REG(MSPI,DMATOTCOUNT)=(count<<2);       // set total number of bytes
AM_REG(MSPI,DMACFG)=AM_REG_MSPI_DMACFG_DMAEN | // enable DMA
                    AM_REG_MSPI_DMACFG_DMADIR_P2M;  // peripheral to memory
```

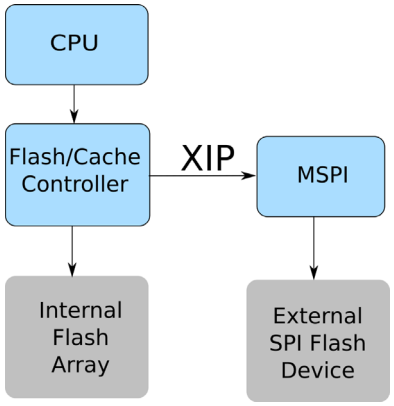
When complete, the MSPI will issue the DMACPL interrupt and software can monitor the status by reading the DMATIP bit in the DMASTAT register. Transfers to the flash device are initiated by setting the DMADIR field to M2P (Memory to Peripheral).

The controller will use the template in the FLASH register to determine whether to send the instruction and address phases (XIPSENDI, XIPSENDA) and whether to insert turnaround cycles (XIPENTURN). Instruction and address lengths are determined by the settings in the CFG register and the address and transfer count are set by the DMADEVADDR and DMATOTCOUNT registers. The instruction send for read (RX) operations is specified in the READINSTR field of the FLASH register and likewise the WRITEINSTR field is used when transmitting data to the flash device.

If the AUTO DMA cannot be used because the device's characteristics don't fit into the template, software can issue PIO operations to initiate a more complex transfer setup and then enable DMA for just the bulk DMA portions using the DMAEN_EN instead of DMAEN_AUTO.

Optionally, the MSPI can turn off it's power domain at the end of a DMA transfer if the DMAPWROFF bit is set in the DMACFG register. The domain will only power off once the entire DMA is complete (i.e. writes have been committed to system memory or have completed to the external flash device).

7.5 Execute in Place (XIP) Operations



The XIP mode of operation allows devices on the MSPI interface to be mapped into the flash cache's address space and appear as an extension to the internal flash array(s). Once enabled by the XIPEN bit in the FLASH register, the flash/cache module will decode the address region and forward operations to the MSPI interface for completion. XIP mode uses the same configuration information as DMA mode and will automatically execute a cache line read fetch from the attached device and return it to the cache controller.

XIP and DMA/PIO operations can all be interleaved since the MSPI controller will allow the current operation to complete before performing the XIP operation. Generally DMA read operations can safely be interleaved with XIP, however, XIP mode may have to be disabled during flash programming operations since the flash array within the device may not be available during program or erase operations and thus would return invalid data.

7.5.1 XIPMM Operation

For Apollo3 Rev B, the MSPI additionally supports a memory-mapped XIP mode (XIPMM) that enables full read/write mapping of an MSPI device such as a PSRAM to the CPU's peripheral address map at offset 0x51000000-0x51FFFFFF. This is mapped to offset 0 of the device on the MSPI bus and is not cached (unlike XIP space) and thus can be used as an extension to system SRAM. The MSPI device can be accessed by both XIP and XIPMM accesses (the regions overlap), but it is recommended that XIP used for static data/instructions and that a separate area of the MSPI device is used for read/write operations to avoid having stale data visible in the cache.

XIPMM seamlessly supports word, halfword, and byte read and write accesses, however, there are a few restrictions and caveats:

As mentioned above, writes to XIPMM do not flush cached data to the same address.

For scrambled regions, XIPMM can only be written safely by writing words (byte and halfword writes will corrupt the scrambled data at that location). Byte, halfword, and word reads may all be performed to scrambled regions.

Read/Write performance to the XIPMM region will be significantly slower than accesses to internal SRAM since there are multiple cycles of command, addressing, and data transfer overhead. For this reason, internal SRAM should be used for frequently accessed data and XIPMM should be used for infrequently used data.

To help minimize the access penalty on XIPMM writes, a 2-entry write FIFO buffers all writes to XIPMM. These will be immediately accepted on the CPU's bus allowing it to continue execution. However, the CPU will stall if another write or peripheral read is performed before the previous writes complete.

7.5.2 Optimized XIP Addressing

Some SPI flash devices support an optimized XIP mode that minimizes the number of instruction/address cycles that must be transmitted in order to reduce overall fetch latency. To activate this mode, software should program the flash device's registers to enter the device XIP mode, and then update the DEVCFG field to the specified number of address bytes and then disable the XIPSENDI field in the FLASH register (assuming that no instruction needs to be sent). To exit the device's XIP mode, software should reconfigure the MSPI interface in order to send the required XIP exit sequence to the device.

7.5.3 Micron XIP Support

Micron flash devices support an XIP mode that does not require the instruction byte to be transmitted, which minimizes the access time to the device. In order to transition in and out of this mode, the MSPI controller must issue an acknowledgment of XIP mode during the first turnaround cycle for each XIP access. When transitioning into and out of XIP mode, software must set the XIPACK field of the MSPI's FLASH register appropriately.

Under normal operation, the XIPACK should be set to NOACK (0x0), indicating that no acknowledgment should be sent. To transition into XIP mode, software should perform the following actions:

1. Activate XIP in the Micron device by writing the Volatile Configuration register
2. Set the XIPACK bit field in the MSPI FLASH register to ACK (0x2)
3. Perform a memory read from the Micron device (instruction must be sent). This access will allow the MSPI controller to acknowledge switching into the XIP mode
4. Set the XIPSENDI bit field in the MSPI FLASH register to 0 to indicate that the instruction byte no longer needs to be sent.

The MSPI will now transmit just the address to the Micron device and drive a 0 onto the data lines on the first turnaround cycle to remain in XIP mode. It is important that software ONLY perform read operations to the flash device until XIP mode has been exited.

To terminate XIP mode, software should perform the following sequence:

1. Set the XIPACK bit field in the MSPI FLASH register to TERMINATE (0x3)
2. Issue a memory read to the Micron device. This will allow the MSPI controller to signal termination of XIP mode by driving the data lines high during the first turnaround cycle.
3. Set the XIPACK bit field to NOACK (0x0) and the XIPSENDI bit field to 1

After this sequence has completed, software can erase, program, or send any other instructions to the Micron flash again.

7.6 Command Queueing (CQ)

The MSPI's command queuing (CQ) interface is similar to command queuing implementation in the IOM and BLE modules. To utilize the command queue, software basically constructs a series of register operations that would be issued to the MSPI device, but instead places them in an array in system SRAM (or internal flash). The start of this buffer is then written to the CQADDR register and the commands can issued by enabling the CQEN bit in the CQCFG register. The CQ logic then reads the address/data pairs via DMA operations and will continue executing them until the end of the command queue, which is denoted as a write to the STOP bit in the CQPAUSE register. As the CQ logic issues register operations, it will automatically pause fetching new operations while the transfer module is busy or can be paused to wait for external events based on the status of the CQPAUSE and CQFLAGS registers.

The primary limitation of CQ operations is that all addresses must reside within the MSPI module since the operations are executed internally by the MSPI module (i.e. it cannot write register in other modules, etc).

7.6.1 Command Queue Data Format

As the command queue resides in system memory, the general format is pairs of words that form the register address to write as well as the data to write. Assuming the CQ base address is 0x10000, system SRAM might look like the following table:

Table 323: Command Queue Example

Address	Data	Description
0x10000	0x50014258	DMATARGADDR register address
0x10004	0x00002800	Data to write to DMATARGADDR (i.e. 0x2800 is the target buffer)
0x10008	0x5001425C	DMADEVADDR register address
0x1000C	0x00304000	Address within flash device
0x10010	0x50014260	DMATOTCOUNT register address
0x10014	0x00000100	Transfer 256 bytes of data
0x10018	0x50014250	DMACFG register address
0x1001C	0x00000003	AUTO DMA enable on peripheral to memory transfer
0x10020	0x50014288	CQPAUSE register address
0x10024	0x00008000	End of Command Queue (write to STOP bit)

The AM_REG macros can be used to construct the CQ table in a manner similar to below:

```
uint32_t *cqptr = 0x10000;
*cqptr++ = AM_REG_ADDR(MSPI,DMADEVADDR);
*cqptr++ = devaddr;    // set device address (for encryption)
*cqptr++ = AM_REG_ADDR(MSPI,DMATARGADDR);
*cqptr++ = data_buffer;   // set source address in memory
*cqptr++ = AM_REG_ADDR(MSPI,DMATOTCOUNT);
*cqptr++ = 4*num_words;              // set total number of bytes
*cqptr++ = AM_REG_ADDR(MSPI,DMACFG);
*cqptr++ =   AM_REG_MSPI_DMACFG_DMAEN_AUTO |
             AM_REG_MSPI_DMACFG_DMADIR_M2P); // enable DMA write
*cqptr++ = AM_REG_ADDR(MSPI,CQPAUSE);
*cqptr++ = AM_REG_MSPI_CQFLAGS_STOP_M;
```

7.6.2 CQ Interrupts

The MSPI CQ module provides several interrupts to provide feedback to software as the MSPI works through its command queue.

- CQERR: Indicates that the command queue encountered an error when fetching the command queue instructions. This can be caused by an invalid CQ pointer that points to an invalid flash or SRAM address (SRAM powered down, etc).
- CQPAUSED: Indicates that the command queue has encountered a pause condition. This can be triggered by an index match or when the CQ is waiting on a software or hardware flag.
- CQCMP: Indicates that the command queue has completed operations. This is typically used when the command queue is executing a single-shot set of commands which end with the CQ writing the STOP bit in the CQPAUSE register.
- CQUPD: Indicates a generic CQ update interrupt which is triggered by execution of a command queue entry.

Software can generate a CQUPD interrupt at any point during command queue operation by setting bit[0] of the register address of the command to a 1 (basically OR 0x1 with the address portion of a CQ entry.

This can be useful when software would like intermediate interrupts as operations complete such as after each CQ index is updated.

7.6.3 Pausing CQ Operations

While the basic operation of the CQ functionality is pretty straightforward, constructing more complex scenarios such as queuing of multiple operations requires additional logic to accommodate handshaking with the software managing the queue and other modules within the chip. The MSPI accomplishes both of these by providing the ability to pause the CQ processing using a pause mask (CQPAUSE register) and software and hardware pause flags.

After the MSPI executes a CQ write operation, it will check all bits specified in the CQPAUSE register against their CQFLAGS status, and will pause operation if all of the associated CQFLAGS bits are set. Since all registers are available to be written by both CPU software and CQ commands, there are numerous ways these can be used, but two common scenarios are

- Software can initially set a mask in CQPAUSE and CQ operation will continue until the matching CQFLAGS condition is encountered.
- The CQ command stream can set the CQPAUSE register during execution and pause until the status in FLAGS changes to indicate that it should restart.

The CQFLAGS register contains 8 soft flags (register bits that can be controlled by either the CPU or the QC operation) and an additional 8 hard flags, which are hardware status flags tied to logic in the MSPI module or other modules in the chip. The lowest two soft flags are also exported to the IOM SPI modules to facilitate communication between an IOM and the MSPI to enable management of common MSPI/IOM buffers via the command queues. The table below lists the flags available in the MSPI:

Table 324: CQFLAGS

Bit	Type	Mnemonic	Description/Use
15	Hard	STOP	CQ Stop Flag. When set to 1, CQ processing will terminate and the CQCPL interrupt will be generated.
14	Hard	CQIDX	CQ Index Pointer Match. Will be set to 1 when the CURIDX and ENDIDX pointers match. Generally used by software when forming a request queue.
13	Hard		Reserved
12	Hard		Reserved
11	Hard		Reserved
10	Hard		Reserved
9	Hard	IOM1READY	IOM Buffer 1 Ready Status. This hardware bit represents the XOR of the soft IOM1START with the incoming IOM1 ready status bit and indicates that buffer 1 has been emptied by the IOM.
8	Hard	IOM0READY	IOM Buffer 0 Ready Status. This hardware bit represents the XOR of the soft IOM0START with the incoming IOM0 ready status bit and indicates that buffer 0 has been emptied by the IOM.
7	Soft	SWFLAG7	Software flag
6	Soft	SWFLAG6	Software flag

Table 324: CQFLAGS

Bit	Type	Mnemonic	Description/Use
5	Soft	SWFLAG5	Software flag
4	Soft	SWFLAG4	Software flag
3	Soft	SWFLAG3	Software flag
2	Soft	SWFLAG2	Software flag
1	Soft	IOM1START	Flag wired to IOM devices as a hard flag for intercommunication. Typically indicates that buffer 1 has been filled by MSPI and can be emptied by the IOM.
0	Soft	IOM0START	Flag wired to IOM devices as a hard flag for intercommunication. Typically indicates that buffer 0 has been filled by MSPI and can be emptied by the IOM.

The soft flags can be set/cleared/toggled via writes to the CQSETCLEAR register and their status can be read by software by reading the CQFLAGS register directly. The CQPAUSE mask bits are enumerated in the same manner.

In order to minimize the need to pause for individual operations, the CQ will automatically pause any time the MSPI's transfer block is active (for PIO, DMA, or XIP operations). Thus, whenever the CQ enables a DMA operation, there is an implicit pause until the operation completes, and then the CQ will resume fetching additional commands. To terminate the CQ processing, the CQ or software should set the top-most CQPAUSE bit (STOP), which will cause the MSPI to terminate processing of the command queue and issue a CQCPL interrupt.

7.6.4 Using the CQ Index registers

The MSPI command queuing implementation also includes a pair of registers that allow software to manage a list of outstanding operations: CQCURIDX and CQENDIDX. When initializing the command queue software can set both of these registers to the same value, which indicate an index or reference into the position of the command queue. The CQPAUSE can then be set to CQIDX and the command queue enabled. Since the CQCURIDX equals the CQENDIDX, the command queue will immediately pause and wait for them to be

For each group of commands in the command queue, software can place a write to the CQCURIDX after each DMA operation in the command queue and then directly write the CQENDIDX register with the index of the last operation in the queue. Since the CQENDIDX now mismatches the CQCURIDX, the command queue will begin processing commands and start working its way through the queue. After completing the first operation, the command queue will include a write to the CQCURIDX to indicate that the operation has completed, and the CQ logic will check to see if the CQCURIDX equals the CQENDIDX and either pause or continue processing until the two are equal again.

This mechanism allows software to asynchronously post additional operations to the command queue by simply writing the new commands to memory and then updating the CQENDIDX to the index of the last operation. Because the MSPI CQ hardware simply looks for a match between the registers, software may roll over from 0xFF to 0x00 or use the indices in any manner they see fit as long as the end index value is not found elsewhere in the command queue.

Software can monitor the progress of the MSPI's CQ processing by enabling the DMACPL interrupt, which will generate an interrupt after each DMA completion. The interrupt routine can read the CQCURIDX register to determine which operations have completed in order to return the proper status to the application.

7.6.5 MSPI and IOM Intercommunication

The MSPI module and IOM modules can be linked through the command queue flags to allow a simple form of handshaking to facilitate data flow between the two modules. The MSPI only has a single pair of hardware flags dedicated to IOM communication so software must write the IOMSEL field in the MSPICFG register to select which IOM is paired with the MSPI.

A typical use model for this feature is for transmitting blocks of data stored in external flash to a device (such as a display) on the IOM interface. In this scenario, software would allocate two buffers in SRAM which would be filled by the MSPI and emptied by the IOM. At the beginning of the operation, software would clear the IOM0START and IOM1START flags and initialize the MSPI command queue with two read operations to load data into buffer 0 and buffer 1. Software would also initialize the corresponding flags in the IOM and set up the IOM command queue to point to begin reading at buffer 0, but pause the IOM until it sees the buffer0 status is ready.

When the MSPI command queue is enabled, it will check the IOM0READY flag (which will be zero since the incoming bit is zero and the IOM0START flag is zero) and begin processing the operation which would DMA data from the external flash to fill buffer 0. At the end of the operation, the CQ would write the CQPAUSE register with the mask for IOM1READY. The status of IOM1READY will also be zero, so it will continue processing to fill buffer 1. At the end of this operation, the CQ will write the CQPAUSE register to IOM0READY again, but this time it will likely pause because the IOM is still reading data out of buffer 0. Once the IOM finishes its reads from buffer 0, it's CQ will set the flag for buffer 0, which will in turn cause the IOM0READY hardware flag to become zero and allow the MSPI to continue processing (which would fill buffer 0 again). In this manner, software would only need to continue adding commands to the MSPI command queue in order to continuously feed data frames to the IOM device.

7.7 Data Scrambling

In order to protect customer data stored on external flash devices, the MSPI module supports a data scrambling algorithm to obfuscate data on the MSPI bus. Scrambling can be enabled by programming the SCRSTART and SCREND registers to correspond to the address range to be encrypted and setting the SCRENABLE bit in the CFG register. Scrambling is enabled for all DMA and XIP operations that fall within the scrambling window.

Accesses to the scrambling region must always be to an aligned, four-byte boundary (i.e. device address must always end in 0x0, 0x4, 0x8, 0xC). Accesses through the XIP region are always aligned to cache lines, but software must ensure that DMA operations are properly aligned. In the case of a mis-aligned DMA access, the MSPI will issue the SCRERR interrupt (SCRambling ERRor).

7.8 Auto Power Down

The MSPI module has the ability to power itself down at the end of a DMA or CQ operation. This would usually be done while the system is going into deep sleep but desires the MSPI to transfer data to or from a flash device during the beginning of the sleep period. To enable auto-power down, software should enable the DMA with the DMAPWROFF bit set or command queuing with the CQPWROFF bit set.

7.9 Pad Configuration and Enables

The MSPI transfer block generally handles the bit/byte alignment for transfers, but the MSPI also provides a set of internal pin muxes controlled by the PADCFG register to provide system level designers more options when connecting flash devices by allowing the association of chip enable with a mixture of data pins - for instance, chip enable 1 may be used in a quad configuration with data pins 0, 1, 6, 7 of the SPI interface. The pin muxing also controls the separation of I/O operations for serial devices or transfer modes, where pin 0 is typically MOSI and pin 1 is MISO instead of being a shared tristate pin.

The MSPI supports the following external connections. The columns to the right indicate which bits are used in each configuration (S=serial, D=dual, Q=quad, QP=quad-pair, O=octal with CE#). Within the table, O=output pin, I=input pin, and X=bidirectional.

Table 325: MSPI Pin Muxing

Pin Name	Direction	GPIO	Description	S0	S1	D0	D1	Q0	Q1	QP	O0	O1
ce0	Output	1,7,10,15,19, 28	MSPI CE0	O		O		O		O	O	
ce1	Output	12,32,36,41, 43,46	MSPI CE1		O		O		O	O		O
mspi8	Output	24	MSPI CLK	O	O	O	O	O	O	O	O	O
mspi7	Input/Output	3	MSPI Data Bit7						X	X	X	X
mspi6	Input/Output	2	MSPI Data Bit6						X	X	X	X
mspi5	Input/Output	1	MSPI Data Bit5		I		X		X	X	X	X
mspi4	Input/Output	0	MSPI Data Bit4		O		X		X	X	X	X
mspi3	Input/Output	23	MSPI Data Bit3					X		X	X	X
mspi2	Input/Output	4	MSPI Data Bit2					X		X	X	X
mspi1	Input/Output	26	MSPI Data Bit1	I		X		X		X	X	X
mspi0	Input/Output	22	MSPI Data Bit0	O		X		X		X	X	X

The PADOUTEN register should be programmed to enable the proper pins for the selected mode. While the MSPI will automatically drive and sample data from the proper data lines, the MSPI also contains the ability to map bit lanes from the lower quad to the upper quad in case the system-level design cannot accommodate mapping the four contiguous pins within the lower quad. This is done via the PADCFG register, which has separate input and output muxing options.

Typically, most serial SPI devices use a separate MOSI and MISO when operating in serial mode. The SEPIO bit should be set when software needs to read data from devices in serial mode, since it redirects the MISO input from pin 1 down to input data pin 0 of the MSPI's RX logic.

7.9.1 Internal Pin Muxing Options

The MSPI also has the ability to swap some pin functionality to provide some additional flexibility in selecting pins used by the MSPI. Table 325 shows the MSPI pin mapping through the GPIO module, which has fixed pads for each MSPI pin, however, the MSPI can also perform a minimal second layer of function swapping within the MSPI pins using the PADCFG register as shown in Table 326.

Table 326: PADCFG Description

Bitfield	Description
REVCS	Swaps chip enable outputs, allowing pins mapped to CS1 to be associated with the lower quad of data pins.

Table 326: PADCFG Description

Bitfield	Description
IN3,IN2,IN1,IN0	Allows muxing of individual bit inputs from the upper quad (MSPI data bits 7:4) into the lower quad. Typically the OUT7-4 bits would be set to match.
OUT7,OUT6,OUT5,OUT4	Allows muxing of individual bit outputs from the lower quad to the upper quad. Typically the IN3-0 bits would be set to match.
OUT3	Allows MSPI pin [3] to be used as the clock output.

Since the data lines within a quad are balanced with respect to each other, it is recommended that customers do not use the internal muxing features unless pin requirements disallow the use of a contiguous quad.

7.9.2 MSPI Pin Timing Board/Package Considerations

The MSPI pins in the chip pin muxing are grouped by the two sets of quad pairs. The lower MSPI quad should be used whenever possible since these pads are closer to the MSPI logic and have less delay than the upper quad. When running in octal mode, the timings through the upper pins will dictate the speed of the interface.

The MSPI logic contains controls to adjust I/O timings to accommodate differences in board or device timings through the RXCAP, RXNEG, and TXNEG bits in the MSPICFG register. The discussion below assumes SPI mode 0 (CPHA=0, CPOL=0) and that in dual/quad/octal modes that MOSI refers to all pins in transmission mode and MISO refers to all pins when in receive mode.

If there were no delays in the chip/board/device, then ideally, data is launched on the negedge of the clock and captured on the posedge of the clock at both the master (MSPI) and target (flash) device. However the presence of delays in the system complicates timing and the timing diagram shown in Figure 13 indicates how these delays are accommodated in the MSPI interface design. The CLK (int) refers to the internal 48MHz clock used by the MSPI, and the SCLK/MOSI (int) are the internal chip timings for the outgoing clock and MOSI lines. Likewise, the @ Dev signals indicate the timing at the target device's pins. (Delays shown are just representative and may not reflect actual device timings.)

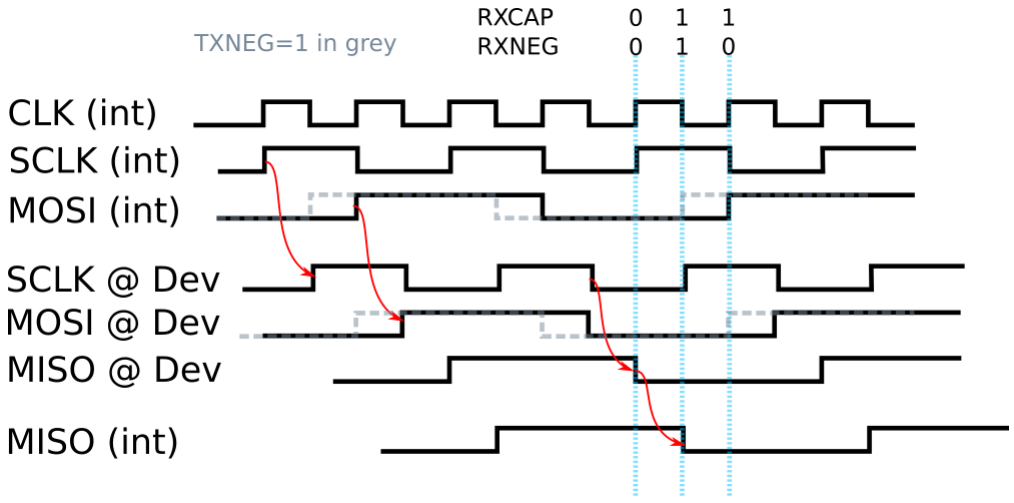


Figure 13. MSPI Interface Diagram

Note that bit transmission from the MSPI to the target is fairly straightforward since both the SCLK and MOSI are delayed by similar amounts (two red arrows on the left). Depending on which pins are used, there may be some skew between the SCLK and MOSI, however, it should be relatively small compared to the half-cycle of setup time. If additional setup is required, however, setting the TXNEG register to 1 will launch MOSI a half cycle (~10ns) early, which is indicated by the dotted gray waveform on the MOSI signals.

The target to master (MISO) timings on SPI interfaces are a bit more difficult to handle because of the cumulative round trip delay that consists of the clock delay from master to target, the access time at the target itself, and the return delay MISO path (first, third, and fourth red arrows). For this reason, read timings often dictate the frequency of a SPI bus.

The RXCAP and RXNEG bits are used together to determine the incoming RX data capture point. In an ideal world (zero delays), the MSPI would capture data at the rising edge of the internal SCLK, which would correspond to the setting of RXCAP=0, RXNEG=0 (the first vertical blue bar). It is useful, however, to push out the RX capture point to accommodate the late arrival of MISO. A setting of RXCAP=1,RXNEG=0 is the ideal setting and will delay the capture point by about 20ns (one internal 48MHz clock) as indicated in the third vertical blue line. At a 24MHz MSPI clock this should correspond to about the time the device starts driving the next data and thus should be the ideal setting. The MSPI also supports an RXCAP=1/RXNEG=1 combination, which samples in between these two points.

7.10 MSPI Registers

Multibit SPI Master

INSTANCE 0 BASE ADDRESS:0x50014000

7.10.1 Register Memory Map

Table 327: MSPI Register Map

Address(s)	Register Name	Description
0x50014000	CTRL	MSPI PIO Transfer Control/Status Register
0x50014004	CFG	MSPI Transfer Configuration Register
0x50014008	ADDR	MSPI Transfer Address Register
0x5001400C	INSTR	MSPI Transfer Instruction
0x50014010	TXFIFO	TX Data FIFO
0x50014014	RXFIFO	RX Data FIFO
0x50014018	TXENTRIES	TX FIFO Entries
0x5001401C	RXENTRIES	RX FIFO Entries
0x50014020	THRESHOLD	TX/RX FIFO Threshhold Levels
0x50014100	MSPICFG	MSPI Module Configuration
0x50014104	PADCFG	MSPI Output Pad Configuration
0x50014108	PADOUTEN	MSPI Output Enable Pad Configuration
0x5001410C	FLASH	Configuration for XIP/DMA support of SPI flash modules.
0x50014120	SCRAMBLING	External Flash Scrambling Controls
0x50014200	INTEN	MSPI Master Interrupts: Enable
0x50014204	INTSTAT	MSPI Master Interrupts: Status
0x50014208	INTCLR	MSPI Master Interrupts: Clear
0x5001420C	INTSET	MSPI Master Interrupts: Set
0x50014250	DMACFG	DMA Configuration Register
0x50014254	DMASTAT	DMA Status Register
0x50014258	DMATARGADDR	DMA Target Address Register
0x5001425C	DMADEVADDR	DMA Device Address Register
0x50014260	DMATOTCOUNT	DMA Total Transfer Count
0x50014264	DMABCOUNT	DMA BYTE Transfer Count
0x50014278	DMATHRESH	DMA Transmit Trigger Threshhold
0x500142A0	CQCFG	Command Queue Configuration Register
0x500142A8	CQADDR	CQ Target Read Address Register
0x500142AC	CQSTAT	Command Queue Status Register
0x500142B0	CQFLAGS	Command Queue Flag Register
0x500142B4	CQSETCLEAR	Command Queue Flag Set/Clear Register
0x500142B8	CQPAUSE	Command Queue Pause Mask Register
0x500142C0	CQCURIDX	Command Queue Current Index
0x500142C4	CQENDIDX	Command Queue End Index

7.10.2 MSPI Registers

7.10.2.1 CTRL Register

MSPI PIO Transfer Control/Status Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x50014000

This register is used to enable individual PIO based transactions to a device on the bus. The CFG register must be programmed properly for the transfer, and the ADDR and INSTR registers should be programmed if the SENDI and SENDA fields are enabled.

Table 328: CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
XFERBYTES																RSVD				PIOSCRAMBLE	TXRX	SENDI	SENDA	ENTURN	BIGENDIAN	RSVD		QUADCMD	BUSY	STATUS	START

Table 329: CTRL Register Bits

Bit	Name	Reset	RW	Description
31:16	XFERBYTES	0x0	RW	Number of bytes to transmit or receive (based on TXRX bit)
15:12	RSVD	0x0	RO	RESERVED
11	PIOSCRAMBLE	0x0	RW	Enables data scrambling for PIO opertions. This should only be used for data operations and never for commands to a device.
10	TXRX	0x0	RW	1 Indicates a TX operation, 0 indicates an RX operation of XFERBYTES
9	SENDI	0x0	RW	Indicates whether an instruction phase should be sent (see INSTR field and ISIZE field in CFG register)
8	SENDA	0x0	RW	Indicates whether an address phase should be sent (see ADDR register and ASIZE field in CFG register)
7	ENTURN	0x0	RW	Indicates whether TX->RX turnaround cycles should be enabled for this operation (see TURNAROUND field in CFG register).
6	BIGENDIAN	0x0	RW	1 indicates data in FIFO is in big endian format (MSB first); 0 indicates little endian data (default, LSB first).
5:4	RSVD	0x0	RO	RESERVED

Table 329: CTRL Register Bits

Bit	Name	Reset	RW	Description
3	QUADCMD	0x0	RW	Flag indicating that the operation is a command that should be replicated to both devices in paired QUAD mode. This is typically only used when reading/writing configuration registers in paired flash devices (do not set for memory transfers).
2	BUSY	0x0	RO	Command status: 1 indicates controller is busy (command in progress)
1	STATUS	0x0	RO	Command status: 1 indicates command has completed. Cleared by writing 1 to this bit or starting a new transfer.
0	START	0x0	RW	Write to 1 to initiate a PIO transaction on the bus (typically the entire register should be written at once with this bit set).

7.10.2.2 CFG Register

MSPI Transfer Configuration Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x50014004

Command formatting for PIO based transactions (initiated by writes to CTRL register)

Table 330: 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														CPOL	CPHA	RSVD		TURNAROUND						SEPIO	ISIZE	ASIZE		DEVCFG			

Table 331: CFG Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	CPOL	0x0	RW	Serial clock polarity. LOW = 0x0 - Clock inactive state is low. HIGH = 0x1 - Clock inactive state is high.
16	CPHA	0x0	RW	Serial clock phase. MIDDLE = 0x0 - Clock toggles in middle of data bit. START = 0x1 - Clock toggles at start of data bit.
15:14	RSVD	0x0	RO	RESERVED

Table 331: CFG Register Bits

Bit	Name	Reset	RW	Description
13:8	TURNAROUND	0x0	RW	Number of turnaound cycles (for TX->RX transitions). Qualified by ENTURN or XIPENTURN bit field.
7	SEPIO	0x0	RW	Separate IO configuration. This bit should be set when the target device has separate MOSI and MISO pins. Respective IN/OUT bits below should be set to map pins.
6	ISIZE	0x0	RW	Instruction Size
5:4	ASIZE	0x0	RW	Address Size. Address bytes to send from ADDR register
3:0	DEVCFG	0x1	RW	Flash configuration for XIP and AUTO DMA operations. Controls value for SER (Slave Enable) for XIP operations and address generation for DMA/XIP modes. Also used to configure SPIFRF (frame format). SERIAL0 = 0x1 - Single bit SPI flash on chip select 0 SERIAL1 = 0x2 - Single bit SPI flash on chip select 1 DUAL0 = 0x5 - Dual SPI flash on chip select 0 DUAL1 = 0x6 - Dual bit SPI flash on chip select 1 QUAD0 = 0x9 - Quad SPI flash on chip select 0 QUAD1 = 0xA - Quad SPI flash on chip select 1 OCTAL0 = 0xD - Octal SPI flash on chip select 0 OCTAL1 = 0xE - Octal SPI flash on chip select 1 QUADPAIRED = 0xF - Dual Quad SPI flash on chip selects 0/1. QUADPAIRED_SERIAL = 0x3 - Dual Quad SPI flash on chip selects 0/1, but transmit in serial mode for initialization operations

7.10.2.3 ADDR Register

MSPI Transfer Address Register

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x50014008

Optional Address field to send for PIO transfers

Table 332: ADDR Register

31	30	29	28	27	26	25	24	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 333: ADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	ADDR	0x0	RW	Optional Address field to send (after optional instruction field) - qualified by ASIZE in CMD register. NOTE: This register is aliased to DMADEVADDR.

7.10.2.4 INSTR Register

MSPI Transfer Instruction

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x5001400C

Optional Instruction field to send for PIO transfers

Table 334: INSTR Register

31	30	29	28	27	26	25	24	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																INSTR															

Table 335: INSTR Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:0	INSTR	0x0	RW	Optional Instruction field to send (1st byte) - qualified by ISEND/ISIZE

7.10.2.5 TXFIFO Register

TX Data FIFO

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x50014010

TX Data FIFO

Table 336: TXFIFO Register

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

Table 337: TXFIFO Register Bits

Bit	Name	Reset	RW	Description
31:0	TXFIFO	0x0	WO	Data to be transmitted. Data should normall be aligned to the LSB (pad the upper bits with zeros) unless BIGENDIAN is set.

7.10.2.6 RXFIFO Register

RX Data FIFO

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x50014014

RX Data FIFO

Table 338: RXFIFO Register

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

Table 339: RXFIFO Register Bits

Bit	Name	Reset	RW	Description
31:0	RXFIFO	0x0	RO	Receive data. Data is aligned to the LSB (padded zeros on upper bits) unless BIGENDIAN is set.

7.10.2.7 TXENTRIES Register

TX FIFO Entries

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x50014018

Number of words in TX FIFO

Table 340: TXENTRIES Register

31	30	29	28	27	26	25	24	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																											TXENTRIES				

Table 341: TXENTRIES Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4:0	TXENTRIES	0x0	RO	Number of 32-bit words/entries in TX FIFO

7.10.2.8 RXENTRIES Register

RX FIFO Entries

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x5001401C

Number of words in RX FIFO

Table 342: RXENTRIES Register

31	30	29	28	27	26	25	24	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																											RXENTRIES				

Table 343: RXENTRIES Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4:0	RXENTRIES	0x0	RO	Number of 32-bit words/entries in RX FIFO

7.10.2.9 THRESHOLD Register

TX/RX FIFO Threshhold Levels

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x50014020

Threshold levels that trigger RXFull and TXEmpty interrupts

Table 344: THRESHOLD Register

31	30	29	28	27	26	25	24	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			RXTHRESH					RSVD				TXTHRESH			

Table 345: THRESHOLD Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:13	RSVD	0x0	RO	RESERVED
12:8	RXTHRESH	0x0	RW	Number of entries in TX FIFO that cause RXE interrupt

Table 345: THRESHOLD Register Bits

Bit	Name	Reset	RW	Description
7:5	RSVD	0x0	RO	RESERVED
4:0	TXTHRESH	0x0	RW	Number of entries in TX FIFO that cause TXF interrupt

7.10.2.10MSPICFG Register

MSPI Module Configuration

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x50014100

Timing configuration bits for the MSPI module. PRSTN, IPRSTN, and FIFORESET can be used to reset portions of the MSPI interface in order to clear error conditions. The remaining bits control clock frequency and TX/RX capture timings.

Table 346: MSPICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRSTN	IPRSTN	FIFORESET	RSVD															CLKDIV						RSVD	IOMSEL				TXNEG	RXNEG	RXCAP	APBCLK

Table 347: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
31	PRSTN	0x1	RW	Peripheral reset. Master reset to the entire MSPI module (DMA, XIP, and transfer state machines). 1=normal operation, 0=in reset.
30	IPRSTN	0x1	RW	IP block reset. Write to 0 to put the transfer module in reset or 1 for normal operation. This may be required after error conditions to clear the transfer on the bus.
29	FIFORESET	0x0	RW	Reset MSPI FIFO (active high). 1=reset FIFO, 0=normal operation. May be used to manually flush the FIFO in error handling.
28:14	RSVD	0x0	RO	RESERVED

Table 347: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
13:8	CLKDIV	0x2	RW	Clock Divider. Allows dividing 48 MHz base clock by integer multiples. Enumerations are provided for common frequency, but any integer divide from 48 MHz is allowed. Odd divide ratios will result in a 33/66 percent duty cycle with a long low clock pulse (to allow longer round-trip for read data). CLK24 = 0x2 - 24 MHz MSPI clock CLK12 = 0x4 - 12 MHz MSPI clock CLK6 = 0x8 - 6 MHz MSPI clock CLK3 = 0x10 - 3 MHz MSPI clock CLK1_5 = 0x20 - 1.5 MHz MSPI clock
7	RSVD	0x0	RO	RESERVED
6:4	IOMSEL	0x0	RW	Selects which IOM is selected for CQ handshake status. IOM0 = 0x0 - ERROR: desc VALUE MISSING IOM1 = 0x1 - ERROR: desc VALUE MISSING IOM2 = 0x2 - ERROR: desc VALUE MISSING IOM3 = 0x3 - ERROR: desc VALUE MISSING IOM4 = 0x4 - ERROR: desc VALUE MISSING IOM5 = 0x5 - ERROR: desc VALUE MISSING DISABLED = 0x7 - No IOM selected. Signals always zero.
3	TXNEG	0x0	RW	Launches TX data a half clock cycle (~10ns) early. This should normally be programmed to zero (NORMAL). NORMAL = 0x0 - TX launched from posedge internal clock NEGEDGE = 0x1 - TX data launched from negedge of internal clock
2	RXNEG	0x0	RW	Adjusts the RX capture phase to the negedge of the 48MHz internal clock (~10ns early). For normal operation, it is expected that RXNEG will be set to 0. NORMAL = 0x0 - RX data sampled on posedge of internal clock NEGEDGE = 0x1 - RX data sampled on negedge of internal clock
1	RXCAP	0x0	RW	Controls RX data capture phase. A setting of 0 (NORMAL) captures read data at the normal capture point relative to the internal clock launch point. However, to accomodate chip/pad/board delays, a setting of RXCAP of 1 is expected to be used to align the capture point with the return data window. This bit is used in conjunction with RXNEG to provide 4 unique capture points, all about 10ns apart. NORMAL = 0x0 - RX Capture phase aligns with CPHA setting DELAY = 0x1 - RX Capture phase is delayed from CPHA setting by one clock edge
0	APBCLK	0x0	RW	Enable continuous APB clock. For power-efficient operation, APBCLK should be set to 0. DIS = 0x0 - Disable continuous clock. EN = 0x1 - Enable continuous clock.

7.10.2.11PADCFG Register

MSPI Output Pad Configuration

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x50014104

Configuration bits for the MSPI pads. Allows pads associated with the upper quad to be mapped to corresponding bits on the lower quad. Use of Quad0 pins is recommended for optimal timing.

Table 348: PADCFG Register

31	30	29	28	27	26	25	24	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										REVCS	IN3	IN2	IN1	IN0		RSVD											OUT7	OUT6	OUT5	OUT4	OUT3

Table 349: PADCFG Register Bits

Bit	Name	Reset	RW	Description
31:22	RSVD	0x0	RO	RESERVED
21	REVCS	0x0	RW	Reverse CS connections. Allows CS1 to be associated with lower data lanes and CS0 to be associated with upper data lines
20	IN3	0x0	RW	Data Input pad 3 pin muxing: 0=pad[3] 1=pad[7]
19	IN2	0x0	RW	Data Input pad 2 pin muxing: 0=pad[2] 1=pad[6]
18	IN1	0x0	RW	Data Input pad 1 pin muxing: 0=pad[1] 1=pad[5]
17:16	IN0	0x0	RW	Data Input pad 0 pin muxing: 0=pad[0] 1=pad[4] 2=pad[1] 3=pad[5]
15:5	RSVD	0x0	RO	RESERVED
4	OUT7	0x0	RW	Output pad 7 configuration. 0=data[7] 1=data[3]
3	OUT6	0x0	RW	Output pad 6 configuration. 0=data[6] 1=data[2]
2	OUT5	0x0	RW	Output pad 5 configuration. 0=data[5] 1=data[1]
1	OUT4	0x0	RW	Output pad 4 configuration. 0=data[4] 1=data[0]
0	OUT3	0x0	RW	Output pad 3 configuration. 0=data[3] 1=CLK

7.10.2.12PADOUTEN Register

MSPI Output Enable Pad Configuration

OFFSET: 0x00000108

INSTANCE 0 ADDRESS: 0x50014108

Enable bits for the MSPI output pads. Each active MSPI line should be set to 1 in the OUTEN field below.

Table 350: PADOUTEN Register

31	30	29	28	27	26	25	24	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																							OUTEN								

Table 351: PADOUTEN Register Bits

Bit	Name	Reset	RW	Description
31:9	RSVD	0x0	RO	RESERVED
8:0	OUTEN	0x0	RW	Output pad enable configuration. Indicates which pads should be driven. Bits [3:0] are Quad0 data, [7:4] are Quad1 data, and [8] is clock. QUAD0 = 0x10F - Quad0 (4 data + 1 clock) QUAD1 = 0x1F0 - Quad1 (4 data + 1 clock) OCTAL = 0x1FF - Octal (8 data + 1 clock) SERIAL0 = 0x103 - Serial (2 data + 1 clock)

7.10.2.13FLASH Register

Configuration for XIP/DMA support of SPI flash modules.

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x5001410C

When any SPI flash is configured, this register must be properly programmed before XIP or AUTO DMA operations commence.

Table 352: FLASH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
READINSTR								WRITEINSTR								RSVD						XIPMIXED		XIPSENDI	XIPSENDA	XIPENTURN	XIPBIGENDIAN	XIPACK		RSVD	XIPEN

Table 353: FLASH Register Bits

Bit	Name	Reset	RW	Description
31:24	READINSTR	0xb	RW	Read command sent to flash for DMA/XIP operations
23:16	WRITEINSTR	0x6	RW	Write command sent for DMA operations

Table 353: FLASH Register Bits

Bit	Name	Reset	RW	Description
15:11	RSVD	0x0	RO	RESERVED
10:8	XIPMIXED	0x0	RW	Reserved. Set to 0x0
7	XIPSENDI	0x0	RW	Indicates whether XIP/AUTO DMA operations should send an instruction (see READINSTR field and ISIZE field in CFG)
6	XIPSENDA	0x0	RW	Indicates whether XIP/AUTO DMA operations should send an an address phase (see DMADEVADDR register and ASIZE field in CFG)
5	XIPENTURN	0x0	RW	Indicates whether XIP/AUTO DMA operations should enable TX->RX turn-around cycles
4	XIPBIGENDIAN	0x0	RW	Indicates whether XIP/AUTO DMA data transfers are in big or little endian format
3:2	XIPACK	0x0	RW	Controls transmission of Micron XIP acknowledge cycles (Micron Flash devices only) NOACK = 0x0 - No acknowledege sent. Data IOs are tristated the first turn-around cycle ACK = 0x2 - Positive acknowledege sent. Data IOs are driven to 0 the first turnaround cycle to acknowledge XIP mode TERMINATE = 0x3 - Negative acknowledege sent. Data IOs are driven to 1 the first turnaround cycle to terminate XIP mode. XIPSENDI should be reenabled for the next transfer
1	RSVD	0x0	RO	RESERVED
0	XIPEN	0x0	RW	Enable the XIP (eXecute In Place) function which effectively enables the address decoding of the MSPI device in the flash/cache address space at address 0x04000000-0x07FFFFFF.

7.10.2.14SCRAMBLING Register

External Flash Scrambling Controls

OFFSET: 0x00000120

INSTANCE 0 ADDRESS: 0x50014120

Enables data scrambling for the specified range external flash addresses. Scrambling does not impact flash access performance.

Table 354: SCRAMBLING Register

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

Table 355: SCRAMBLING Register Bits

Bit	Name	Reset	RW	Description
31	SCRENABLE	0x0	RW	Enables Data Scrambling Region. When 1 reads and writes to the range will be scrambled. When 0, data will be read/written unmodified. Address range is specified in 64K granularity and the START/END ranges are included within the range.
30:26	RSVD	0x0	RO	RESERVED
25:16	SCREND	0x0	RW	Scrambling region end address [25:16] (64K block granularity). The END block is the LAST block included in the scrambled address range.
15:10	RSVD	0x0	RO	RESERVED
9:0	SCRSTART	0x0	RW	Scrambling region start address [25:16] (64K block granularity). The START block is the FIRST block included in the scrambled address range.

7.10.2.15INTEN Register

MSPI Master Interrupts: Enable

OFFSET: 0x00000200

INSTANCE 0 ADDRESS: 0x50014200

Set bits in this register to allow this module to generate the corresponding interrupt.

Table 356: 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																			SCRERR	CQERR	CQPAUSED	CQUPD	CQCMP	DERR	DCMP	RXF	RXO	RXU	TXO	TXE	CMDCMP

Table 357: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	SCRERR	0x0	RW	Scrambling Alignment Error. Scrambling operations must be aligned to word (4-byte) start address.
11	CQERR	0x0	RW	Command Queue Error Interrupt
10	CQPAUSED	0x0	RW	Command Queue is Paused.
9	CQUPD	0x0	RW	Command Queue Update Interrupt. Issued whenever the CQ performs an operation where address bit[0] is set. Useful for triggering CURIDX interrupts.
8	CQCMP	0x0	RW	Command Queue Complete Interrupt
7	DERR	0x0	RW	DMA Error Interrupt
6	DCMP	0x0	RW	DMA Complete Interrupt
5	RXF	0x0	RW	Receive FIFO full
4	RXO	0x0	RW	Receive FIFO overflow (cannot happen in MSPI design -- MSPI bus pins will stall)
3	RXU	0x0	RW	Receive FIFO underflow (only occurs when SW reads from an empty FIFO)
2	TXO	0x0	RW	Transmit FIFO Overflow (only occurs when SW writes to a full FIFO).
1	TXE	0x0	RW	Transmit FIFO empty.
0	CMDCMP	0x0	RW	Transfer complete. Note that DMA and CQ operations are layered, so CMDCMP, DCMP, and CQ* can all be signalled simultaneously

7.10.2.16INTSTAT Register

MSPI Master Interrupts: Status

OFFSET: 0x00000204

INSTANCE 0 ADDRESS: 0x50014204

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

Table 358: 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																			SCRERR	CQERR	CQPAUSED	CQUPD	CQCMP	DERR	DCMP	RXF	RXO	RXU	TXO	TXE	CMDCMP

Table 359: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	SCRERR	0x0	RW	Scrambling Alignment Error. Scrambling operations must be aligned to word (4-byte) start address.
11	CQERR	0x0	RW	Command Queue Error Interrupt
10	CQPAUSED	0x0	RW	Command Queue is Paused.
9	CQUPD	0x0	RW	Command Queue Update Interrupt. Issued whenever the CQ performs an operation where address bit[0] is set. Useful for triggering CURIDX interrupts.
8	CQCMP	0x0	RW	Command Queue Complete Interrupt
7	DERR	0x0	RW	DMA Error Interrupt
6	DCMP	0x0	RW	DMA Complete Interrupt
5	RXF	0x0	RW	Receive FIFO full
4	RXO	0x0	RW	Receive FIFO overflow (cannot happen in MSPI design -- MSPI bus pins will stall)
3	RXU	0x0	RW	Receive FIFO underflow (only occurs when SW reads from an empty FIFO)
2	TXO	0x0	RW	Transmit FIFO Overflow (only occurs when SW writes to a full FIFO).
1	TXE	0x0	RW	Transmit FIFO empty.
0	CMDCMP	0x0	RW	Transfer complete. Note that DMA and CQ operations are layered, so CMDCMP, DCMP, and CQ* can all be signalled simultaneously

7.10.2.17INTCLR Register

MSPI Master Interrupts: Clear

OFFSET: 0x00000208

INSTANCE 0 ADDRESS: 0x50014208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 360: 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																			SCRERR	CQERR	CQPAUSED	CQUPD	CQCMP	DERR	DCMP	RXF	RXO	RXU	TXO	TXE	CMDCMP

Table 361: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	SCRERR	0x0	RW	Scrambling Alignment Error. Scrambling operations must be aligned to word (4-byte) start address.
11	CQERR	0x0	RW	Command Queue Error Interrupt
10	CQPAUSED	0x0	RW	Command Queue is Paused.
9	CQUPD	0x0	RW	Command Queue Update Interrupt. Issued whenever the CQ performs an operation where address bit[0] is set. Useful for triggering CURIDX interrupts.
8	CQCMP	0x0	RW	Command Queue Complete Interrupt
7	DERR	0x0	RW	DMA Error Interrupt
6	DCMP	0x0	RW	DMA Complete Interrupt
5	RXF	0x0	RW	Receive FIFO full
4	RXO	0x0	RW	Receive FIFO overflow (cannot happen in MSPI design -- MSPI bus pins will stall)
3	RXU	0x0	RW	Receive FIFO underflow (only occurs when SW reads from an empty FIFO)
2	TXO	0x0	RW	Transmit FIFO Overflow (only occurs when SW writes to a full FIFO).

Table 361: INTCLR Register Bits

Bit	Name	Reset	RW	Description
1	TXE	0x0	RW	Transmit FIFO empty.
0	CMDCMP	0x0	RW	Transfer complete. Note that DMA and CQ operations are layered, so CMDCMP, DCMP, and CQ* can all be signalled simultaneously

7.10.2.18INTSET Register

MSPI Master Interrupts: Set

OFFSET: 0x0000020C

INSTANCE 0 ADDRESS: 0x5001420C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 362: 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																			SCRERR	CQERR	CQPAUSED	CQUPD	CQCMP	DERR	DCMP	RXF	RXO	RXU	TXO	TXE	CMDCMP

Table 363: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:13	RSVD	0x0	RO	RESERVED
12	SCRERR	0x0	RW	Scrambling Alignment Error. Scrambling operations must be aligned to word (4-byte) start address.
11	CQERR	0x0	RW	Command Queue Error Interrupt
10	CQPAUSED	0x0	RW	Command Queue is Paused.
9	CQUPD	0x0	RW	Command Queue Update Interrupt. Issued whenever the CQ performs an operation where address bit[0] is set. Useful for triggering CURIDX interrupts.
8	CQCMP	0x0	RW	Command Queue Complete Interrupt
7	DERR	0x0	RW	DMA Error Interrupt
6	DCMP	0x0	RW	DMA Complete Interrupt

Table 363: INTSET Register Bits

Bit	Name	Reset	RW	Description
5	RXF	0x0	RW	Receive FIFO full
4	RXO	0x0	RW	Receive FIFO overflow (cannot happen in MSPI design -- MSPI bus pins will stall)
3	RXU	0x0	RW	Receive FIFO underflow (only occurs when SW reads from an empty FIFO)
2	TXO	0x0	RW	Transmit FIFO Overflow (only occurs when SW writes to a full FIFO).
1	TXE	0x0	RW	Transmit FIFO empty.
0	CMDCMP	0x0	RW	Transfer complete. Note that DMA and CQ operations are layered, so CMDCMP, DCMP, and CQ* can all be signalled simultaneously

7.10.2.19DMACFG Register

DMA Configuration Register

OFFSET: 0x00000250

INSTANCE 0 ADDRESS: 0x50014250

DMA Configuration Register

Table 364: DMACFG Register

31	30	29	28	27	26	25	24	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													DMAPWROFF	RSVD													DMAPRI		DMADIR		DMAEN

Table 365: DMACFG Register Bits

Bit	Name	Reset	RW	Description
31:19	RSVD	0x0	RO	RESERVED.
18	DMAPWROFF	0x0	RW	Power off MSPI domain upon completion of DMA operation.
17:5	RSVD	0x0	RO	RESERVED.

Table 365: DMACFG Register Bits

Bit	Name	Reset	RW	Description
4:3	DMAPRI	0x0	RW	Sets the Priority of the DMA request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately) AUTO = 0x2 - Auto Priority (priority raised once TX FIFO empties or RX FIFO fills)
2	DMADIR	0x0	RW	Direction P2M = 0x0 - Peripheral to Memory (SRAM) transaction M2P = 0x1 - Memory to Peripheral transaction
1:0	DMAEN	0x0	RW	DMA Enable. Setting this bit to EN will start the DMA operation DIS = 0x0 - Disable DMA Function EN = 0x3 - Enable HW controlled DMA Function to manage DMA to flash devices. HW will automatically handle issuance of instruction/address bytes based on settings in the FLASH register.

7.10.2.20DMASTAT Register

DMA Status Register

OFFSET: 0x00000254

INSTANCE 0 ADDRESS: 0x50014254

DMA Status Register

Table 366: DMASTAT Register

31	30	29	28	27	26	25	24	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																												SCRERR	DMAERR	DMACPL	DMATIP

Table 367: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED.
3	SCRERR	0x0	RW	Scrambling Access Alignment Error. This active high bit signals that a scrambling operation was specified for a non-word aligned DEVADDR.
2	DMAERR	0x0	RW	DMA Error. This active high bit signals that an error was encountered during the DMA operation.
1	DMACPL	0x0	RW	DMA Transfer Complete. This signals the end of the DMA operation.

Table 367: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
0	DMATIP	0x0	RO	DMA Transfer In Progress indicator. 1 will indicate that a DMA transfer is active. The DMA transfer may be waiting on data, transferring data, or waiting for priority. All of these will be indicated with a 1. A 0 will indicate that the DMA is fully complete and no further transactions will be done.

7.10.2.21DMATARGADDR Register

DMA Target Address Register

OFFSET: 0x00000258

INSTANCE 0 ADDRESS: 0x50014258

DMA Target Address Register

Table 368: DMATARGADDR Register

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

Table 369: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	TARGADDR	0x0	RW	Target byte address for source of DMA (either read or write). In cases of non-word aligned addresses, the DMA logic will take care for ensuring only the target bytes are read/written.

7.10.2.22DMADEVADDR Register

DMA Device Address Register

OFFSET: 0x0000025C

INSTANCE 0 ADDRESS: 0x5001425C

DMA Device Address Register

Table 370: DMADEVADDR Register

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

Table 371: DMADEVADDR Register Bits

Bit	Name	Reset	RW	Description
31:0	DEVADDR	0x0	RW	SPI Device address for automated DMA transactions (both read and write).

7.10.2.23DMATOTCOUNT Register

DMA Total Transfer Count

OFFSET: 0x00000260

INSTANCE 0 ADDRESS: 0x50014260

DMA Total Transfer Count

Table 372: DMATOTCOUNT Register

31	30	29	28	27	26	25	24	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																TOTCOUNT															

Table 373: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved
15:0	TOTCOUNT	0x0	RW	Total Transfer Count in bytes.

7.10.2.24DMABCOUNT Register

DMA BYTE Transfer Count

OFFSET: 0x00000264

INSTANCE 0 ADDRESS: 0x50014264

DMA BYTE Transfer Count

Table 374: DMABCOUNT Register

31	30	29	28	27	26	25	24	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																								BCOUNT							

Table 375: DMABCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	Reserved
7:0	BCOUNT	0x0	RW	Burst transfer size in bytes. This is the number of bytes transferred when a FIFO trigger event occurs. Recommended values are 16 or 32.

7.10.2.25DMATHRESH Register

DMA Transmit Trigger Threshhold

OFFSET: 0x00000278

INSTANCE 0 ADDRESS: 0x50014278

Indicates FIFO level at which a DMA should be triggered. For most configurations, a setting of 8 is recommended for both read and write operations.

Table 376: DMATHRESH Register

31	30	29	28	27	26	25	24	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																												DMATHRESH			

Table 377: DMATHRESH Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3:0	DMATHRESH	0x8	RW	DMA transfer FIFO level trigger. For read operations, DMA is triggered when the FIFO level is greater than this value. For write operations, DMA is triggered when the FIFO level is less than this level. Each DMA operation will consist of BCOUNT bytes.

7.10.2.26CQCFG Register

Command Queue Configuration Register

OFFSET: 0x000002A0

INSTANCE 0 ADDRESS: 0x500142A0

This register controls Command Queueing (CQ) operations in a manner similar to the DMACFG register.

Table 378: CQCFG Register

31	30	29	28	27	26	25	24	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																												CQAUTOCLEARMASK	CQPWROFF	CQPRI	CQEN

Table 379: CQCFG Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED.
3	CQAUTO-CLEARMASK	0x0	RW	Eanble clear of CQMASK after each pause operation. This may be useful when using software flags to pause CQ.
2	CQPWROFF	0x0	RW	Power off MSPI domain upon completion of DMA operation.
1	CQPRI	0x0	RW	Sets the Priority of the command queue dma request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
0	CQEN	0x0	RW	Command queue enable. When set, will enable the processing of the command queue DIS = 0x0 - Disable CQ Function EN = 0x1 - Enable CQ Function

7.10.2.27CQADDR Register

CQ Target Read Address Register

OFFSET: 0x000002A8

INSTANCE 0 ADDRESS: 0x500142A8

Location of the command queue in SRAM or flash memory. This register will increment as CQ operations commence. Software should only write CQADDR when CQEN is disabled, however the command queue script itself may update CQADDR in order to perform queue management functions (like resetting the pointers)

Table 380: CQADDR Register

31	30	29	28	27	26	25	24	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			CQADDR																												

Table 381: CQADDR Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	Reserved
28:0	CQADDR	0x0	RW	Address of command queue buffer in SRAM or flash. The buffer address must be aligned to a word boundary.

7.10.2.28CQSTAT Register

Command Queue Status Register

OFFSET: 0x000002AC

INSTANCE 0 ADDRESS: 0x500142AC

Command Queue Status Register

Table 382: CQSTAT Register

31	30	29	28	27	26	25	24	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																												CQPAUSED	CQERR	CQCPL	CQTIP

Table 383: CQSTAT Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED.
3	CQPAUSED	0x0	RO	Command queue is currently paused status.
2	CQERR	0x0	RW	Command queue processing Error. This active high bit signals that an error was encountered during the CQ operation.
1	CQCPL	0x0	RW	Command queue operation Complete. This signals the end of the command queue operation.

Table 383: CQSTAT Register Bits

Bit	Name	Reset	RW	Description
0	CQTIP	0x0	RO	Command queue Transfer In Progress indicator. 1 will indicate that a CQ transfer is active and this will remain active even when paused waiting for external event.

7.10.2.29CQFLAGS Register

Command Queue Flag Register

OFFSET: 0x000002B0

INSTANCE 0 ADDRESS: 0x500142B0

Command Queue Flag Register

Table 384: CQFLAGS Register

31	30	29	28	27	26	25	24	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																CQFLAGS															

Table 385: CQFLAGS Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved

Table 385: CQFLAGS Register Bits

Bit	Name	Reset	RW	Description
15:0	CQFLAGS	0x0	RO	Current flag status (read-only). Bits [7:0] are software controllable and bits [15:8] are hardware status. STOP = 0x8000 - CQ Stop Flag. When set, CQ processing will complete. CQIDX = 0x4000 - CQ Index Pointers (CURIDX/ENDIDX) match. DMACPL = 0x800 - DMA Complete Status (hardwired DMACPL bit in DMASTAT) CMDCPL = 0x400 - PIO Operation completed (STATUS bit in CTRL register) IOM1READY = 0x200 - IOM Buffer 1 Ready Status (from selected IOM). This status is the result of XNOR'ing the IOM0START with the incoming status from the IOM. When high, MSPI can send to the buffer. IOM0READY = 0x100 - IOM Buffer 0 Ready Status (from selected IOM). This status is the result of XNOR'ing the IOM0START with the incoming status from the IOM. When high, MSPI can send to the buffer. SWFLAG7 = 0x80 - Software flag 7. Can be used by software to start/pause operations SWFLAG6 = 0x40 - Software flag 6. Can be used by software to start/pause operatoins SWFLAG5 = 0x20 - Software flag 5. Can be used by software to start/pause operations SWFLAG4 = 0x10 - Software flag 4. Can be used by software to start/pause operatoins SWFLAG3 = 0x8 - Software flag 3. Can be used by software to start/pause operations SWFLAG2 = 0x4 - Software flag 2. Can be used by software to start/pause operatoins SWFLAG1 = 0x2 - Software flag 1. Can be used by software to start/pause operations SWFLAG0 = 0x1 - Software flag 0. Can be used by software to start/pause operatoins IOM1START = 0x2 - IOM Buffer 1 status (same as SWFLAG1). When linked to IOM, indicates to IOM that buffer 1 is ready. IOM0START = 0x1 - IOM Buffer 0 status (same as SWFLAG0). When linked to IOM, indicates to IOM that buffer 0 is ready.

7.10.2.30CQSETCLEAR Register

Command Queue Flag Set/Clear Register

OFFSET: 0x000002B4

INSTANCE 0 ADDRESS: 0x500142B4

Command Queue Flag Set/Clear Register

Table 386: CQSETCLEAR Register

31	30	29	28	27	26	25	24	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								CQFCLR								CQFTOGGLE								CQFSET							

Table 387: CQSETCLEAR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	Reserved
23:16	CQFCLR	0x0	WO	Clear CQFlag status bits.
15:8	CQFTOGGLE	0x0	RO	Toggle CQFlag status bits
7:0	CQFSET	0x0	WO	Set CQFlag status bits. Set has priority over clear if both are high.

7.10.2.31CQPAUSE Register

Command Queue Pause Mask Register

OFFSET: 0x000002B8

INSTANCE 0 ADDRESS: 0x500142B8

Command Queue Pause Mask Register

Table 388: CQPAUSE Register

31	30	29	28	27	26	25	24	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																CQMASK															

Table 389: CQPAUSE Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved

Table 389: CQPAUSE Register Bits

Bit	Name	Reset	RW	Description
15:0	CQMASK	0x0	RW	CQ will pause processing until all specified events are satisfied. STOP = 0x8000 - CQ Stop Flag. When set, CQ processing will complete. CQIDX = 0x4000 - CQ Index Pointers (CURIDX/ENDIDX) match. DMACPL = 0x800 - DMA Complete Status (hardwired DMACPL bit in DMASTAT) CMDCPL = 0x400 - PIO Operation completed (STATUS bit in CTRL register) IOM1READY = 0x200 - IOM Buffer 1 Ready Status (from selected IOM). This status is the result of XOR'ing the IOM0START with the incoming status from the IOM. When high, MSPI can send to the buffer. IOM0READY = 0x100 - IOM Buffer 0 Ready Status (from selected IOM). This status is the result of XOR'ing the IOM0START with the incoming status from the IOM. When high, MSPI can send to the buffer. SWFLAG7 = 0x80 - Software flag 7. Can be used by software to start/pause operations SWFLAG6 = 0x40 - Software flag 6. Can be used by software to start/pause operatoins SWFLAG5 = 0x20 - Software flag 5. Can be used by software to start/pause operations SWFLAG4 = 0x10 - Software flag 4. Can be used by software to start/pause operatoins SWFLAG3 = 0x8 - Software flag 3. Can be used by software to start/pause operations SWFLAG2 = 0x4 - Software flag 2. Can be used by software to start/pause operatoins SWFLAG1 = 0x2 - Software flag 1. Can be used by software to start/pause operations SWFLAG0 = 0x1 - Software flag 0. Can be used by software to start/pause operatoins IOM1START = 0x2 - IOM Buffer 1 status (same as SWFLAG1). When linked to IOM, indicates to IOM that buffer 1 is ready. IOM0START = 0x1 - IOM Buffer 0 status (same as SWFLAG0). When linked to IOM, indicates to IOM that buffer 0 is ready.

7.10.2.32CQCURIDX Register

Command Queue Current Index

OFFSET: 0x000002C0

INSTANCE 0 ADDRESS: 0x500142C0

This register can be used in conjunction with the CQENDIDX register to manage the command queue. Typically software will initialize the CQCURIDX and CQENDIDX to the same value, which will cause the CQ to be paused when enabled. Software may then add entries to the command queue (in SRAM) and update CQENDIDX. The command queue operations will then increment CQCURIDX as it processes operations. Once CQCURIDX==CQENDIDX, the command queue hardware will automatically pause since no additional operations have been appended to the queue.

Table 390: CQCURIDX Register

31	30	29	28	27	26	25	24	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																								CQCURIDX							

Table 391: CQCURIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:0	CQCURIDX	0x0	RW	Can be used to indicate the current position of the command queue by having CQ operations write this field. A CQ hardware status flag indicates when CURIDX and ENDIDX are not equal, allowing SW to pause the CQ processing until the end index is updated.

7.10.2.33CQENDIDX Register

Command Queue End Index

OFFSET: 0x000002C4

INSTANCE 0 ADDRESS: 0x500142C4

Command Queue End Index

Table 392: CQENDIDX Register

31	30	29	28	27	26	25	24	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																								CQENDIDX							

Table 393: CQENDIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7:0	CQENDIDX	0x0	RW	Can be used to indicate the end position of the command queue. A CQ hardware status bit indices when CURIDX != ENDIDX so that the CQ can be paused when it reaches the end pointer.

8. I²C/SPI Master Module

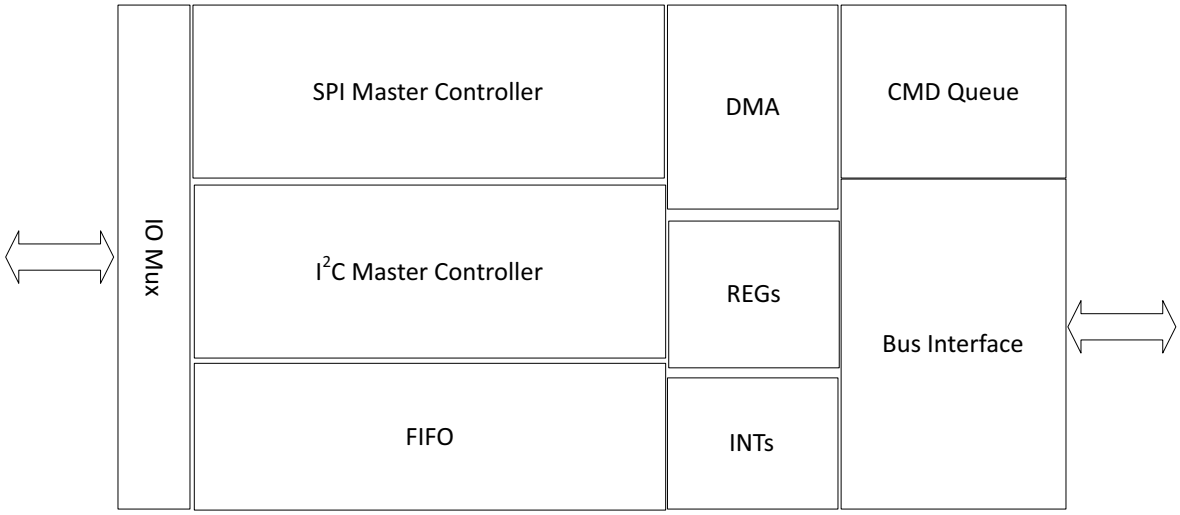


Figure 14. Block Diagram for the I²C/SPI Master Module

8.1 Functional Overview

The Apollo3 Blue MCU includes six I²C/SPI High Speed Master Modules, shown in Figure 14, each of which functions as the Master of an I²C or SPI interface as selected by the REG_IOMSTRn_IOMCFG_IFCSEL bit (n=0 or 1). A 64-byte bidirectional FIFO and a sophisticated Command mechanism allow simple initiation of I/O operations without requiring software interaction.

In I²C mode the I²C/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 I²C/SPI Master supports up to 4 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 of up to 3 bytes and raw read/write transfers. Interface timing limits are as specified in Table 1155 on page 788 and Table 1156 on page 789.

The active interface is selected by enabling the module enable bit for the interface in the REG_IOMn_SUBMODCTL. Only one interface can be active at a time. Each module contains a separate pair of 32-byte FIFOs, each of which is dedicated to data flow in a single direction (input or output). The modules support data transfer to or from the module through either direct or DMA paths. SRAM can be used as the source or the sink of data, and flash data can be used as source data for IOM transaction. Command Queue operations are also supported to allow commands to be placed in memory and fetched and executed in series. The Command Queue interface also includes inter-module flags which allows event communication between other IOM modules, MSPI modules and external pins through the GPIO interface.

Also supported in the design are test modes for use in setup and power measurements, and debug facilities to aid in software/hardware debug.

8.1.1 Main Features

No resources are shared between IOM modules, but within a single IOM module, the submodules share a common set of FIFO and command resources.

8.1.1.1 Features common to all submodules

- 2 Independent 32-byte FIFOs, one dedicated each direction of data transfer
- Direct access of all FIFO data from MCU interface, including non-destructive reads.
- FIFO mode read/write access (push/pop mechanism)
- Direct command, direct data mode. (Command and data written to/read from the module registers directly)
- Direct command, DMA data mode. Commands are written directly to the module, but data is written to/read from the main SRAM array.
- Command queuing operations. Registers write operations are read from main SRAM memory and fed to the register unit in series.
- Programmable interrupts
- Programmable threshold interrupt level
- Configurable clock selection
- Read data synchronized internally for MCU access
- Ability to send multi-byte offset addresses, with single command
- Ability to view FIFO data without causing pop operation
- Capability to store data for multiple commands in either FIFO
- Programmable number of byte offsets of 0-3

8.1.1.2 I2C Master features

- Support for standard mode (100KHz), Fast mode (400KHz), and Fast mode+ (1MHz)
- Support for 7b and 10b addressing modes
- Transfer burst sizes of 0 to 255 bytes.
- Configurable LSB or MSB data transfer.
- Clock stretching support.

8.1.1.3 SPI Master features

- Support for transaction sizes up to 4095 bytes
- Programmable number of byte offsets of 0-3
- Programmable operation in all polarity modes
- 3-wire and 4-wire read and write support
- Flow control for reads or writes, based on MISO (write flow control), or external, selectable PIO.
- Full duplex operation

8.2 Functional Description

8.2.1 Power Control

The 6 IOM modules are separated into 2 power domains, referred to as HPCB and HCPC. IOM modules 0,1 and 2 are contained in HCPB, while IOM modules 3,4 and 5 are contained in HCPC power domain. The power domain must be enabled in the REG_PWRCTRL_DEVPWREN register prior to access and operation.

8.2.2 Clocking and Resets

The IOM design uses 2 main clocks, APB_CLK and IO_CLK. The APB_CLK is used for all register and DMA accesses. It runs at 24Mhz and will be interfaced via the APB fabric synchronous interface. The

IO_CLK is used as the source of the interface clock and has selectable frequencies. The overview of the clocking structure is shown below:

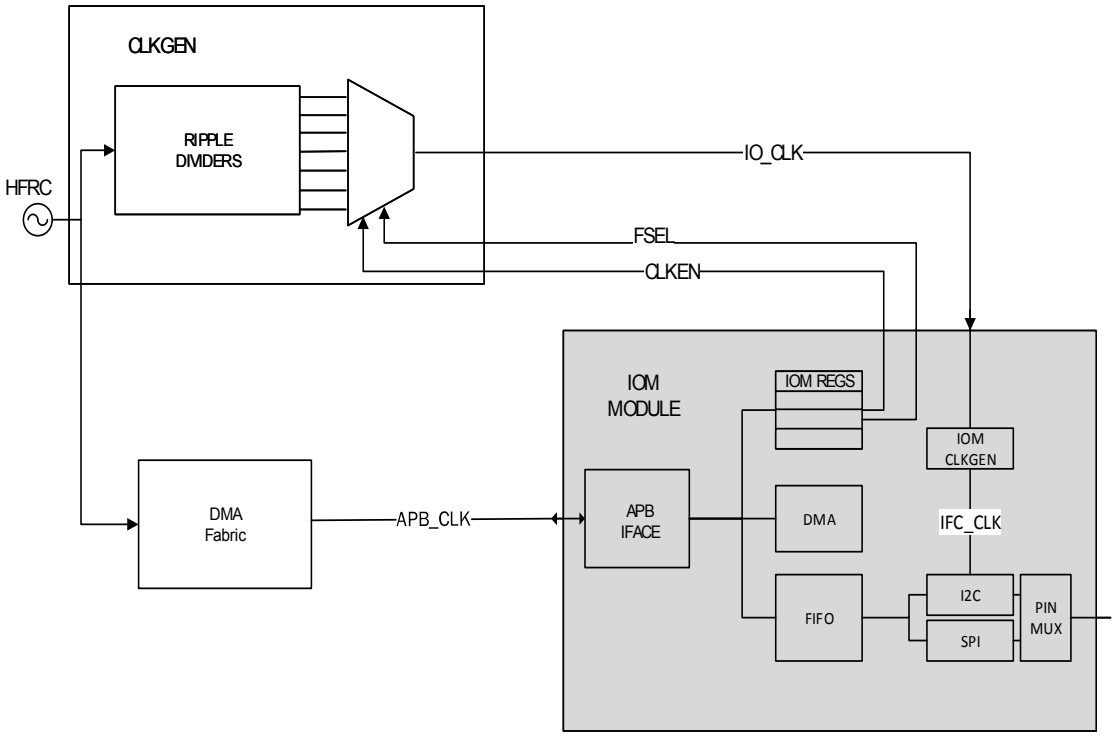


Figure 15. Clocking Structure for IOM Module

The APB_CLK is an internal clock sourced from the bus fabric and operates at a fixed 24MHz frequency. It is used for internal communication and is heavily clock gated to reduce dynamic power.

The IO_CLK is generated within the central clocking module and enabled through the REG_IOM_CLKCFG.IOCLKEN field. This clock must be enabled by software prior to module operation. The primary frequency of the IO_CLK is selected via the REG_IOM_CLKCFG.FSEL field, and further divided by either or both of the internal divide by 3 divider (enabled via the REG_IOM_CLKCFG.DIV3 field), or a programmable divider (enabled by REG_IOM_CLKCFG.DIVEN and division set by REG_IOM_CLKCFG.TOTPER and REG_IOMCLKCFG.LOWPER fields) as shown below.

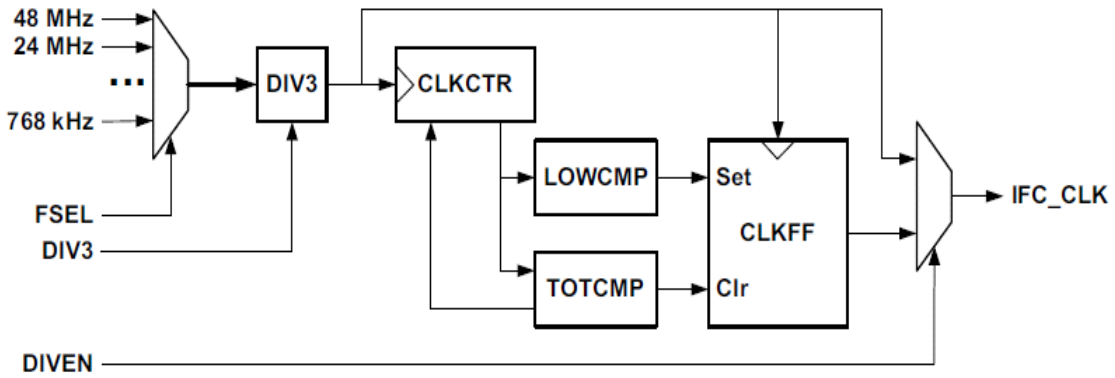


Figure 16. IO_CLK Generation

The divided by 3 divider is optional and will provide a 50% duty cycle divided by 3 clock. This divider is bypassed when the DIV3 field is set to 0.
The output of the DIV3 module is then fed to the programmable divider. This divider can be bypassed or enabled via the DIVEN field in the CLKCFG. It will divide at a rate of TOTPER+1 (subtract 1 from actual value when writing TOTPER field), and will toggle at LOWPER+1 clock count of the base IO_CLK from the DIV3 module. This will generate the final IO_CLK used by the interface module.

8.2.2.1 I2C Clock Generation

The I^2C output clock (SCL) is derived from dividing the final IO_CLK by 2. For example, for 1Mhz I^2C operation, an IO_CLK frequency of 2Mhz is required. Because the state machine will operate at 2x the target frequency of the interface frequency, the nominal output clk (SCL) duty cycle will be 50%, regardless of the duty cycle of the IO_CLK. However, the timing specification of some I^2C modes require an asymmetrical duty cycle on the SCL output, with the high period of the clock less than the low period of the clock. The clocking module allows a programmable delay prior to propagating the rising edge of the SCL output. This delay is in units of the source IO_CLK period (prior to any enabled division). This delay is specified in the REG_IOM_MI2CCFG.SCLENDLY register field. The recommended settings for this register for each mode are detailed below.

If clock stretching is done by the slave devices attached to the IOM interface, further restrictions must be observed during the setup of the clock controls. This is due to the possible clock stretch event done within a single cycle on the I^2C SCL. In this case, the minimum SCL high time must be maintained, regardless of the time the slave releases the SCL. To detect the event within the single I^2C cycle, the SCL signal needs to be sub-sampled. The source IO_CLK is used for this purpose also and allows for sampling of the SCL signal by a programmable number of source IO_CLK cycles. The sample granularity is determined by the ratio of the source IO_CLK to final IO_CLK frequency and must allow for synchronization time between the two domains. The recommended settings for each mode are below. Only speeds of 100KHz, 400KHz and 1MHz are supported. Contact Ambiq Micro for use of other frequencies. are highlighted for each mode.

Mode	FSEL	DIV3	DIV EN	TOT PER	LOW PER	SMP CNT	SDAEN DLY	SCLEN DLY
Standard Mode (100 KHz)	3	0	1	60	39	3	15	0
Fast Mode (400 KHz)	2	0	1	31	19	15	15	2
Fast Mode+ (1000 KHz)	3	0	1	6	3	1	3	0

Table 394: Recommended Mode Settings for Standard I2C Clock Speeds

8.2.2.2 SPI Clock Generation

The final IO_CLK is used directly as the SPI clock output. No additional settings are needed.

8.2.3 FIFO

The IOM module contains 2 uni-directional FIFOs, each 32 bytes wide. These FIFOs are used only for data storage during IO transactions. The FIFO supports both single (half duplex) and duplex modes of operation.

During direct mode data transfer operations, IO data transfer between the IOM module and the MCU is done by accessing the REG_IOM_FIFOPOP and REG_IOM_FIFOPUSH registers. These registers allow read (FIFOPOP) and write (FIFOPUSH) of data into and out of the FIFO, and automatic adjustment of pointers used by the submodules. Only word accesses are permitted to these registers and any unused byte locations will be ignored or filled with zero. If DMA is enabled during the IO command operation, data will automatically be read or written into the FIFO from the DMA address and the pointers updated. The FIFO pointers and data are NOT reset after each command, and care must be taken to not leave any extra data in the FIFO, as this will be used for subsequent transfers. If needed, there is a manual reset of the FIFO pointers that can be done using the REG_IOM_FIFOCTRL.FIFORSTN field. Additional information on data alignment is covered in the later sections of this document.

The submodules will prevent overruns or underruns from the FIFO by pausing the active transaction, usually by stopping the output clock. Once data is available (write operations) or there is room in the FIFO (read operations), the transaction will continue.

For debug operations, the IOM module also allows direct access to the FIFO contents through the REG_IOM_FIFO aperture. Access via this path does not affect the pointers used by the submodules and cannot be used to send or receive data as part of the IO operation. The FIFO aperture allows read and write operations into the write FIFO and read access into the read FIFO. The current FIFO pointers are readable via the FIFOLOC register. For the write FIFO, this will point to the next location to be written, while the read FIFO pointer will indicate the next location to be read.

FIFO ACCESS NOTE: When DMA operations are in progress, the FIFOPUSH and FIFOPOP registers should not be accessed, as this will interfere with the DMA data.

8.2.4 Data Alignment

All data accesses between the MCU and the IOM interface are word aligned. Since the transfer size is specified in bytes, unused bytes within the word will either be discarded (for write operations) or filled with zero (read operations) to align to the next word boundary. DMA operations support a byte starting address, and the programmed DMA address does not have to be word aligned. Direct mode write operations will start transferring the least significant byte of the word (little endian style) at the current write FIFO pointer. If any remaining bytes are unused in a word at the end of the write operation, they will be discarded, and the write pointer will be set to the next word location. Direct mode read operations will store the first received byte into the least significant byte of location specified by the read FIFO pointer, and will fill any unused byte locations with zero if the transaction size is not a word multiple. The FIFO read pointer will point to the next FIFO location in the read FIFO, which will be word aligned.

8.2.4.1 Direct Mode Data Transfers

Direct mode data is enabled when DMA is disabled via the REG_IOM_DMACFG.DMAEN and the data transfer size (TSIZE) is greater than 0. In this mode, the MCU transfers data via direct writes or reads to registers in the IOM. The IOM maintains separate FIFO pointers for the read and write FIFOs, and updates these when a PUSH or POP register is accessed. Writing to the REG_IOM_FIFOPUSH register will perform a push event of the word into the FIFO and update the write pointer by 4 bytes. Only word accesses are supported to the IOM, and any unused bytes within a word will be discarded. An example of a 5 byte write transfer is shown below.

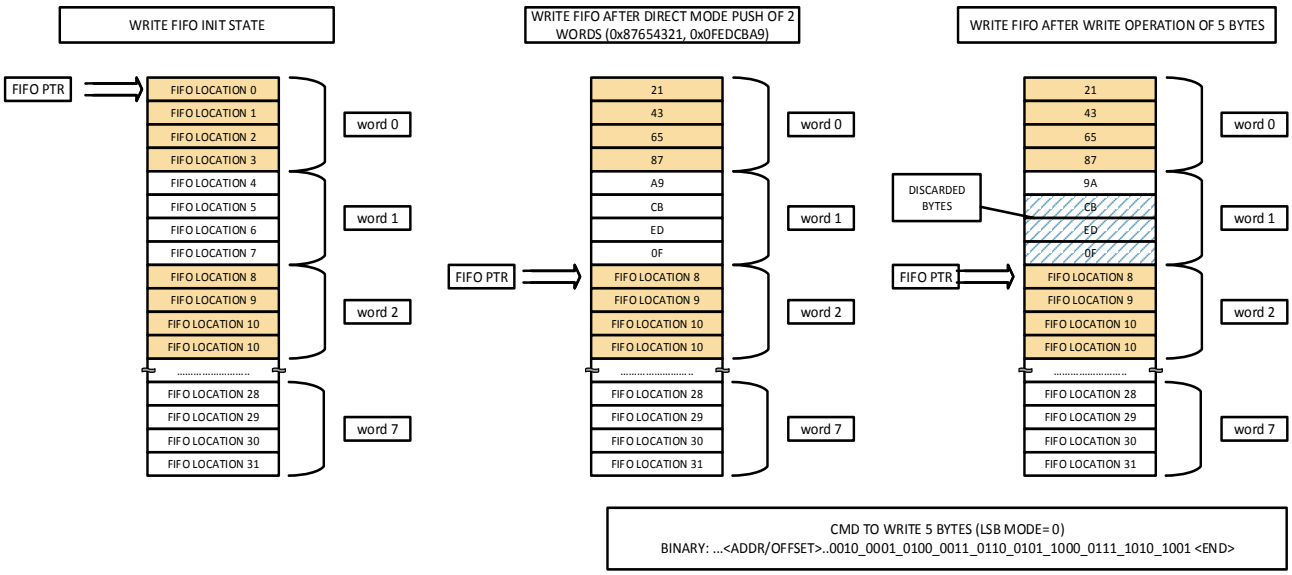


Figure 17. Direct Mode 5-byte Write Transfer

Reading from the REG_IOM_FIFOPOP register will perform a POP operation, return 4 bytes of data and advance the internal read FIFO pointer by 4 bytes. Any unused bytes within the read data will be filled with 0's and aligned to a word boundary at the end of the transaction. An example of a 5 byte read operation is shown below.

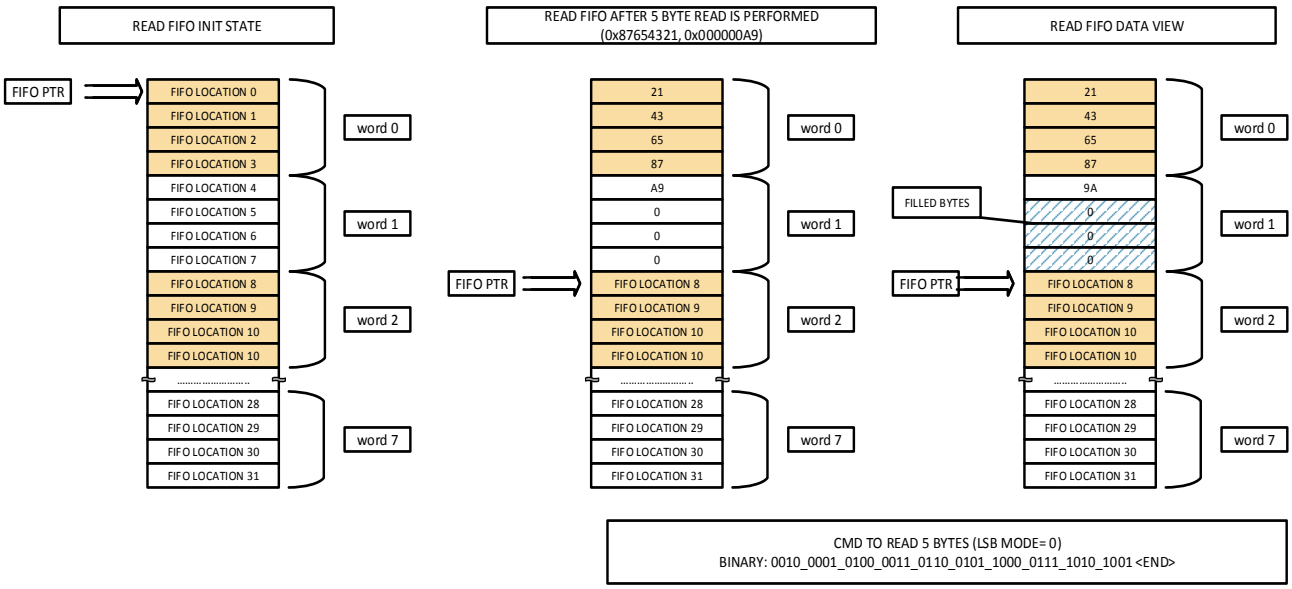


Figure 18. Direct Mode 5-byte Read

The IOM also supports a non-destructive POP mechanism to prevent unintended POP events from occurring. If the REG_IOM_FIFOCTRL.POPWR field is active (1), a write to the REG_IOM_FIFOPOP register will be required in order to complete the POP event. Reads will return the current data.

An active transaction will be paced by data availability and will hold the clock low if there is not enough data to continue write operations, or if the read FIFO is full during read operations. This wait condition is indicated when the REG_IOM_CMDSTAT.CMDSTAT field is 0x6. Once new data or FIFO locations are present, the command will continue operation automatically.

8.2.4.2 DMA Data transfers

DMA transfers are enabled by configuring the DMA related registers, enabling the DMA channel, and then issuing the command. The command will automatically fetch and store the data associated with the command without MCU intervention. The DMA channel is enabled via the REG_IOM_DMACFG.DMAEN field. P2M DMA operations transfer data from peripheral to memory and are used in IOM READ operations. M2P DMA operations transfer data from memory to peripheral and are used in IOM write operations. DMA transfer size is programmed into the REG_IOM_DMATOTCOUNT register and supports up to 4095 bytes of data transfer. The DMA transfer size is independent from the transaction size, and allows a single DMA setting to be used across multiple commands. The direction of DMA data transfer must match the command. The REG_IOM_DMACFG.DMAEN field enables/disables the DMA transfer capability and must be set last when configuring the DMA, generally prior to sending the command.

The DMA engine within the module will initiate a transfer of data when a trigger event occurs. There are 2 types of triggers available, threshold (THR) and command completion (CMDCMP). The THR trigger will activate when the threshold programmed into the FIFOWTHR or FIFORTHR in the REG_IOM_FIFOTHR register meets the data criteria. Because the MCU access to the interface is 32 bits wide, only the word count of the selected THR is used, and the low order bits of the FIFOWTHR or FIFORTHR are ignored.

During the transfer, the TOTCOUNT register is decremented to reflect the number of bytes transferred.

For IOM write operations (data written from IOM out to an external device), the THR trigger will activate when the write FIFO contains FIFOWTHR[5:2] free words. If the remaining DMA transfer size is less than this, only the needed number of words are transferred.

For IOM read operations (data read from external device), the THR trigger will activate when the read FIFO contains FIFORTHR[5:2] words of valid data. If the remaining DMA transfer size is less than the RTHR words, then the CMDCMP trigger can be enabled to transfer the remaining data. If the CMDCMP trigger is disabled, and the number of bytes in the read FIFO is greater to or equal to the current TOTCOUNT, a DMA transfer of TOTCOUNT will be done to complete the DMA operation. This mode requires that the THR trigger be enabled as well.

The CMDCMP trigger activates when the command is complete and will transfer the lesser of the TOTCOUNT or the number of bytes in the read FIFO. Note, this trigger is not needed for write operations, and the THR trigger should be used in this case.

If DMA transfer size is matched to the IOM transaction size, it is recommended to program both the FIFORTHR and FIFOWTHR to 0x10 (16 bytes) and only enable the THR trigger.

8.2.5 Transaction Initiation

To start a transaction, the IOM module must be powered up and the target external pins enabled via the GPIO module. For SPI transactions, this will generally require 4 pins to be enabled via the function select field of the PADREG registers in the GPIO module. The CEN pin for SPI transaction requires setting of the FNCSEL field of the appropriate pin, as well as the CFGREG of the corresponding pin. This also includes the setting of the default value of the CEN. This is needed to allow the IOM module to power down and not activate the CEN signal.

Once the IOM module is powered on, and the external pins configured, the IOM submodule must be enabled via the REG_IOM_SUBMODCTRL register. This will activate either the SPI or I2C interface. Once this is complete, the submodule specific registers should be configured to set the desired mode and features. If DMA is desired, the DMA registers should also be set, with the REG_IOM_DMACTRL.DMAEN field set last. The registers relating to DMA operations are as follows:

- REG_IOM_DMATRIGEN – Sets the trigger source for starting a DMA transfer
- REG_IOM_DMACFG – Sets the DMA direction and enable for DMA
- REG_IOM_DMATOTCOUNT – Sets the total count of bytes to be transferred via the DMA operation. Recommended to match the REG_IOM_CMD.TSIZE field for simplicity.

- REG_IOM_DMATARGADDR – The source or destination address of the DMA data. Sources can be either SRAM or FLASH. Destination address can only be SRAM. This is the memory mapped address of the DMA data as accessed by the MCU.

After the module setup is complete, the command register is written. This will start the IO transfer. The REG_IOM_CMD register contains the command itself, along with other fields used in the command, such as channel number, offset counts and transfer size. The IOM supports 2 main commands, read and write. A read command will write user selectable number of offset bytes (0 to 3), and then read REG_IOM_CMD.TSIZE bytes, storing the data into the read FIFO. A write command will write the user selectable number of offset bytes (0 to 3), followed by a write of REG_IOM_CMD.TSIZE bytes sourced from the write FIFO. Transfer sizes can be 0-255 for I2C and 0-4095 for SPI operations. The number of offset bytes for each command is specified in the REG_IOM_CMD.OFFSETCNT field.

8.2.6 Command Queue

The IOM module can also fetch register write data from SRAM or FLASH, and update the registers as if the write was performed via the MCU. Register data is stored as a doublet of 2 words. The first word is the module register address offset, word aligned. The second word is the write data value. Once enabled, the command queue (CQ) will fetch the address and perform a write to the register If no command is started by the register write, the next doublet will be fetched by the CQ. If a command is started (write to REG_IOM_CMD register is done), the CQ processing will wait until the transaction is complete before fetching the next register write doublet This is shown in the diagram below. No prefetching is done via the CQ, and the register write operations are performed in series with the transactions. This allows a predictable path for execution of commands. DMA enabled commands should be used during CQ operation, as there is no support to perform a direct mode read operation via the CQ.

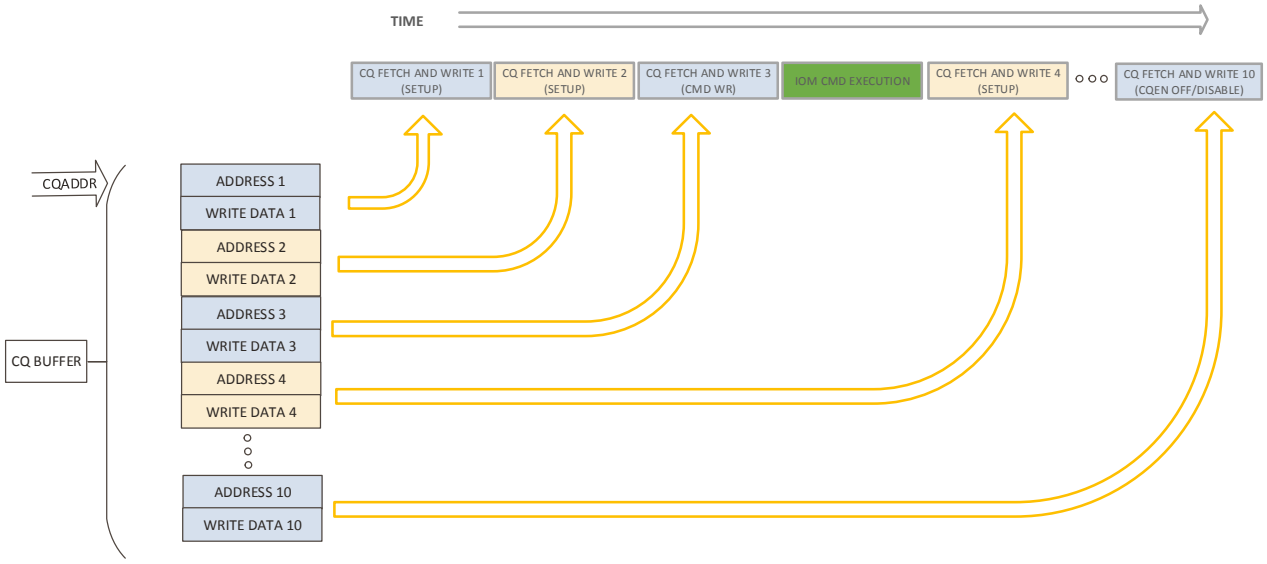


Figure 19. Register Write Data Fetches

The CQ starting fetch address is specified in the REG_IOM_CQADDR register. The CQ operation will start to fetch when the REG_IOM_CQCFG.CQEN field is set. This field should only be set when the IOM is idle and the FIFOs are empty. Once enabled, the CQ will continue to fetch sequentially until it encounters a pause event. A pause event can be caused by a CQ register write operation, or from external signals. This is shown in the sequence below.

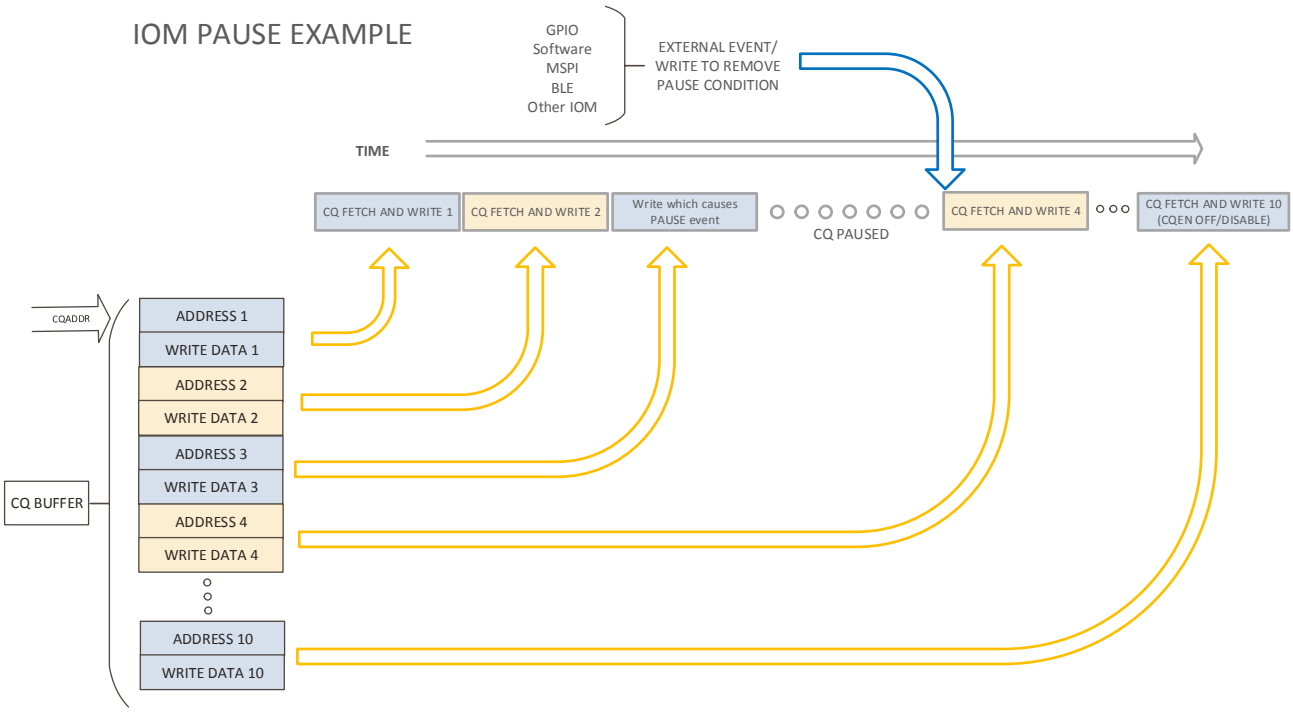


Figure 20. IOM Pause Example

Each pause source is independently enabled via the REG_IOM_PAUSEEN register. In addition to independent enable of the pause bits, there is also independent control of which pause event will signal a CQPAUSE interrupt. This is controlled through the REG_IOM_CQFLAGS.CQIRQMASK field.

There are 16 possible pause sources. When the value of the pause source is set, and the pause is enabled in the REG_IOM_PAUSEEN register, the CQ will stop fetching. The REG_IOM_CQADDR is updated after each fetch, and when paused, will point to the next doublet to be fetched when the pause condition is removed. The connection of the pause bits are shown below. The SW Flags are accessed via the IOM_CQSETCLEAR register.

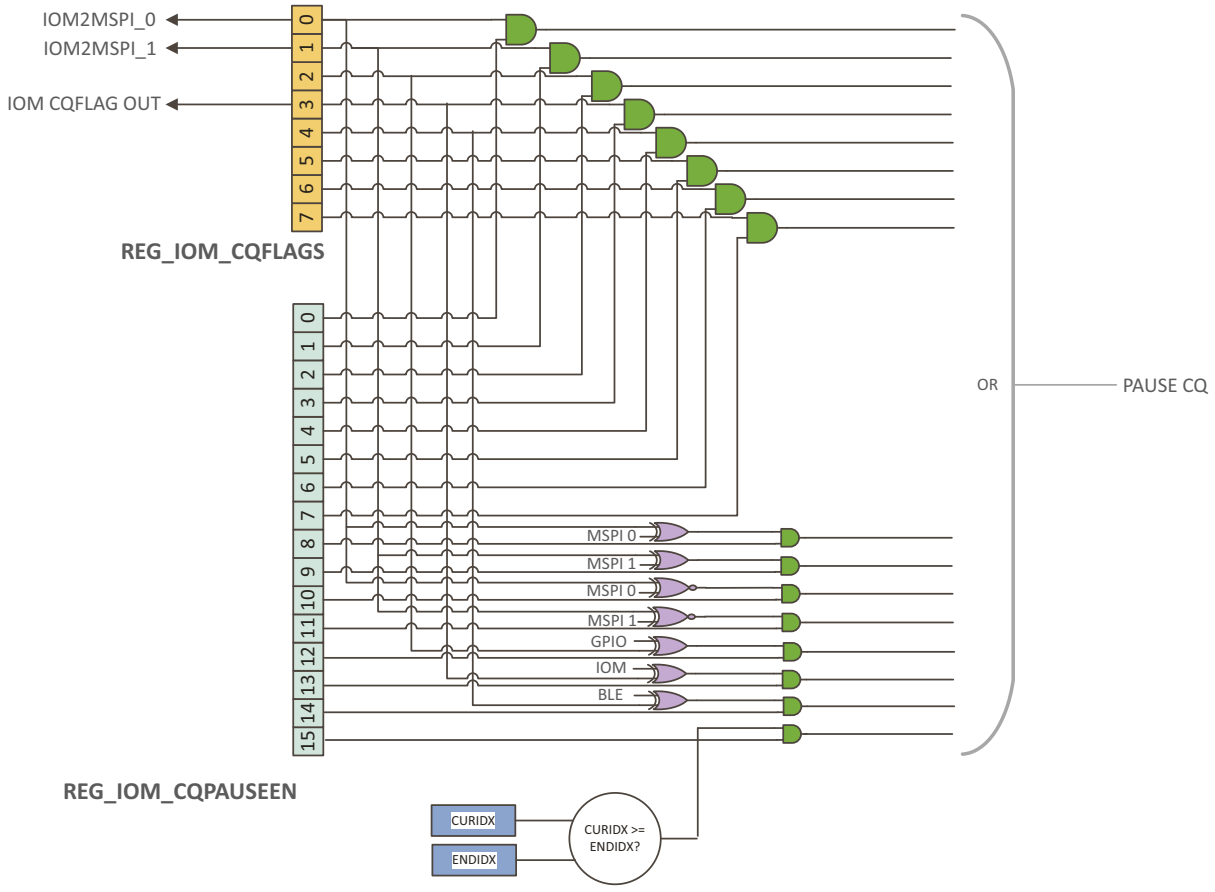


Figure 21. CQ Pause Bit Fetching

The first 8 pause sources (bits 7:0) are register bits which are directly writable via the MCU or through the CQ. These first 8 locations are called SW Flags. Because the CQ does not support a read-modify-write operation, special facilities are available to set, reset or toggle the SW Flags. This is accessed through the REG_IOM_CQSETCLEAR register. The 3 fields in this register allow a per bit set, reset or toggle of the SW Flag bits.

The next 7 pause sources (bits 14:8) use the SW Flags along with an external signal to set the pause event. The external signals are from the GPIO module, the MSPI module, or other IOM modules. On some cases, such as the MSPI interface, 4 of the SW Flags are used and combined with 2 similar signals from the MSPI module to facilitate a ping pong method of sharing 2 buffers and preventing overruns without MCU intervention. The logic and connections for each of the

The last pause source (bit 15) is use for index pausing. If this pause bit is enabled, the CQ will pause when the value of the REG_IOM_CURIDX matches the REG_IOM_ENDIDX. This is useful for software to be able to update the CQ buffer without causing a race condition between the CQ data buffer writes and the CQ fetches.

8.2.6.1 CQ programming notes

- Additional restrictions when using the CQ function is that the DMA must be disabled prior to writing the REG_IOM_CQADDR register, either from the MCU or from the CQ itself.

- For multiple commands using DMA, the DMAEN must be reset after the command is done and before the DMA registers are set for the next transaction.
- It is possible for the CQ to write the REG_IOM_CQADDR register during the CQ operation. The new address will take effect on the next fetch and allows the CQ to be relocated or looped.
- When starting the CQ operation, 1 doublet will be fetched regardless of the state of the pause status and bits. If any pause is active, it will take effect after the first fetch. For this reason, it is generally advisable to have a dummy register write as the first CQ doublet.
- CQ write operations to SW flags used in combination with pause events 15:8 must first disable the pause enable, perform the SW flag write, then re-enable the pause enable register. SW flags 7:0 can be written without this restriction and will cause a pause immediately if activated.

8.3 Programmer's Reference

An example register sequence to initiate an operation is shown below (note this does not show the data portion of the operation, only the command):

SPI SAMPLE OPERATION:

```
 // Enable clock for 24MHz SPI operation
AM_REG(IOM,CLKFG) =( 0 << AM_REG_IOM_CLKCFG_LOWPER_S) |
                ( 0 << AM_REG_IOM_CLKCFG_TOTPER_S) |
                ( 0 << AM_REG_IOM_CLKCFG_DIVEN_S ) |
                ( 1 << AM_REG_IOM_CLKCFG_DIV3_S ) |
                ( 1 << AM_REG_IOM_CLKCFG_FSEL_S ) |
                ( 1 << AM_REG_IOM_CLKCFG_IOCLKEN_S ) ;

 // Setup the SPI configuration register.MSB first, no flow control, not full duplex, mode 0
AM_REG(IOM, SPICFG) = ((0 << AM_REG_IOM_MSPICFG_MSPIRST_S) & AM_REG_IOM_MSPICFG_MSPIRST_M)|
                ((0 << AM_REG_IOM_MSPICFG_DOUTDLY_S) & AM_REG_IOM_MSPICFG_DOUTDLY_M)|
                ((0 << AM_REG_IOM_MSPICFG_DINDLY_S) & AM_REG_IOM_MSPICFG_DINDLY_M) |
                ((0 << AM_REG_IOM_MSPICFG_SPILSB_S) & AM_REG_IOM_MSPICFG_SPILSB_M) |
                ((0 << AM_REG_IOM_MSPICFG_RDFCPOL_S) & AM_REG_IOM_MSPICFG_RDFCPOL_M)|
                ((0 << AM_REG_IOM_MSPICFG_WTFCPOL_S) & AM_REG_IOM_MSPICFG_WTFCPOL_M)|
                ((0 << AM_REG_IOM_MSPICFG_WTFCIRQ_S) & AM_REG_IOM_MSPICFG_WTFCIRQ_M)|
                ((0 << AM_REG_IOM_MSPICFG_MOSIINV_S) & AM_REG_IOM_MSPICFG_MOSIINV_M)|
                ((0 << AM_REG_IOM_MSPICFG_RDFC_S) & AM_REG_IOM_MSPICFG_RDFC_M)|
                ((0 << AM_REG_IOM_MSPICFG_WTFC_S) & AM_REG_IOM_MSPICFG_WTFC_M) |
                ((0 << AM_REG_IOM_MSPICFG_FULLDUP_S) & AM_REG_IOM_MSPICFG_FULLDUP_M)|
                ((0 << AM_REG_IOM_MSPICFG_SPHA_S) & AM_REG_IOM_MSPICFG_SPHA_M)|
                ((0 << AM_REG_IOM_MSPICFG_SPOL_S) & AM_REG_IOM_MSPICFG_SPOL_M);

 // Send a read command (2) of size 0x20 using 1 byte offset of 0x32 to device on CEN
AM_REG(IOM, CMD) = (( 2 << AM_REG_IOM_CMD_CMD_S) & AM_REG_IOM_CMD_CMD_M) | // READ COMMAND
                (( 0 << AM_REG_IOM_CMD_CMDSEL_S) & AM_REG_IOM_CMD_CMDSEL_M)|
                (( 0x20 << AM_REG_IOM_CMD_TSIZE_S) & AM_REG_IOM_CMD_TSIZE_M) |
                (( 0 << AM_REG_IOM_CMD_CONT_S) & AM_REG_IOM_CMD_CONT_S) |
                (( 1 << AM_REG_IOM_CMD_OFFSETCNT_S) & AM_REG_IOM_CMD_OFFSETCNT_M) |
                (( 0x32 << AM_REG_IOM_CMD_OFFSETLO_S) & AM_REG_IOM_CMD_OFFSETLO_M);
```

8.4 Interface Clock Generation

The I^2C/SPI Master can generate a wide range of I/O interface clocks, as shown in Figure 22. The source clock is a scaled version of the HFRC 48 MHz clock, selected by REG_IOMSTRn_CLKCFG_FSEL. A divide-by-3 circuit may be selected by REG_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, REG_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 REG_IOMSTRn_CLKCFG_DIVEN, the length of the low period of the clock is specified by REG_IOMSTRn_CLKCFG_LOWPER + 1. Otherwise, the clock will have a 50% duty cycle.

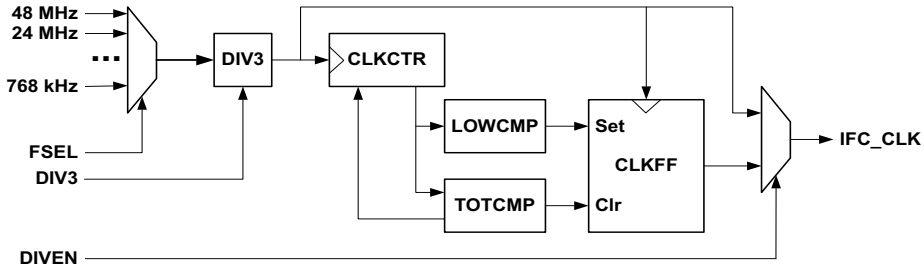


Figure 22. I²C/SPI Master Clock Generation

8.5 Command Operation

In order to minimize the amount of time the CPU must be awake during I²C/SPI Master operations, the architecture of the I²C/SPI Master is organized around processing commands which transfer data to and from an internal 64-byte FIFO.

The IOMn_CMD Register in “IOM Registers” on page 282 is used for command operations for both the SPI and I2C communication channels.

For writes to the interface, software writes data to the FIFO (REG_IOMn_FIFO_FIFO) and then sends a single command to the REG_IOMn_CMD Register. Unless the TSIZE 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 I²C slave address or the SPI channel select, the desired address offset and the length of the transfer. At that point the I²C/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, REG_IOMn_FIFOPTR_FIFOSIZ, drops below the write threshold REG_IOMn_FIFOTHR_FIFOWTHR so the CPU can wake up and refill the FIFO. The I²C/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 REG_IOMn_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 I²C/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.

8.6 FIFO

The I2C/SPI Master includes a 64-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 REG_IOMSTRn_FIFOPTR_FIFOSIZ register by 4 and decrement the REG_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.

8.7 I2C Interface

The I2C/SPI Master supports a flexible set of Commands to implement a variety of standard I2C operations. 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 Apollo3 Blue MCU I2C 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 23) and are described in the following sections

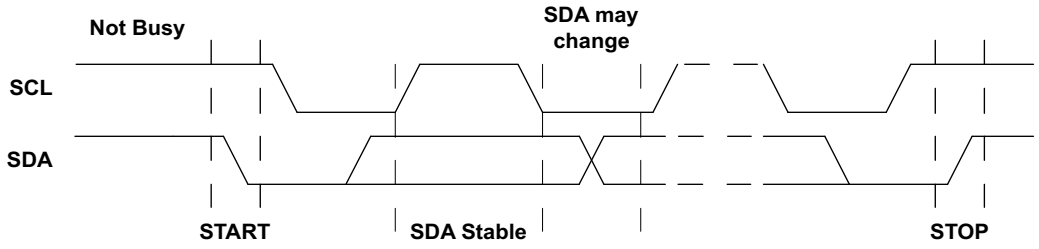


Figure 23. Basic I2C Conditions

8.7.1 Bus Not Busy

Both SDA and SCL remain high.

8.7.2 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.

8.7.3 Stop Data Transfer

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

8.7.4 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.

8.7.5 Acknowledge

Each byte of eight bits is followed by one acknowledge (ACK) bit as shown in Figure 24. 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 I2C 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.

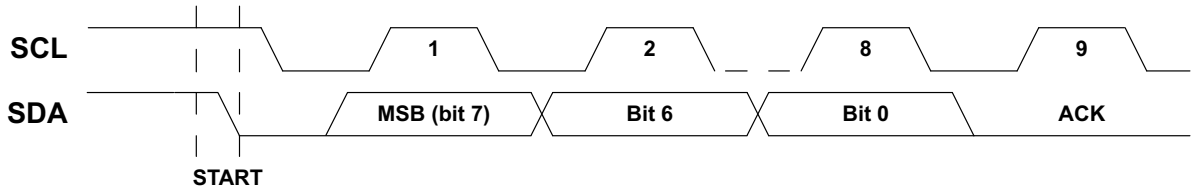


Figure 24. I²C Acknowledge

8.7.6 I²C Slave Addressing

For normal I²C 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 25 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.

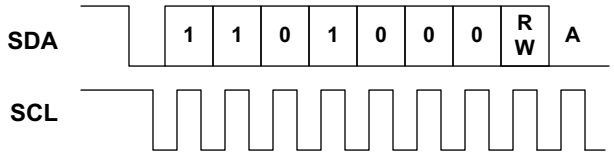


Figure 25. I2C 7-bit Address Operation

Figure 26 shows the operation with which the master addresses the Apollo3 Blue MCU 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.

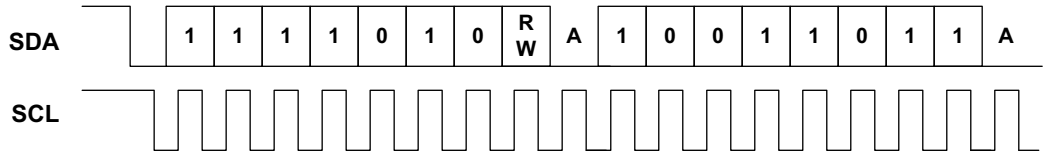


Figure 26. I2C 10-bit Address Operation

8.7.7 I2C Offset Address Transmission

If the OPER field of the CMD 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 27. The Offset Address is loaded into the Address Pointer of the slave.

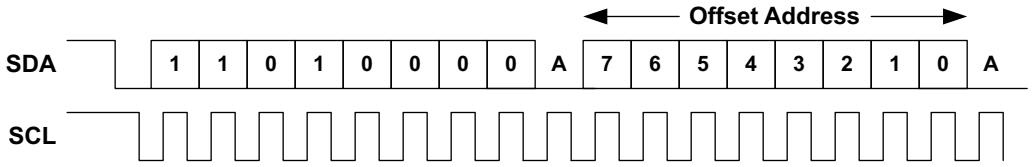


Figure 27. I2C Offset Address Transmission

8.7.8 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 27. 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 28.

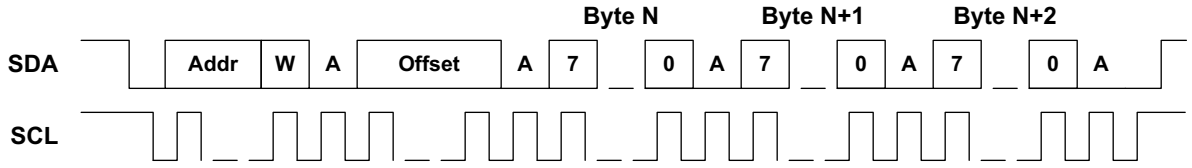


Figure 28. I²C Normal Write Operation

8.7.9 I²C Normal Read Operation

If a Normal Read operation is selected in the OPER field of the Command, 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 29, 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.

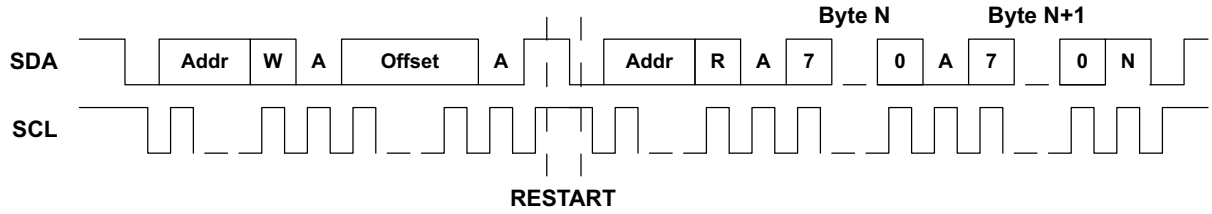


Figure 29. I²C Normal Read Operation

8.7.10 I²C Raw Write Operation

If a Raw Write is selected in the OPER field of the Command, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes as shown in Figure 30. This provides support for slave devices which do not implement the standard offset address architecture. The OFFSET field of Command is not used in this case.

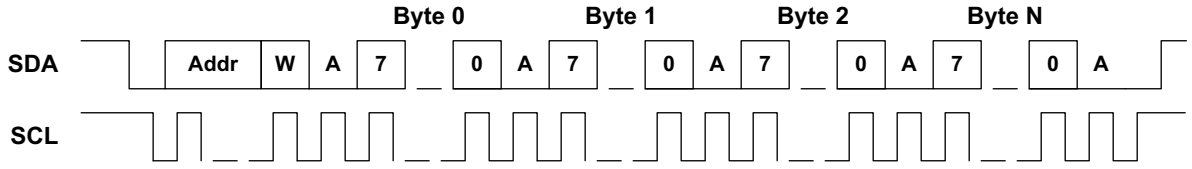


Figure 30. I²C Raw Write Operation

8.7.11 I²C Raw Read Operation

If a Raw Read is selected in the OPER field of Command, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes with a read as shown in Figure 31. 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.

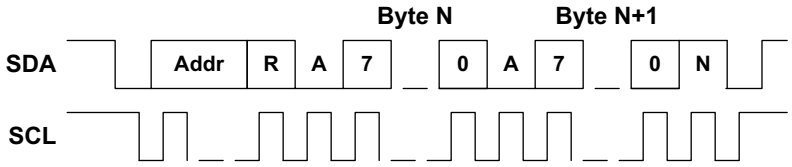


Figure 31. I2C Raw Read Operation

8.7.12 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 Apollo3 Blue MCU to hold the interface between Commands to insure that another master does not inadvertently access the same slave that the Apollo3 Blue MCU 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.

8.7.13 I2C Multi-master Arbitration

The Apollo3 Blue MCU 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU. 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 Apollo3 Blue MCU 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.

8.8 SPI Operations

8.8.1 SPI Configuration

The I2C/SPI Master supports all combinations of the polarity (CPOL) and phase (CPHA) modes of SPI using the REG_IOMSTRn_IOMCFG_SPOL and REG_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 TSIZE field in the CMD register provides a 12-bit length specification.

8.8.2 SPI Slave Addressing

In SPI mode, the Command specifies the slave channel to be used in theCMDSEL field. The I2C/SPI Master supports up to four 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.

8.8.3 SPI Normal Write

Figure 32 shows the case of a SPI Normal Write operation selected in the OPER field. The operation is initiated when the I2C/SPI Master pulls one of the four nCE signals low. At that point the I2C/SPI Master begins generating the clock on SCK and the offset address is transmitted from the master on the MOSI line, with the upper RW bit of the offset field indicating read (if 0) or write (if 1). In this example the RW bit is a one selecting a write operation. The entire one, two or three byte offset, the length of which is specified by the OFFSETCNT field, is taken from the OFFSETLO field of the CMD and, depending on the value in OFFSETCNT, the OFFSETHI field in the OFFSETHI register. The msb of the entire OFFSET should be set to 1 if the slave expects a RW bit.If the slave does not expect a RW 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 I2C/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 I2C/SPI Master.

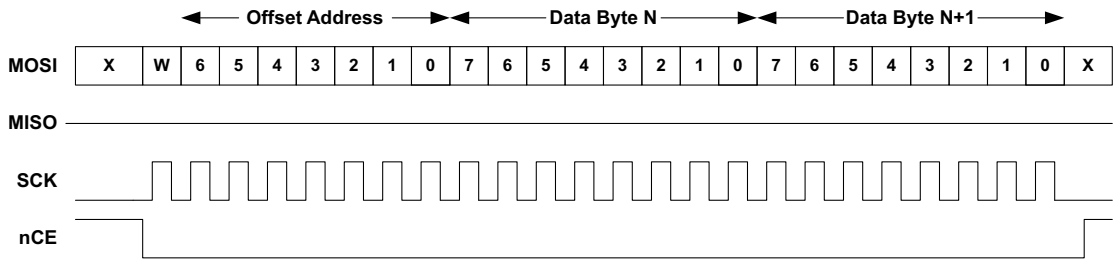


Figure 32. SPI Normal Write Operation

8.8.4 SPI Normal Read

Figure 33 shows the case of a Normal Read operation selected in the OPER field. The operation is initiated when the I2C/SPI Master pulls one of the four nCE signals low. At that point the I2C/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 RW bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the I2C/SPI Master stops driving the MOSI line and begins loading the FIFO with the data on the MISO line. The transfer continues until the I2C/SPI Master brings the nCE line high.

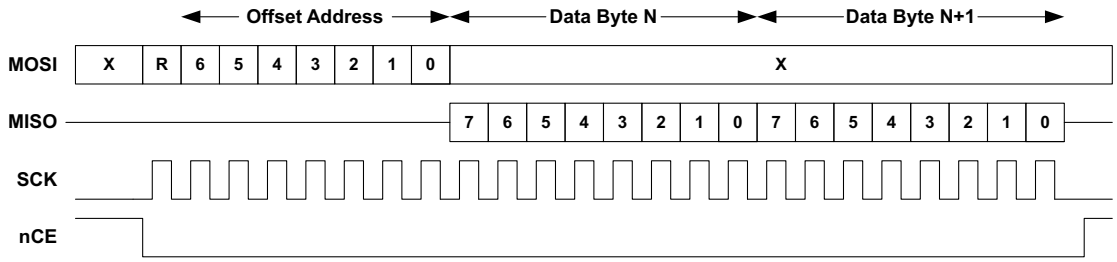


Figure 33. SPI Normal Read Operation

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

8.8.5 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 34. The OFFSET field is not used in this case.

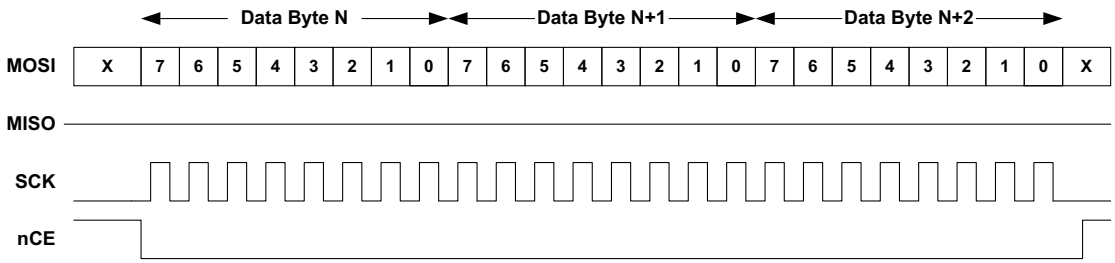


Figure 34. SPI Raw Write Operation

8.8.6 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 35. The OFFSET field is not used in this case.

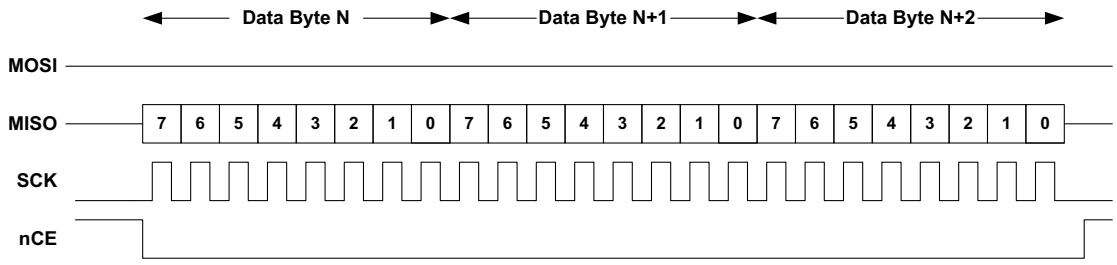


Figure 35. SPI Raw Read Operation

8.8.7 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 Apollo3 Blue MCU. 3-wire mode is configured by selecting the MxWIR3 alternative (x = 0 to 5 selecting the I2C/SPI Master) in the GPIO Pad Multiplexor rather than the MxMOSI and MxMISO alternatives. Detailed configuration information is supplied in the GPIO and Pad Configuration Module chapter.

8.8.8 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 36 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.

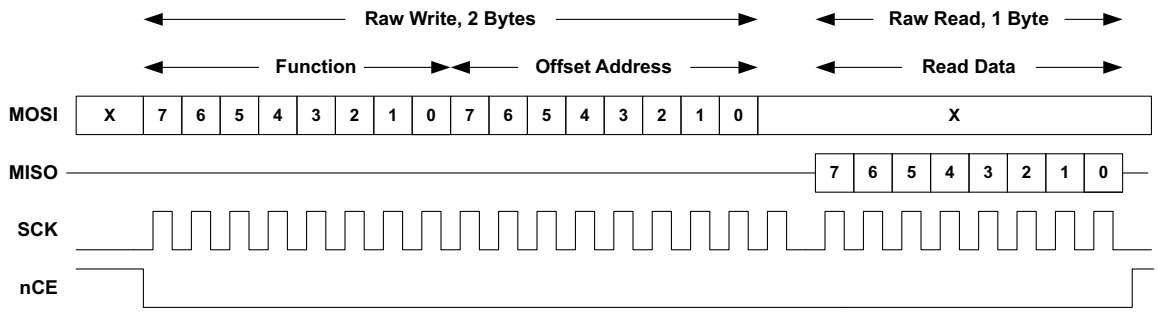


Figure 36. SPI Combined Operation

8.8.9 SPI Polarity and Phase

The Apollo3 Blue MCU supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode, as defined by the SPOL and SPHA bits. Figure 37 shows how these two bits affect the interface signal behavior.

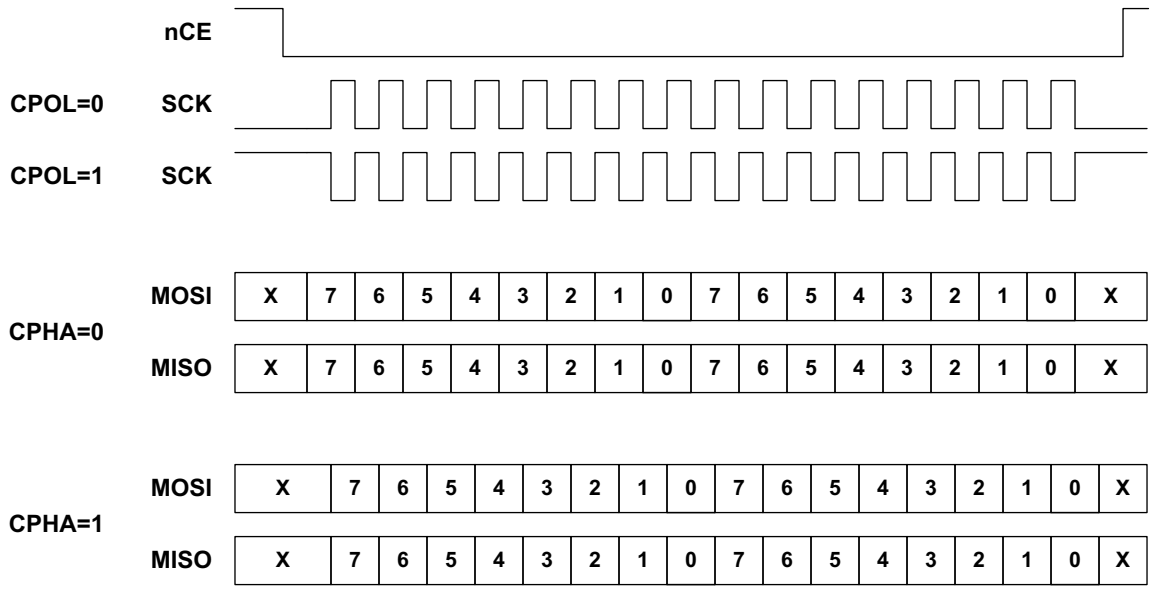


Figure 37. SPI CPOL and CPHA

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 SPOL and SPHA bits may be changed between Commands if different slave devices have different requirements. In this case the IFCEN 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.

8.9 Repeating a Command

Some peripherals, particularly sensors such as accelerometers and gyroscopes, have multiple registers which hold sample data (2 bytes each of X, Y and Z are common), and FIFOs behind these registers which hold multiple samples. In order to allow software to retrieve several samples with a single operation, the Apollo3 Blue MCU I^2C/SPI Master includes the capability to execute the same Command multiple times. If multiple Commands are desired, the REG_IOMSTRn_CMDRPT Register is loaded with the number of additional times to execute the next Command (e.g. loading CMDRPT with the value 3 will cause the next Command to be executed a total of 4 times). When a Command is written to the Command Register, the Command is then executed multiple times, filling or emptying the FIFO as appropriate. The series of repeated Commands behaves as if it was a single long Command, with a single CMDCMP interrupt occurring at the end and THR interrupts occurring if the FIFO crosses the relevant threshold. At the end of any Command the CMDRPT Register has the value 0, so that single Commands may always be executed without requiring a write to CMDRPT.

As an example, assume the peripheral has 6 bytes of sensor sample data located at register offsets 10, 11, 12, 13, 14 and 15. Also assume the internal FIFO threshold of the peripheral has been set so that an interrupt occurs when the FIFO contains 8 samples. The CMDRPT register is set to 7, and a read

Command is executed with an offset of 10 and a length of 6. This Command will be executed 8 times, each time bursting 6 bytes of data from registers 10-15 in the peripheral to the I2C/SPI Master FIFO. When CMDCMP is received the FIFO in the I2C/SPI Master will contain 48 bytes of data. The bytes of data are packed in the FIFO – there are no gaps between samples.

8.10 Bit Orientation

In both I2C and SPI modes, the I2C/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.

8.11 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 I2C/SPI Master supports this type of operation when the REG_IOMSTRn_IOMCFG_FULLDUP bit is set.

When FULLDUP is set, the I2C/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.

8.12 SPI Flow Control

The I2C/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 REG_IOMSTRn_IOMCFG_RDFC bit, in which case the I2C/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 REG_GPIO_IOMnIRQ register corresponding to the particular I2C/SPI Master. The polarity of the active state of the Flow Control IRQ is selected by the REG_IOMSTRn_IOMCFG_RDFCPOL

Write flow control is enabled by setting the REG_IOMSTRn_IOMCFG_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 REG_IOMSTRn_IOMCFG_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 REG_IOMSTRn_IOMCFG_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 REG_IOMSTRn_IOMCFG_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 38 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.

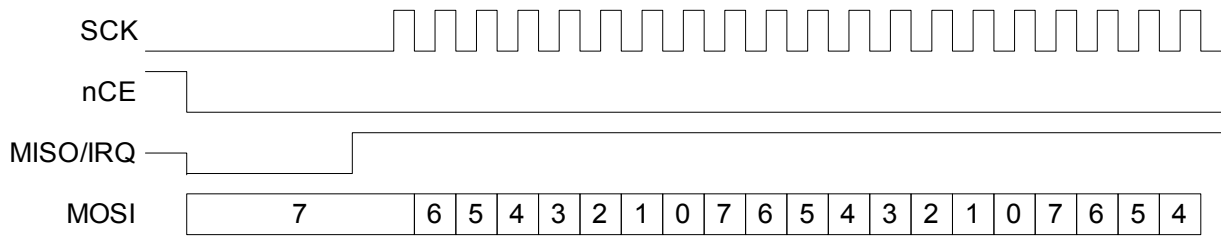


Figure 38. Flow Control at Beginning of a Write Transfer

Figure 39 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.

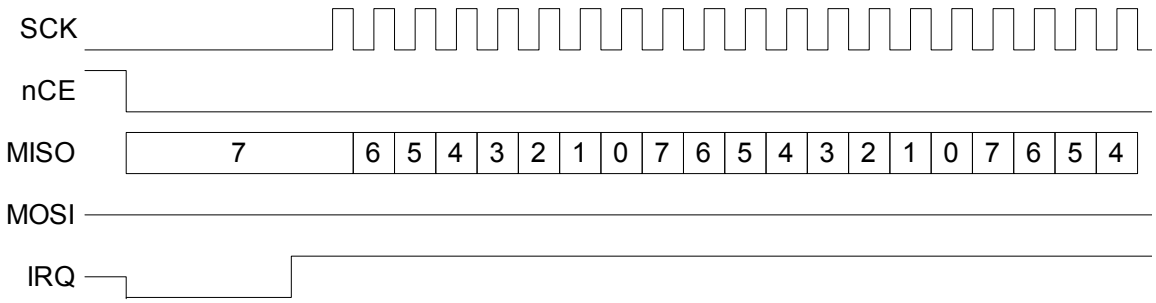


Figure 39. Flow Control at Beginning of a Raw Read Transfer

Figure 40 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 (labelled 7) and before the falling edge of the 7th bit of the byte (labelled 1) in order to insure that SCK stops at the end of the byte. Deasserting MISO or IRQ outside of that window can produce unpredictable results. SCK will resume at some point after the assertion of MISO or IRQ.

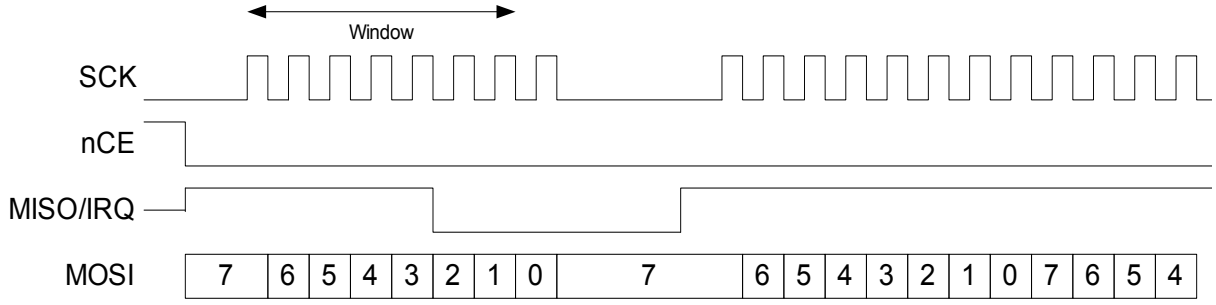


Figure 40. Flow Control in the Middle of a Write Transfer

Figure 41 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 (labelled 7) and before the falling edge of the 7th bit of the byte (labelled 1) in order to insure that SCK stops at the end of the byte. Deasserting IRQ outside of that window can produce unpredictable results. SCK will resume at some point after the assertion of IRQ.

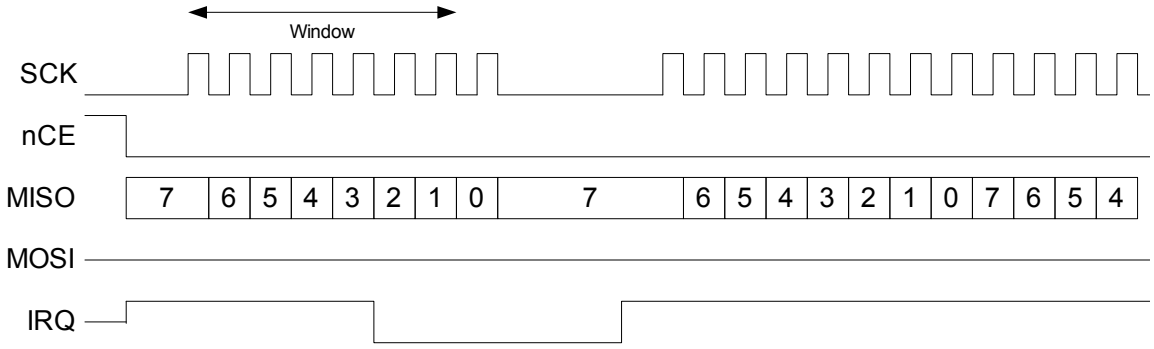


Figure 41. Flow Control in the Middle of a Read Transfer

8.13 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 I2C 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.

8.14 Minimizing Power

Each I2C/SPI Master has a global interface enable bit REG_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. When the module is not in use, the field should also be kept at 0 to minimize power. This is important even if IFCEN is a 0.

8.15 IOM Registers

IO Peripheral 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

Registers associated with the IOM master module. The IOM master module is responsible for performing SPI and I2C master operations to/from external devices. Multiple devices

8.15.1 Register Memory Map

Table 395: IOM Register Map

Address(s)	Register Name	Description
0x50004000 0x50005000 0x50006000 0x50007000 0x50008000 0x50009000	FIFO	FIFO Access Port
0x50004100 0x50005100 0x50006100 0x50007100 0x50008100 0x50009100	FIFOPTR	FIFO size and remaining slots open values
0x50004104 0x50005104 0x50006104 0x50007104 0x50008104 0x50009104	FIFOTHR	FIFO Threshold Configuration
0x50004108 0x50005108 0x50006108 0x50007108 0x50008108 0x50009108	FIFOPOP	FIFO POP register
0x5000410C 0x5000510C 0x5000610C 0x5000710C 0x5000810C 0x5000910C	FIFOPUSH	FIFO PUSH register
0x50004110 0x50005110 0x50006110 0x50007110 0x50008110 0x50009110	FIFOCTRL	FIFO Control Register
0x50004114 0x50005114 0x50006114 0x50007114 0x50008114 0x50009114	FIFOLOC	FIFO Pointers
0x50004200 0x50005200 0x50006200 0x50007200 0x50008200 0x50009200	INTEN	IO Master Interrupts: Enable

Table 395: IOM Register Map

Address(s)	Register Name	Description
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
0x50004210 0x50005210 0x50006210 0x50007210 0x50008210 0x50009210	CLKCFG	I/O Clock Configuration
0x50004214 0x50005214 0x50006214 0x50007214 0x50008214 0x50009214	SUBMODCTRL	Submodule control
0x50004218 0x50005218 0x50006218 0x50007218 0x50008218 0x50009218	CMD	Command and offset Register
0x5000421C 0x5000521C 0x5000621C 0x5000721C 0x5000821C 0x5000921C	DCX	DCX Control Register
0x50004220 0x50005220 0x50006220 0x50007220 0x50008220 0x50009220	OFFSETHI	High order 2 bytes of 3 byte offset for IO transaction
0x50004224 0x50005224 0x50006224 0x50007224 0x50008224 0x50009224	CMDSTAT	Command status

Table 395: IOM Register Map

Address(s)	Register Name	Description
0x50004240 0x50005240 0x50006240 0x50007240 0x50008240 0x50009240	DMATRIGEN	DMA Trigger Enable Register
0x50004244 0x50005244 0x50006244 0x50007244 0x50008244 0x50009244	DMATRIGSTAT	DMA Trigger Status Register
0x50004280 0x50005280 0x50006280 0x50007280 0x50008280 0x50009280	DMACFG	DMA Configuration Register
0x50004288 0x50005288 0x50006288 0x50007288 0x50008288 0x50009288	DMATOTCOUNT	DMA Total Transfer Count
0x5000428C 0x5000528C 0x5000628C 0x5000728C 0x5000828C 0x5000928C	DMATARGADDR	DMA Target Address Register
0x50004290 0x50005290 0x50006290 0x50007290 0x50008290 0x50009290	DMASTAT	DMA Status Register
0x50004294 0x50005294 0x50006294 0x50007294 0x50008294 0x50009294	CQCFG	Command Queue Configuration Register
0x50004298 0x50005298 0x50006298 0x50007298 0x50008298 0x50009298	CQADDR	CQ Target Read Address Register
0x5000429C 0x5000529C 0x5000629C 0x5000729C 0x5000829C 0x5000929C	CQSTAT	Command Queue Status Register

Table 395: IOM Register Map

Address(s)	Register Name	Description
0x500042A0 0x500052A0 0x500062A0 0x500072A0 0x500082A0 0x500092A0	CQFLAGS	Command Queue Flag Register
0x500042A4 0x500052A4 0x500062A4 0x500072A4 0x500082A4 0x500092A4	CQSETCLEAR	Command Queue Flag Set/Clear Register
0x500042A8 0x500052A8 0x500062A8 0x500072A8 0x500082A8 0x500092A8	CQPAUSEEN	Command Queue Pause Enable Register
0x500042AC 0x500052AC 0x500062AC 0x500072AC 0x500082AC 0x500092AC	CQCURIDX	IOM Command Queue current index value . Compared to the CQENDIDX reg contents to generate the IDXEQ Pause event for command queue
0x500042B0 0x500052B0 0x500062B0 0x500072B0 0x500082B0 0x500092B0	CQENDIDX	IOM Command Queue current index value . Compared to the CQCURIDX reg contents to generate the IDXEQ Pause event for command queue
0x500042B4 0x500052B4 0x500062B4 0x500072B4 0x500082B4 0x500092B4	STATUS	IOM Module Status Register
0x50004300 0x50005300 0x50006300 0x50007300 0x50008300 0x50009300	MSPICFG	SPI module master configuration
0x50004400 0x50005400 0x50006400 0x50007400 0x50008400 0x50009400	MI2CCFG	I2C Master configuration
0x50004404 0x50005404 0x50006404 0x50007404 0x50008404 0x50009404	DEVCFG	I2C Device Configuration register

Table 395: IOM Register Map

Address(s)	Register Name	Description
0x50004410 0x50005410 0x50006410 0x50007410 0x50008410 0x50009410	IOMDBG	IOM Debug Register

8.15.2 IOM Registers

8.15.2.1 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

Provides direct random access to both input and output fifos. The state of the FIFO is not distured by reading these locations (ie no POP will be done). FIFO0 is accessible from addresses 0x0 - 0x1C, and is used for data outuput from the IOM to external devices. These FIFO locations can be read and written directly.

Table 396: 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 397: FIFO Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFO	0x0	RW	FIFO direct access. Only locations 0 - 3F will return valid information.

8.15.2.2 FIFOPTR Register

FIFO size and remaining slots open values

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

Provides the current valid byte count of data within the FIFO as seen from the internal state machines. FIFO0 is dedicated to outgoing transactions and FIFO1 is dedicated to incoming transactions. All counts are specified in units of bytes.

Table 398: 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
FIFO1REM								FIFO1SIZ								FIFO0REM								FIFO0SIZ							

Table 399: FIFOPTR Register Bits

Bit	Name	Reset	RW	Description
31:24	FIFO1REM	0x0	RO	The number of remaining data bytes slots currently in FIFO 1 (written by interface, read by MCU)
23:16	FIFO1SIZ	0x0	RO	The number of valid data bytes currently in FIFO 1 (written by interface, read by MCU)
15:8	FIFO0REM	0x0	RO	The number of remaining data bytes slots currently in FIFO 0 (written by MCU, read by interface)
7:0	FIFO0SIZ	0x0	RO	The number of valid data bytes currently in the FIFO 0 (written by MCU, read by interface)

8.15.2.3 FIFOTHR Register

FIFO Threshold Configuration

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

Sets the threshold values for incoming and outgoing transactions. The threshold values are used to assert the interrupt if enabled, and also used during DMA to set the transfer size as a result of DMATHR trigger.

Table 400: 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 401: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
31:14	RSVD	0x0	RO	RESERVED
13:8	FIFOWTHR	0x0	RW	FIFO write threshold in bytes. A value of 0 will disable the write FIFO level from activating the threshold interrupt. If this field is non-zero, it will trigger a threshold interrupt when the write fifo contains FIFOWTHR free bytes, as indicated by the FIFO0REM field. This is intended to signal when a transfer of FIFOWTHR bytes can be done from the host to the IOM write fifo to support large IOM write operations.
7:6	RSVD	0x0	RO	RESERVED
5:0	FIFORTHR	0x0	RW	FIFO read threshold in bytes. A value of 0 will disable the read FIFO level from activating the threshold interrupt. If this field is non-zero, it will trigger a threshold interrupt when the read fifo contains FIFORTHR valid bytes of data, as indicated by the FIFO1SIZ field. This is intended to signal when a data transfer of FIFORTHR bytes can be done from the IOM module to the host via the read fifo to support large IOM read operations.

8.15.2.4 FIFOPOP Register

FIFO POP register

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

Will advance the internal read pointer of the incoming FIFO (FIFO1) when read, if POPWR is not active. If POPWR is active, a write to this register is needed to advance the internal FIFO pointer.

Table 402: FIFOPOP Register

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

Table 403: FIFOPOP Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFODOUT	0x0	RW	This register will return the read data indicated by the current read pointer on reads. If the POPWR control bit in the FIFOCTRL register is reset (0), the fifo read pointer will be advanced by one word as a result of the read.

8.15.2.5 FIFOPUSH Register

FIFO PUSH register

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

Will write new data into the outgoing FIFO and advance the internal write pointer.

Table 404: FIFOPUSH Register

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

Table 405: FIFOPUSH Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFODIN	0x0	RW	This register is used to write the FIFORAM in FIFO mode and will cause a push event to occur to the next open slot within the FIFORAM. Writing to this register will cause the write point to increment by 1 word(4 bytes).

8.15.2.6 FIFOCTRL Register

FIFO Control 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

Provides controls for the operation of the internal FIFOs. Contains fields used to control the operation of the POP register, and also controls to reset the internal pointers of the FIFOs.

Table 406: FIFOCTRL Register

31	30	29	28	27	26	25	24	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																														FIFORSTN	POPWR

Table 407: FIFOCTRL Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED
1	FIFORSTN	0x1	RW	Active low manual reset of the fifo. Write to 0 to reset fifo, and then write to 1 to remove the reset.
0	POPWR	0x0	RW	Selects the mode in which 'pop' events are done for the fifo read operations. A value of '1' will prevent a pop event on a read operation, and will require a write to the FIFOPOP register to create a pop event.

8.15.2.7 FIFOLOC Register

FIFO Pointers

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x50004114

INSTANCE 1 ADDRESS: 0x50005114

INSTANCE 2 ADDRESS: 0x50006114

INSTANCE 3 ADDRESS: 0x50007114

INSTANCE 4 ADDRESS: 0x50008114

INSTANCE 5 ADDRESS: 0x50009114

Provides a read only value of the current read and write pointers. This register is read only and can be used alogn with the FIFO direct access method to determine the next data to be used for input and output functions.

Table 408: FIFOLOC Register

31	30	29	28	27	26	25	24	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																				FIFORPTR				RSVD				FIFOWPTR			

Table 409: FIFOLOC Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	Reserved
11:8	FIFORPTR	0x0	RW	Current FIFO read pointer. Used to index into the incoming FIFO (FIFO1), which is used to store read data returned from external devices during a read operation.
7:4	RSVD	0x0	RO	Reserved
3:0	FIFOWPTR	0x0	RW	Current FIFO write pointer. Value is the index into the outgoing FIFO (FIFO0), which is used during write operations to external devices.

8.15.2.8 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 410: 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																	CQERR	CQUPD	CQPAUSED	DERR	DCMP	ARB	STOP	START	ICMD	IACC	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 411: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:15	RSVD	0x0	RO	RESERVED
14	CQERR	0x0	RW	Error during command queue operations
13	CQUPD	0x0	RW	CQ write operation performed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
12	CQPAUSED	0x0	RO	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
11	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
10	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
9	ARB	0x0	RW	Arbitration loss interrupt. Asserted when arbitration is enabled and has been lost to another master on the bus.
8	STOP	0x0	RW	STOP command interrupt. Asserted when another master on the bus has signaled a STOP command.
7	START	0x0	RW	START command interrupt. Asserted when another master on the bus has signaled a START command.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	NAK	0x0	RW	I2C NAK interrupt. Asserted when an unexpected NAK has been received on the I2C bus.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. This occurs when software tries to pop from an empty fifo.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.

Table 411: INTEN Register Bits

Bit	Name	Reset	RW	Description
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

8.15.2.9 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 412: 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																	CQERR	CQUPD	CQPAUSED	DERR	DCMP	ARB	STOP	START	ICMD	IACC	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 413: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:15	RSVD	0x0	RO	RESERVED
14	CQERR	0x0	RW	Error during command queue operations
13	CQUPD	0x0	RW	CQ write operation performed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
12	CQPAUSED	0x0	RO	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.

Table 413: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
11	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
10	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
9	ARB	0x0	RW	Arbitration loss interrupt. Asserted when arbitration is enabled and has been lost to another master on the bus.
8	STOP	0x0	RW	STOP command interrupt. Asserted when another master on the bus has signaled a STOP command.
7	START	0x0	RW	START command interrupt. Asserted when another master on the bus has signaled a START command.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	NAK	0x0	RW	I2C NAK interrupt. Asserted when an unexpected NAK has been received on the I2C bus.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. This occurs when software tries to pop from an empty fifo.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

8.15.2.10INTCLR 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 414: 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																	CQERR	CQUPD	CQPAUSED	DERR	DCMP	ARB	STOP	START	ICMD	IACC	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 415: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:15	RSVD	0x0	RO	RESERVED
14	CQERR	0x0	RW	Error during command queue operations
13	CQUPD	0x0	RW	CQ write operation performed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
12	CQPAUSED	0x0	RO	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
11	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
10	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
9	ARB	0x0	RW	Arbitration loss interrupt. Asserted when arbitration is enabled and has been lost to another master on the bus.
8	STOP	0x0	RW	STOP command interrupt. Asserted when another master on the bus has signaled a STOP command.
7	START	0x0	RW	START command interrupt. Asserted when another master on the bus has signaled a START command.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.

Table 415: INTCLR Register Bits

Bit	Name	Reset	RW	Description
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	NAK	0x0	RW	I2C NAK interrupt. Asserted when an unexpected NAK has been received on the I2C bus.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. This occurs when software tries to pop from an empty fifo.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

8.15.2.11INTSET 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 416: 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																	CQERR	CQUPD	CQPAUSED	DERR	DCMP	ARB	STOP	START	ICMD	IACC	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 417: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:15	RSVD	0x0	RO	RESERVED
14	CQERR	0x0	RW	Error during command queue operations
13	CQUPD	0x0	RW	CQ write operation performed a register write with the register address bit 0 set to 1. The low address bits in the CQ address fields are unused and bit 0 can be used to trigger an interrupt to indicate when this register write is performed by the CQ operation.
12	CQPAUSED	0x0	RO	Command queue is paused due to an active event enabled in the PAUSEEN register. The interrupt is posted when the event is enabled within the PAUSEEN register, the mask is active in the CQIRQMASK field and the event occurs.
11	DERR	0x0	RW	DMA Error encountered during the processing of the DMA command. The DMA error could occur when the memory access specified in the DMA operation is not available or incorrectly specified.
10	DCMP	0x0	RW	DMA Complete. Processing of the DMA operation has completed and the DMA submodule is returned into the idle state
9	ARB	0x0	RW	Arbitration loss interrupt. Asserted when arbitration is enabled and has been lost to another master on the bus.
8	STOP	0x0	RW	STOP command interrupt. Asserted when another master on the bus has signaled a STOP command.
7	START	0x0	RW	START command interrupt. Asserted when another master on the bus has signaled a START command.
6	ICMD	0x0	RW	illegal command interrupt. Asserted when a command is written when an active command is in progress.
5	IACC	0x0	RW	illegal FIFO access interrupt. Asserted when there is a overflow or underflow event
4	NAK	0x0	RW	I2C NAK interrupt. Asserted when an unexpected NAK has been received on the I2C bus.
3	FOVFL	0x0	RW	Write FIFO Overflow interrupt. This occurs when software tries to write to a full fifo. The current operation does not stop.
2	FUNDFL	0x0	RW	Read FIFO Underflow interrupt. This occurs when software tries to pop from an empty fifo.
1	THR	0x0	RW	FIFO Threshold interrupt. For write operations, asserted when the number of free bytes in the write FIFO equals or exceeds the WTHR field.

Table 417: INTSET Register Bits

Bit	Name	Reset	RW	Description
0	CMDCMP	0x0	RW	Command Complete interrupt. Asserted when the current operation has completed. For repeated commands, this will only be asserted when the final repeated command is completed.

8.15.2.12CLKCFG Register

I/O Clock Configuration

OFFSET: 0x00000210

INSTANCE 0 ADDRESS: 0x50004210

INSTANCE 1 ADDRESS: 0x50005210

INSTANCE 2 ADDRESS: 0x50006210

INSTANCE 3 ADDRESS: 0x50007210

INSTANCE 4 ADDRESS: 0x50008210

INSTANCE 5 ADDRESS: 0x50009210

Provides clock related controls used internal to the BLEIF module, and enablement of 32KHz clock to the BLE Core module. The internal clock sourced is selected via the FSEL and can be further divided by 3 using the DIV3 control.

Table 418: 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							IOCLKEN

Table 419: CLKCFG Register Bits

Bit	Name	Reset	RW	Description
31:24	TOTPER	0x0	RW	Clock total clock count minus 1. This provides the total period of the divided clock -1 when the DIVEN is active. The
23:16	LOWPER	0x0	RW	Clock low clock count minus 1. This provides the number of clocks the divided clock will be low when the DIVEN = 1.
15:13	RSVD	0x0	RO	RESERVED
12	DIVEN	0x0	RW	Enable clock division by TOTPER and LOWPER DIS = 0x0 - Disable TOTPER division. EN = 0x1 - Enable TOTPER division.

Table 419: CLKCFG Register Bits

Bit	Name	Reset	RW	Description
11	DIV3	0x0	RW	Enable divide by 3 of the source IOCLK. Division by 3 is done before the DIVEN programmable divider, and if enabled DIS = 0x0 - Select divide by 1. EN = 0x1 - Select divide by 3.
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 IOM 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.
7:1	RSVD	0x0	RO	RESERVED
0	IOCLKEN	0x0	RW	Enable for the interface clock. Must be enabled prior to executing any IO operations.

8.15.2.13SUBMODCTRL Register

Submodule control

OFFSET: 0x00000214

INSTANCE 0 ADDRESS: 0x50004214

INSTANCE 1 ADDRESS: 0x50005214

INSTANCE 2 ADDRESS: 0x50006214

INSTANCE 3 ADDRESS: 0x50007214

INSTANCE 4 ADDRESS: 0x50008214

INSTANCE 5 ADDRESS: 0x50009214

Provides enable for each submodule. Only a sigle submodule can be enabled at one time.

Table 420: SUBMODCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD																								SMOD1TYPE			SMOD1EN	SMOD0TYPE			SMOD0EN

Table 421: SUBMODCTRL Register Bits

Bit	Name	Reset	RW	Description
31:8	RSRVD	0x0	RO	Reserved
7:5	SMOD1TYPE	0x1	RO	Submodule 0 module type. This is the I2C Master interface MSPI = 0x0 - SPI Master submodule I2C_MASTER = 0x1 - MI2C submodule SSPI = 0x2 - SPI Slave submodule SI2C = 0x3 - I2C Slave submodule NA = 0x7 - NOT INSTALLED
4	SMOD1EN	0x0	RW	Submodule 1 enable (1) or disable (0)
3:1	SMOD0TYPE	0x0	RO	Submodule 0 module type. This is the SPI Master interface. SPI_MASTER = 0x0 - MSPI submodule I2C_MASTER = 0x1 - I2C Master submodule SSPI = 0x2 - SPI Slave submodule SI2C = 0x3 - I2C Slave submodule NA = 0x7 - NOT INSTALLED
0	SMOD0EN	0x0	RW	Submodule 0 enable (1) or disable (0)

8.15.2.14CMD Register

Command and offset Register

OFFSET: 0x00000218

INSTANCE 0 ADDRESS: 0x50004218

INSTANCE 1 ADDRESS: 0x50005218

INSTANCE 2 ADDRESS: 0x50006218

INSTANCE 3 ADDRESS: 0x50007218

INSTANCE 4 ADDRESS: 0x50008218

INSTANCE 5 ADDRESS: 0x50009218

Writes to this register will start an IO transaction, as well as set various parameters for the command itself. Reads will return the command value written to the CMD register.

Table 422: 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
OFFSETLO								RSRVD22		CMDSEL		TSIZE												CONT	OFFSETCNT		CMD				

Table 423: CMD Register Bits

Bit	Name	Reset	RW	Description
31:24	OFFSETLO	0x0	RW	This register holds the low order byte of offset to be used in the transaction. The number of offset bytes to use is set with bits 1:0 of the command.
23:22	RSRVD22	0x0	RO	Reserved
21:20	CMDSEL	0x0	RW	Command Specific selection information. Not used in Master I2C. Used as CEn select for Master SPI transactions
19:8	TSIZE	0x0	RW	Defines the transaction size in bytes. The offset transfer is not included in this size.
7	CONT	0x0	RW	Contine to hold the bus after the current transaction if set to a 1 with a new command issued.
6:5	OFFSETCNT	0x0	RW	Number of offset bytes to use for the command - 0, 1, 2, 3 are valid selections. The second (byte 1) and third byte (byte 2) are read from the OFFSETHI register, and the low order byte is pulled from this register in the OFFSETLO field.
4:0	CMD	0x0	RW	Command for submodule. WRITE = 0x1 - Write command using count of offset bytes specified in the OFFSETCNT field READ = 0x2 - Read command using count of offset bytes specified in the OFFSETCNT field TMW = 0x3 - SPI only. Test mode to do constant write operations. Useful for debug and power measurements. Will continually send data in OFFSET field TMR = 0x4 - SPI Only. Test mode to do constant read operations. Useful for debug and power measurements. Will continually read data from external input

8.15.2.15DCX Register

DCX Control Register

OFFSET: 0x0000021C

INSTANCE 0 ADDRESS: 0x5000421C

INSTANCE 1 ADDRESS: 0x5000521C

INSTANCE 2 ADDRESS: 0x5000621C

INSTANCE 3 ADDRESS: 0x5000721C

INSTANCE 4 ADDRESS: 0x5000821C

INSTANCE 5 ADDRESS: 0x5000921C

Enables use of CE signals to transmit DCX level for SPI transactions. Only used in Apollo3 Revision B. For Revision A, this register MUST NOT be programmed!

Table 424: DCX Register

31	30	29	28	27	26	25	24	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																											DCXEN	CE3OUT	CE2OUT	CE1OUT	CE0OUT

Table 425: DCX Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4	DCXEN	0x0	RW	Revision A: MUST NOT be programmed! Revision B: Bit 4: DCX Signaling Enable via other CE signals. The selected DCX signal (unused CE pin) will be driven low during write of offset byte, and high during transmission of data bytes. EN = 0x1 - Enable DCX. DIS = 0x0 - Disable DCX.
3	CE3OUT	0x0	RW	Revision A: MUST NOT be programmed! Revision B: Enable DCX output for CE3 output.
2	CE2OUT	0x0	RW	Revision A: MUST NOT be programmed! Revision B: Enable DCX output for CE2 output.
1	CE1OUT	0x0	RW	Revision A: MUST NOT be programmed! Revision B: Enable DCX output for CE1 output.
0	CE0OUT	0x0	RW	Revision A: MUST NOT be programmed! Revision B: Enable DCX output for CE0 output.

8.15.2.16OFFSETHI Register

High order 2 bytes of 3 byte offset for IO transaction

OFFSET: 0x00000220

INSTANCE 0 ADDRESS: 0x50004220

INSTANCE 1 ADDRESS: 0x50005220

INSTANCE 2 ADDRESS: 0x50006220

INSTANCE 3 ADDRESS: 0x50007220

INSTANCE 4 ADDRESS: 0x50008220

INSTANCE 5 ADDRESS: 0x50009220

High order 2 bytes of 3 byte offset for IO transaction

Table 426: OFFSETHI Register

31	30	29	28	27	26	25	24	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																OFFSETHI															

Table 427: OFFSETHI Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved
15:0	OFFSETHI	0x0	RW	Holds the high order 2 bytes of the 3 byte addressing/offset field to use with IO commands. The number of offset bytes to use is specified in the command register

8.15.2.17CMDSTAT Register

Command status

OFFSET: 0x00000224

INSTANCE 0 ADDRESS: 0x50004224

INSTANCE 1 ADDRESS: 0x50005224

INSTANCE 2 ADDRESS: 0x50006224

INSTANCE 3 ADDRESS: 0x50007224

INSTANCE 4 ADDRESS: 0x50008224

INSTANCE 5 ADDRESS: 0x50009224

Provides staus on the execution of the command currently in progress. The fields in this register will reflect the real time status of the internal state machines and data transfers within the IOM.

Table 428: CMDSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD0												CTSIZE												CMDSTAT			CCMD				

Table 429: CMDSTAT Register Bits

Bit	Name	Reset	RW	Description
31:20	RSRVD0	0x0	RO	Reserved
19:8	CTSIZE	0x0	RO	The current number of bytes still to be transferred with this command. This field will count down to zero.
7:5	CMDSTAT	0x0	RO	The current status of the command execution. ERR = 0x1 - Error encountered with command ACTIVE = 0x2 - Actively processing command IDLE = 0x4 - Idle state, no active command, no error WAIT = 0x6 - Command in progress, but waiting on data from host
4:0	CCMD	0x0	RO	current command that is being executed

8.15.2.18DMATRIGEN Register

DMA Trigger Enable Register

OFFSET: 0x00000240

INSTANCE 0 ADDRESS: 0x50004240

INSTANCE 1 ADDRESS: 0x50005240

INSTANCE 2 ADDRESS: 0x50006240

INSTANCE 3 ADDRESS: 0x50007240

INSTANCE 4 ADDRESS: 0x50008240

INSTANCE 5 ADDRESS: 0x50009240

Provides control on which event will trigger the DMA transfer after the DMA operation is setup and enabled. The trigger event will cause a number of bytes (depending on trigger event) to be

Table 430: DMATRIGEN Register

31	30	29	28	27	26	25	24	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																														DTHREN	DCMDCMPEN

Table 431: DMATRIGEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DTHREN	0x0	RW	Trigger DMA upon THR level reached. For M2P DMA operations (IOM writes), the trigger will assert when the write FIFO has (WTHR/4) number of words free in the write FIFO, and will transfer (WTHR/4) number of words
0	DCMDCMPEN	0x0	RW	Trigger DMA upon command complete. Enables the trigger of the DMA when a command is completed. When this event is triggered, the number of words transferred will be the lesser of the remaining TOTCOUNT bytes, or

8.15.2.19DMATRIGSTAT Register

DMA Trigger Status Register

OFFSET: 0x00000244

INSTANCE 0 ADDRESS: 0x50004244

INSTANCE 1 ADDRESS: 0x50005244

INSTANCE 2 ADDRESS: 0x50006244

INSTANCE 3 ADDRESS: 0x50007244

INSTANCE 4 ADDRESS: 0x50008244

INSTANCE 5 ADDRESS: 0x50009244

Provides the status of trigger events that have occurred for the transaction. Some of the bits are read only and some can be reset via a write of 0.

Table 432: DMATRIGSTAT Register

31	30	29	28	27	26	25	24	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																													DTOTCMP	DTHR	DCMDCMP

Table 433: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.

Table 433: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
2	DTOTCMP	0x0	RO	DMA triggered when DCMDCMP = 0, and the amount of data in the FIFO was enough to complete the DMA operation (greater than or equal to current TOTCOUNT) when the command completed. This trigger is default active when the DCMDCMP trigger is
1	DTHR	0x0	RO	Triggered DMA from THR event. Bit is read only and can be cleared by disabling the DTHR trigger enable or by disabling DMA.
0	DCMDCMP	0x0	RO	Triggered DMA from Command complete event. Bit is read only and can be cleared by disabling the DCMDCMP trigger enable or by disabling DMA.

8.15.2.20DMACFG Register

DMA Configuration Register

OFFSET: 0x00000280

INSTANCE 0 ADDRESS: 0x50004280

INSTANCE 1 ADDRESS: 0x50005280

INSTANCE 2 ADDRESS: 0x50006280

INSTANCE 3 ADDRESS: 0x50007280

INSTANCE 4 ADDRESS: 0x50008280

INSTANCE 5 ADDRESS: 0x50009280

Configuration control of the DMA process, including the direction of DMA, and enablement of DMA

Table 434: DMACFG Register

31	30	29	28	27	26	25	24	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																						DPWROFF	DMAPRI	RSVD						DMADIR	DMAEN

Table 435: DMACFG Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED.
9	DPWROFF	0x0	RW	Power off module after DMA is complete. If this bit is active, the module will request to power off the supply it is attached to. If there are other units still requiring power from the same domain, power down will not be performed. DIS = 0x0 - Power off disabled EN = 0x1 - Power off enabled

Table 435: DMACFG Register Bits

Bit	Name	Reset	RW	Description
8	DMAPRI	0x0	RW	Sets the Priority of the DMA request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
7:2	RSVD	0x0	RO	RESERVED.
1	DMADIR	0x0	RW	Direction P2M = 0x0 - Peripheral to Memory (SRAM) transaction. To be set when doing IOM read operations, ie reading data from external devices. M2P = 0x1 - Memory to Peripheral transaction. To be set when doing IOM write operations, ie writing data to external devices.
0	DMAEN	0x0	RW	DMA Enable. Setting this bit to EN will start the DMA operation. This should be the last DMA related register set prior to issuing the command DIS = 0x0 - Disable DMA Function EN = 0x1 - Enable DMA Function

8.15.2.21DMATOTCOUNT Register

DMA Total Transfer Count

OFFSET: 0x00000288

INSTANCE 0 ADDRESS: 0x50004288

INSTANCE 1 ADDRESS: 0x50005288

INSTANCE 2 ADDRESS: 0x50006288

INSTANCE 3 ADDRESS: 0x50007288

INSTANCE 4 ADDRESS: 0x50008288

INSTANCE 5 ADDRESS: 0x50009288

Contains the number of bytes to be transferred for this DMA transaction. This register is decremented as the data is transferred, and will be 0 at the completion of the DMA operation.

Table 436: DMATOTCOUNT Register

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

Table 437: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:12	RSRVDD	0x0	RO	Reserved

Table 437: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
11:0	TOTCOUNT	0x0	RW	Triggered DMA from Command complete event occured. Bit is read only and can be cleared by disabling the DTHR trigger enable or by disabling DMA.

8.15.2.22DMATARGADDR Register

DMA Target Address Register

OFFSET: 0x0000028C

INSTANCE 0 ADDRESS: 0x5000428C

INSTANCE 1 ADDRESS: 0x5000528C

INSTANCE 2 ADDRESS: 0x5000628C

INSTANCE 3 ADDRESS: 0x5000728C

INSTANCE 4 ADDRESS: 0x5000828C

INSTANCE 5 ADDRESS: 0x5000928C

The source or destination address internal the SRAM for the DMA data. For write operations, this can only be SRAM data (ADDR bit 28 = 1); For read operations, this can ve either SRAM or FLASH (ADDR bit 28 = 0)

Table 438: DMATARGADDR Register

31	30	29	28	27	26	25	24	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			TARGADDR28	RSVD								TARGADDR																			

Table 439: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	Reserved
28	TARGADDR28	0x0	RW	Bit 28 of the target byte address for source of DMA (either read or write). In cases of non-word aligned addresses, the DMA logic will take care for ensuring only the target bytes are read/written.
27:20	RSVD	0x0	RO	Reserved

Table 439: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
19:0	TARGADDR	0x0	RW	Bits [19:0] of the target byte address for source of DMA (either read or write). The address can be any byte alignment, and does not have to be word aligned. In cases of non-word aligned addresses, the DMA logic will take care for ensuring only the target bytes are read/written.

8.15.2.23DMASTAT Register

DMA Status Register

OFFSET: 0x00000290

INSTANCE 0 ADDRESS: 0x50004290

INSTANCE 1 ADDRESS: 0x50005290

INSTANCE 2 ADDRESS: 0x50006290

INSTANCE 3 ADDRESS: 0x50007290

INSTANCE 4 ADDRESS: 0x50008290

INSTANCE 5 ADDRESS: 0x50009290

Status of the DMA operation currently in progress.

Table 440: DMASTAT Register

31	30	29	28	27	26	25	24	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																													DMAERR	DMACPL	DMATIP

Table 441: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	DMAERR	0x0	RW	DMA Error. This active high bit signals an error was encountered during the DMA operation. The bit can be cleared by writing to 0. Once set, this bit will remain set until cleared by software.
1	DMACPL	0x0	RW	DMA Transfer Complete. This signals the end of the DMA operation. This bit can be cleared by writing to 0, and will also be cleared when a new DMA is started.
0	DMATIP	0x0	RO	DMA Transfer In Progress indicator. 1 will indicate that a DMA transfer is active. The DMA transfer may be waiting on data, transferring data, or waiting for priority.

8.15.2.24CQCFG Register

Command Queue Configuration Register

OFFSET: 0x00000294

INSTANCE 0 ADDRESS: 0x50004294

INSTANCE 1 ADDRESS: 0x50005294

INSTANCE 2 ADDRESS: 0x50006294

INSTANCE 3 ADDRESS: 0x50007294

INSTANCE 4 ADDRESS: 0x50008294

INSTANCE 5 ADDRESS: 0x50009294

Controls parameters and options for execution of the command queue operation. To enable command queue, create this in memory, set the address, and enable it with a write to CQEN.

Table 442: CQCFG Register

31	30	29	28	27	26	25	24	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																														CQPRI	CQEN

Table 443: CQCFG Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	CQPRI	0x0	RW	Sets the Priority of the command queue dma request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
0	CQEN	0x0	RW	Command queue enable. When set, will enable the processing of the command queue and fetches of address/data pairs will proceed from the word address within the CQADDR register. Can be disabled DIS = 0x0 - Disable CQ Function EN = 0x1 - Enable CQ Function

8.15.2.25CQADDR Register

CQ Target Read Address Register

OFFSET: 0x00000298

INSTANCE 0 ADDRESS: 0x50004298

INSTANCE 1 ADDRESS: 0x50005298

INSTANCE 2 ADDRESS: 0x50006298

INSTANCE 3 ADDRESS: 0x50007298

INSTANCE 4 ADDRESS: 0x50008298

INSTANCE 5 ADDRESS: 0x50009298

The SRAM address in this register is fetched on next execution of the CQ operation. This register is updated as the CQ operation progresses, and is the live version of the register. The register can also be written by the Command Queue operation itself, allowing the relocation of successive CQ fetches. In this case, the new CQ address will be used for the next CQ address/data fetch.

Table 444: CQADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD2			CQADDR28	RSRVD1								CQADDR																		RSRVD0	

Table 445: CQADDR Register Bits

Bit	Name	Reset	RW	Description
31:29	RSRVD2	0x0	RO	Reserved
28	CQADDR28	0x0	RW	Bit 28 of target byte address for source of CQ (read only). Used to denote Flash (0) or SRAM (1) access
27:20	RSRVD1	0x0	RO	Reserved
19:2	CQADDR	0x0	RW	Bits 19:2 of target byte address for source of CQ (read only). The buffer must be aligned on a word boundary
1:0	RSRVD0	0x0	RO	Reserved

8.15.2.26CQSTAT Register

Command Queue Status Register

OFFSET: 0x0000029C

INSTANCE 0 ADDRESS: 0x5000429C

INSTANCE 1 ADDRESS: 0x5000529C

INSTANCE 2 ADDRESS: 0x5000629C

INSTANCE 3 ADDRESS: 0x5000729C

INSTANCE 4 ADDRESS: 0x5000829C

INSTANCE 5 ADDRESS: 0x5000929C

Provides the status of the command queue operation. If the command queue is disabled, these bits will be cleared. The bits are read only.

Table 446: CQSTAT Register

31	30	29	28	27	26	25	24	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																													CQERR	CQPAUSED	CQTIP

Table 447: CQSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	CQERR	0x0	RW	Command queue processing Error. This active high bit signals that an error was encountered during the CQ operation.
1	CQPAUSED	0x0	RO	Command queue operation is currently paused.
0	CQTIP	0x0	RO	Command queue Transfer In Progress indicator. 1 will indicate that a CQ transfer is active and this will remain active even when paused waiting for external event.

8.15.2.27CQFLAGS Register

Command Queue Flag Register

OFFSET: 0x000002A0

INSTANCE 0 ADDRESS: 0x500042A0

INSTANCE 1 ADDRESS: 0x500052A0

INSTANCE 2 ADDRESS: 0x500062A0

INSTANCE 3 ADDRESS: 0x500072A0

INSTANCE 4 ADDRESS: 0x500082A0

INSTANCE 5 ADDRESS: 0x500092A0

Command Queue Flag Register

Table 448: CQFLAGS Register

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

Table 449: CQFLAGS Register Bits

Bit	Name	Reset	RW	Description
31:16	CQIRQMASK	0x0	RW	Mask the bits used to generate the command queue interrupt. A '1' in the bit position will enable the pause event to trigger the interrupt, if the CQWT_int interrupt is enabled. Bits definitions are the same as CQPAUSE
15:0	CQFLAGS	0x0	RO	Current flag status (read-only). Bits [7:0] are software controllable and bits [15:8] are hardware status.

8.15.2.28CQSETCLEAR Register

Command Queue Flag Set/Clear Register

OFFSET: 0x000002A4

INSTANCE 0 ADDRESS: 0x500042A4

INSTANCE 1 ADDRESS: 0x500052A4

INSTANCE 2 ADDRESS: 0x500062A4

INSTANCE 3 ADDRESS: 0x500072A4

INSTANCE 4 ADDRESS: 0x500082A4

INSTANCE 5 ADDRESS: 0x500092A4

Set/Clear the command queue software pause flags on a per-bit basis. Contains 3 fields, allowing for setting, clearing or toggling the value in the software flags. Priority when the same bit

Table 450: CQSETCLEAR Register

31	30	29	28	27	26	25	24	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								CQFCLR								CQFTGL								CQFSET							

Table 451: CQSETCLEAR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	Reserved
23:16	CQFCLR	0x0	WO	Clear CQFlag status bits. Will clear to 0 any SWFLAG with a '1' in the corresponding bit position of this field
15:8	CQFTGL	0x0	WO	Toggle the indicated bit. Will toggle the value of any SWFLAG with a '1' in the corresponding bit position of this field
7:0	CQFSET	0x0	WO	Set CQFlag status bits. Will set to 1 the value of any SWFLAG with a '1' in the corresponding bit position of this field

8.15.2.29CQPAUSEEN Register

Command Queue Pause Enable Register

OFFSET: 0x000002A8

INSTANCE 0 ADDRESS: 0x500042A8

INSTANCE 1 ADDRESS: 0x500052A8

INSTANCE 2 ADDRESS: 0x500062A8

INSTANCE 3 ADDRESS: 0x500072A8

INSTANCE 4 ADDRESS: 0x500082A8

INSTANCE 5 ADDRESS: 0x500092A8

Enables a flag to pause an active command queue operation. If a bit is '1' and the corresponding bit in the CQFLAG register is '1', CQ processing will halt until either value is changed to '0'.

Table 452: CQPAUSEEN Register

31	30	29	28	27	26	25	24	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																CQPEN															

Table 453: CQPAUSEEN Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	Reserved

Table 453: CQPAUSEEN Register Bits

Bit	Name	Reset	RW	Description
15:0	CQPEN	0x0	RW	Enables the specified event to pause command processing when active IDXEQ = 0x8000 - Pauses the command queue when the current index matches the last index BLEXOREN = 0x4000 - Pause command queue when input BLE bit XORed with SWFLAG4 is '1' IOMXOREN = 0x2000 - Pause command queue when input IOM bit XORed with SWFLAG3 is '1' GPIOXOREN = 0x1000 - Pause command queue when input GPIO irq_bit XORed with SWFLAG2 is '1' MSPI1XNOREN = 0x800 - Pause command queue when input MSPI1 bit XNORed with SWFLAG1 is '1' MSPI0XNOREN = 0x400 - Pause command queue when input MSPI0 bit XNORed with SWFLAG0 is '1' MSPI1XOREN = 0x200 - Pause command queue when input MSPI1 bit XORed with SWFLAG1 is '1' MSPI0XOREN = 0x100 - Pause command queue when input MSPI0 bit XORed with SWFLAG0 is '1' SWFLAGEN7 = 0x80 - Pause the command queue when software flag bit 7 is '1'. SWFLAGEN6 = 0x40 - Pause the command queue when software flag bit 6 is '1' SWFLAGEN5 = 0x20 - Pause the command queue when software flag bit 5 is '1' SWFLAGEN4 = 0x10 - Pause the command queue when software flag bit 4 is '1' SWFLAGEN3 = 0x8 - Pause the command queue when software flag bit 3 is '1' SWFLAGEN2 = 0x4 - Pause the command queue when software flag bit 2 is '1' SWFLAGEN1 = 0x2 - Pause the command queue when software flag bit 1 is '1' SWFLAGEN0 = 0x1 - Pause the command queue when software flag bit 0 is '1'

8.15.2.30CQCURIDX Register

IOM Command Queue current index value . Compared to the CQENDIDX reg contents to generate the IDXEQ Pause event for command queue

OFFSET: 0x000002AC

INSTANCE 0 ADDRESS: 0x500042AC

INSTANCE 1 ADDRESS: 0x500052AC

INSTANCE 2 ADDRESS: 0x500062AC

INSTANCE 3 ADDRESS: 0x500072AC

INSTANCE 4 ADDRESS: 0x500082AC

INSTANCE 5 ADDRESS: 0x500092AC

Current index value, targeted to be written by register write operations within the command queue. This is compared to the CQENDIDX and will stop the CQ operation if bit 15 of the CQPAUSEEN is '1' and

Table 454: CQCURIDX Register

31	30	29	28	27	26	25	24	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																								CQCURIDX							

Table 455: CQCURIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:0	CQCURIDX	0x0	RW	Holds 8 bits of data that will be compared with the CQENDIX register field. If the values match, the IDXEQ pause event will be activated, which will cause the pausing of command quue operation if the IDXEQ bit is enabled in CQPAUSEEN.

8.15.2.31CQENDIDX Register

IOM Command Queue current index value . Compared to the CQCURIDX reg contents to generate the IDXEQ Pause event for command queue

OFFSET: 0x000002B0

INSTANCE 0 ADDRESS: 0x500042B0

INSTANCE 1 ADDRESS: 0x500052B0

INSTANCE 2 ADDRESS: 0x500062B0

INSTANCE 3 ADDRESS: 0x500072B0

INSTANCE 4 ADDRESS: 0x500082B0

INSTANCE 5 ADDRESS: 0x500092B0

End index value, targeted to be written by software to indicate the last valid register pair contained within the command queue. Register write operations within the command queue.

Table 456: CQENDIDX Register

31	30	29	28	27	26	25	24	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																								CQENDIDX							

Table 457: CQENDIDX Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED

Table 457: CQENDIDX Register Bits

Bit	Name	Reset	RW	Description
7:0	CQENDIDX	0x0	RW	Holds 8 bits of data that will be compared with the CQCURIX register field. If the values match, the IDXEQ pause event will be activated, which will cause the pausing of command quue operation if the IDXEQ bit is enabled in CQPAUSEEN.

8.15.2.32STATUS Register

IOM Module Status Register

OFFSET: 0x000002B4

INSTANCE 0 ADDRESS: 0x500042B4

INSTANCE 1 ADDRESS: 0x500052B4

INSTANCE 2 ADDRESS: 0x500062B4

INSTANCE 3 ADDRESS: 0x500072B4

INSTANCE 4 ADDRESS: 0x500082B4

INSTANCE 5 ADDRESS: 0x500092B4

IOM Module Status Register

Table 458: 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 459: STATUS Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	IDLEST	0x0	RO	indicates if the active I/O state machine is IDLE. Note - The state machine could be in idle state due to holdoffs from data availability, or as the command gets propagated into the logic from the registers. IDLE = 0x1 - The I/O state machine is in the idle state.
1	CMDACT	0x0	RO	Indicates if the active I/O Command is currently processing a transaction, or command is complete, but the FIFO pointers are still syncronizing internally. This bit will go high at ACTIVE = 0x1 - An I/O command is active. Indicates the active module has an active command and is processing this. De-asserted when the command is completed.

Table 459: STATUS Register Bits

Bit	Name	Reset	RW	Description
0	ERR	0x0	RO	Bit has been deprecated. Please refer to the other error indicators. This will always return 0. ERROR = 0x1 - Bit has been deprecated and will always return 0.

8.15.2.33MSPICFG Register

SPI module master configuration

OFFSET: 0x00000300

INSTANCE 0 ADDRESS: 0x50004300

INSTANCE 1 ADDRESS: 0x50005300

INSTANCE 2 ADDRESS: 0x50006300

INSTANCE 3 ADDRESS: 0x50007300

INSTANCE 4 ADDRESS: 0x50008300

INSTANCE 5 ADDRESS: 0x50009300

Controls the configuration of the SPI master module, including POL/PHA, LSB, flow control, and delays for MISO and MOSI

Table 460: MSPICFG Register

31	30	29	28	27	26	25	24	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	MSPIRST	DOUTDLY			DINDLY			SPILSB	RDFCPOL	WTF-	WTFCIRQ	RSVD	MOSIINV	RDFC	WTFC	RSVD													FULLDUP	SPHA	SPOL

Table 461: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	MSPIRST	0x0	RW	Not used. To reset the module, toggle the SMOD_EN for the module
29:27	DOUTDLY	0x0	RW	Delay tap to use for the output signal (MOSI). This give more hold time on the output data
26:24	DINDLY	0x0	RW	Delay tap to use for the input signal (MISO). This gives more hold time on the input data.

Table 461: MSPICFG Register Bits

Bit	Name	Reset	RW	Description
23	SPILSB	0x0	RW	Selects data transfer as MSB first (0) or LSB first (1) for the data portion of the SPI transaction. The offset bytes are always transmitted MSB first. MSB = 0x0 - Send and receive MSB bit first LSB = 0x1 - Send and receive LSB bit first
22	RDFCPOL	0x0	RW	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.
21	WTFCPOL	0x1	RW	selects the write flow control signal polarity. The transfers are halted when the selected flow control signal is OPPOSITE polarity of bit. (For example: WTFCPOL = 0 will allow a IRQ=1 to pause transfers). HIGH = 0x0 - Flow control signal high(1) creates flow control and byte transfers will stop until the flow control signal goes low. LOW = 0x1 - Flow control signal low(0) creates flow control and byte transfers will stop until the flow control signal goes high(1).
20	WTFCIRQ	0x0	RW	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.
19	RSVD	0x0	RO	Reserved
18	MOSIINV	0x0	RW	inverts 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.
17	RDFC	0x0	RW	enables read mode flow control. DIS = 0x0 - Read mode flow control disabled. EN = 0x1 - Read mode flow control enabled.
16	WTFC	0x0	RW	enables write mode flow control. DIS = 0x0 - Write mode flow control disabled. EN = 0x1 - Write mode flow control enabled.
15:3	RSVD	0x0	RO	RESERVED
2	FULLDUP	0x0	RW	Enables full duplex mode for Master SPI write operations. Data will be captured simultaneously into the read fifo
1	SPHA	0x0	RW	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.
0	SPOL	0x0	RW	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.

8.15.2.34MI2CCFG Register

I2C Master configuration

OFFSET: 0x00000400

INSTANCE 0 ADDRESS: 0x50004400

INSTANCE 1 ADDRESS: 0x50005400

INSTANCE 2 ADDRESS: 0x50006400

INSTANCE 3 ADDRESS: 0x50007400

INSTANCE 4 ADDRESS: 0x50008400

INSTANCE 5 ADDRESS: 0x50009400

Controls the configuration of the I2C bus master.

Table 462: MI2CCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSRVD3							STRDIS	SMPCNT								SDAENDLY				SCLENDLY				RSRVD2	MI2CRST	SDADLY		RSRVD1	ARBEN	I2CLSB	ADDRSZ

Table 463: MI2CCFG Register Bits

Bit	Name	Reset	RW	Description
31:25	RSRVD3	0x0	RO	Reserved
24	STRDIS	0x0	RW	Disable detection of clock stretch events smaller than 1 cycle
23:16	SMPCNT	0x0	RW	Number of Base clk cycles to wait before sampling the SCL clock to determine if a clock stretch event has occured
15:12	SDAENDLY	0x0	RW	Number of IOCLK cycles to delay the SDA output en (all transitions affected). Used to delay data relative to clock
11:8	SCLENDLY	0x0	RW	Number of IOCLK cycles to delay the rising edge of the SCL output en (clock will go low on this edge). Used to allow clock shaping.
7	RSRVD2	0x0	RO	Reserved
6	MI2CRST	0x0	RW	Not used. To reset the module, toggle the SMOD_EN for the module
5:4	SDADLY	0x0	RW	Delay to enable on the SDA output. Values are 0x0-0x3.
3	RSRVD1	0x0	RO	Reserved

Table 463: MI2CCFG Register Bits

Bit	Name	Reset	RW	Description
2	ARBEN	0x0	RW	Enables multi-master arbitration for the I2C master. If the bus is known to have only a single master, this function can be disabled to save clock cycles on I2C transactions ARBEN = 0x1 - Enable multi-master bus arbitration support for this i2c master ARBDIS = 0x0 - Disable multi-master bus arbitration support for this i2c master
1	I2CLSB	0x0	RW	Direction of data transmit and receive, MSB(0) or LSB(1) first. Default per I2C specification is MSB first. This applies to both read and write data, and read data will be bit MSBFIRST = 0x0 - Byte data is transmitted MSB first onto the bus/read from the bus LSBFIRST = 0x1 - Byte data is transmitted LSB first onto the bus/read from the bus
0	ADDRSZ	0x0	RW	Sets the I2C master device address size to either 7b (0) or 10b (1). ADDRSZ7 = 0x0 - Use 7b addressing for I2C master transactions ADDRSZ10 = 0x1 - Use 10b addressing for I2C master transactions

8.15.2.35DEVCFG Register

I2C Device Configuration register

OFFSET: 0x00000404

INSTANCE 0 ADDRESS: 0x50004404

INSTANCE 1 ADDRESS: 0x50005404

INSTANCE 2 ADDRESS: 0x50006404

INSTANCE 3 ADDRESS: 0x50007404

INSTANCE 4 ADDRESS: 0x50008404

INSTANCE 5 ADDRESS: 0x50009404

Contains the I2C device address.

Table 464: DEVCFG Register

31	30	29	28	27	26	25	24	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																						DEVADDR									

Table 465: DEVCFG Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	Reserved

Table 465: DEVCFG Register Bits

Bit	Name	Reset	RW	Description
9:0	DEVADDR	0x0	RW	I2C address of the device that the Master will use to target for read/write operations. This can be either a 7b or 10b address.

8.15.2.36IOMDBG Register

IOM Debug Register

OFFSET: 0x00000410

INSTANCE 0 ADDRESS: 0x50004410

INSTANCE 1 ADDRESS: 0x50005410

INSTANCE 2 ADDRESS: 0x50006410

INSTANCE 3 ADDRESS: 0x50007410

INSTANCE 4 ADDRESS: 0x50008410

INSTANCE 5 ADDRESS: 0x50009410

Debug control

Table 466: IOMDBG Register

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

Table 467: IOMDBG Register Bits

Bit	Name	Reset	RW	Description
31:3	DBGDATA	0x0	RW	Debug control for various options. DBGDATA[1:0] is used to select between different debug data available in the DBG0 and DBG1 registers.
2	APBCLKON	0x0	RW	APBCLK debug clock control. Enable APB_CLK to be active when this bit is '1'. Otherwise, the clock is controlled with gating from the logic as needed.
1	IOCLKON	0x0	RW	IOCLK debug clock control. Enable IO_CLK to be active when this bit is '1'. Otherwise, the clock is controlled with gating from the logic as needed.
0	DBGEN	0x0	RW	Debug Enable. Setting bit will enable the update of data within this register, otherwise it is clock gated for power savings

9. I2C/SPI Slave Module

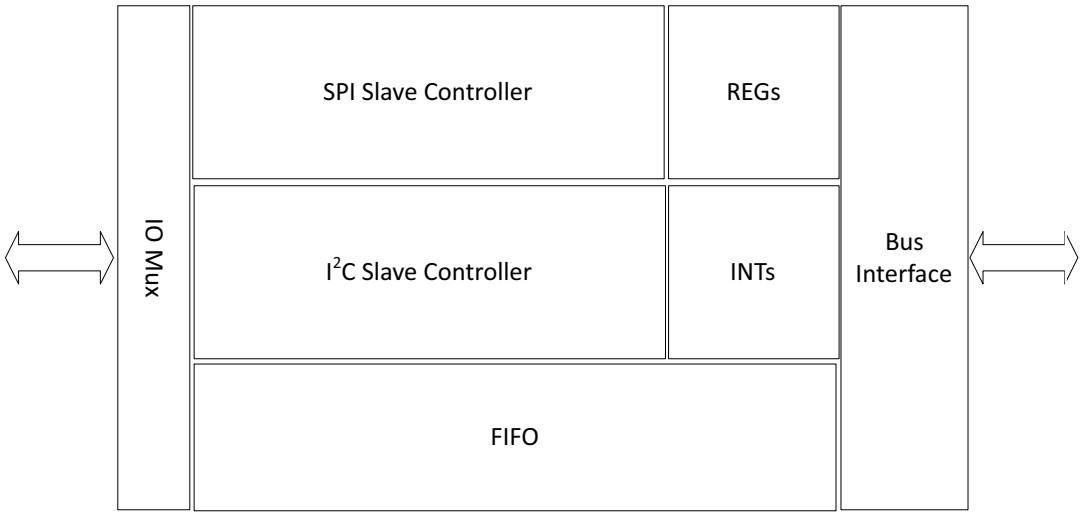


Figure 42. Block diagram for the I2C/SPI Slave Module

9.1 Functional Overview

The I2C/SPI Slave Module, shown in Figure 42, allows the Apollo3 Blue MCU to function as a Slave in an I2C or SPI system. The I2C/SPI Slave operates in an independent fashion, so that the Apollo3 Blue MCU 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 I2C/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 I2C mode the Slave supports fully configurable 7 and 10-bit addressing with interface timing limits as specified in Table 1155. In SPI mode, the Slave supports all polarity/phase combinations and interface frequencies as specified in Table 1156.

9.2 Local RAM Allocation

The I2C/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 I2C/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 (REG_IOSLAVE_FIFOCFG_FIFOBASE) should contain a value that rep-

resents 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 REG_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 Apollo3 Blue 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, REG_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, REG_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 43 below shows the LRAM address mapping between the I/O interface and the AHB.

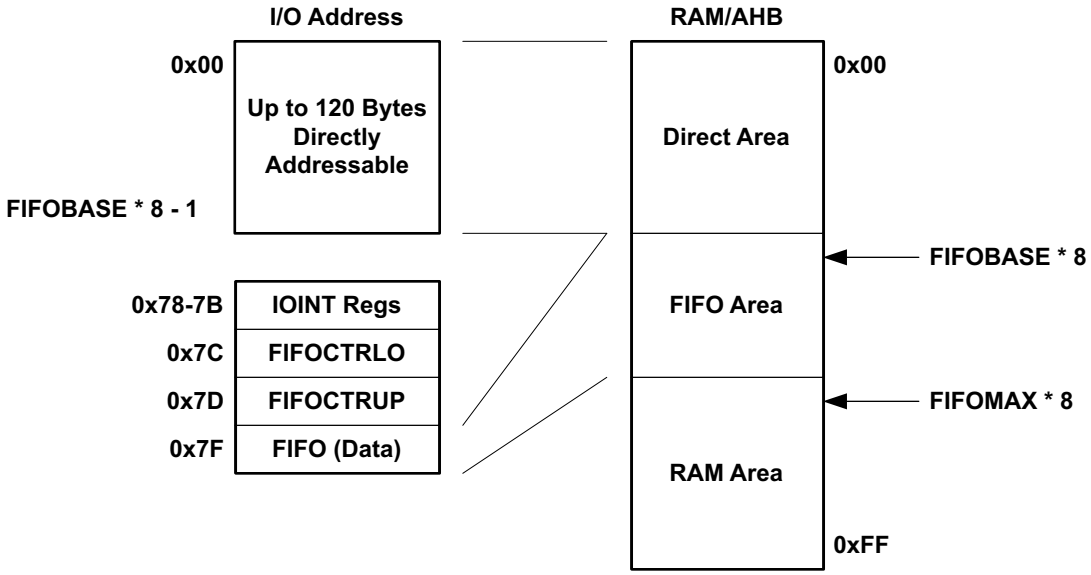


Figure 43. I²C/SPI Slave Module LRAM Addressing

9.3 Direct Area Functions

The Direct Area is used for direct communications between the interface Host and the Apollo3 Blue MCU. 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 I2C or SPI interface in the Direct Area with an offset address which corresponds to a particular REGACC interrupt. Table 468 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 Apollo3 Blue MCU. 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 (i.e. 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 468 lists the offsets to memory locations within the Direct Address Space and corresponding interrupt bit settings in the REGACCINTSTAT register.

Table 468: 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
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 REG_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 (i.e. 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 (REG_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.

9.4 FIFO Area Functions

The FIFO is used to provide very efficient flow of data from the Apollo3 Blue 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 44 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 REG_IOSLAVE_FIFOPTR_FIFOPTR register points to the next byte to be read on the I/O interface. REG_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 REG_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.

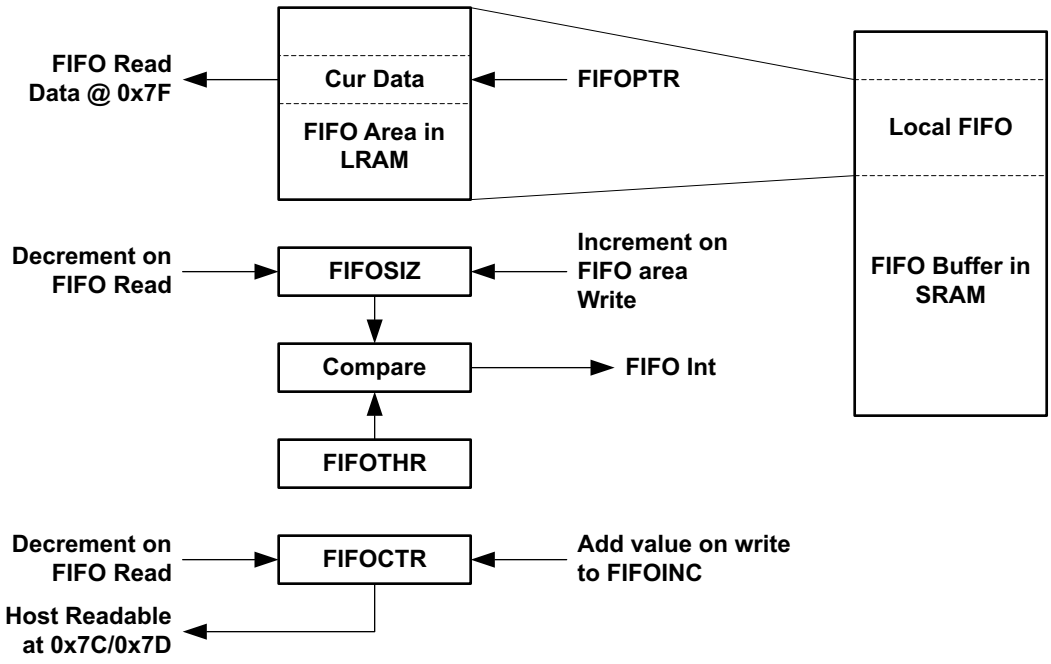


Figure 44. I²C/SPI Slave Module FIFO

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 I²C Slave interrupt block and in the Host interrupt block.

When FIFOSIZ drops below the configured threshold REG_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 I²C/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 REG_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 I²C/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.

9.5 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 Apollo3 Blue MCU 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 REG_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 REG_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.

9.6 Interface Interrupts

The CPU may also signal the Host via the IOINT interrupt, which may be connected to an Apollo3 Blue MCU 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 I2C/SPI Slave includes a mechanism to allow the Host CPU and the Apollo3 Blue MCU to each interrupt the other via a set of eight interrupts. The Host CPU accesses these interrupts via interface locations 0x78-0x7B, and the Apollo3 Blue MCU 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 Apollo3 Blue MCU interrupts in all modules. The Apollo3 Blue MCU 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 469 summarizes these I/O interface interrupts and how they can be controlled and read.

Table 469: I/O Interface Interrupt Control

RAM Location	IOINT Register[1]	Function	MCU 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 MCU
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 MCU
0x7A	IOINTCLR	I/O Interrupt Clear	IOINTCTL_IOINTCLR (W/O	Each interrupt can be individually cleared by the I/O Host, but the MCU 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 MCU

1. Readable by the I/O Host

The Apollo3 Blue MCU 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 Apollo3 Blue MCU to generate a software interrupt to the Host device. In addition, a FIFO underflow interrupt FUNDFL in the I2C/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 Apollo3 Blue MCU 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 I2C/SPI INTSTAT Register. This allows Apollo3 Blue MCU 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.

9.7 Command Completion Interrupts

Four interrupts in the I2C/SPI Slave module are generated when the Host interface device completes a transfer. This allows Apollo3 Blue MCU to be easily awakened for any transfer from the Host while maximizing the time Apollo3 Blue MCU 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.

NOTE: 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.

NOTE: 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.

9.8 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 is referring to accesses from the Host.

9.9 I^2C Interface

The Apollo3 Blue MCU I^2C Slave interface operates as a standard slave. The device is accessed at an address configured in the REG_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 REG_IOSLAVE_IOSCFG_10BIT. The I^2C 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 Apollo3 Blue MCU I^2C 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 45) and are described in the following sections.

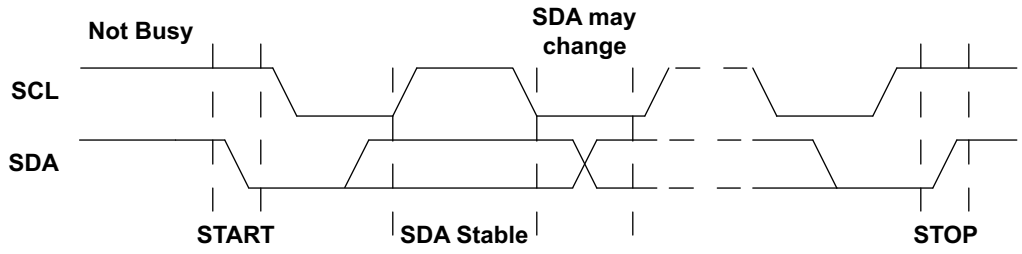


Figure 45. Basic I2C Conditions

9.9.1 Bus Not Busy

Both SDA and SCL remain high.

9.9.2 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.

9.9.3 Stop Data Transfer

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

9.9.4 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 byte-wide and each receiver acknowledges with a ninth bit.

9.9.5 Acknowledge

Each byte of eight bits is followed by one Acknowledge (ACK) bit as shown in Figure 46. 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.

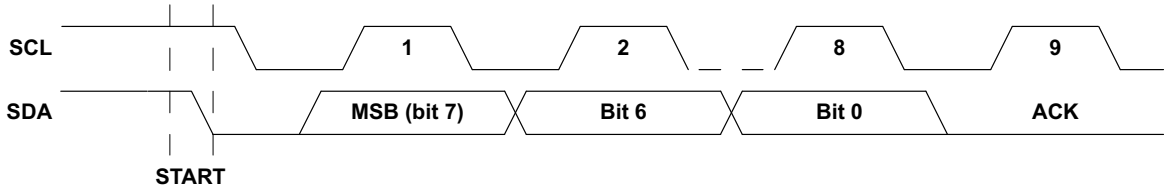


Figure 46. I2C Acknowledge

9.9.6 Address Operation

In I2C mode, the I2C/SPI Slave supports either 7-bit or 10-bit addressing, selected by the 10BIT bit in the IOSCFG Register. Figure 47 shows the operation in 7-bit mode in which the master addresses the Apollo3 Blue MCU with a 7-bit address configured as 0xD2 in the I2CADDR field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of the I2CADDR field, the Apollo3 Blue MCU is selected, the eighth bit indicate a write (RW = 0) or a read (RW = 1) operation and the Apollo3 Blue MCU supplies the ACK. The Apollo3 Blue MCU ignores all other address values and does not respond with an ACK.

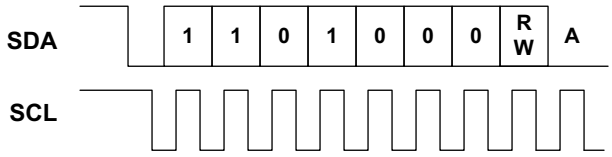


Figure 47. I2C 7-bit Address Operation

Figure 48 shows the operation with which the master addresses the Apollo3 Blue MCU 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 Apollo3 Blue MCU again supplies the ACK. The I2C/SPI Slave ignores all other address values and does not respond with an ACK.

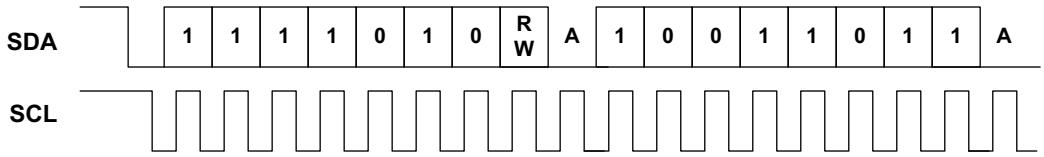


Figure 48. I2C 10-bit Address Operation

9.9.7 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 49. This value is loaded into the Address Pointer of the I2C/SPI Slave.

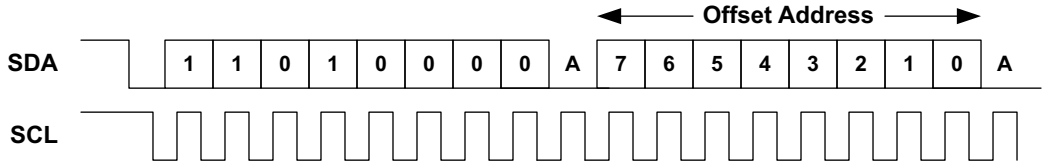


Figure 49. I2C Offset Address Transmission

9.9.8 Write Operation

In a write operation the master transmitter transmits to the Apollo3 Blue MCU slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address as in Figure 49. 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 50. Note that if the Address Pointer is at 0x7F, it will not increment on the write.

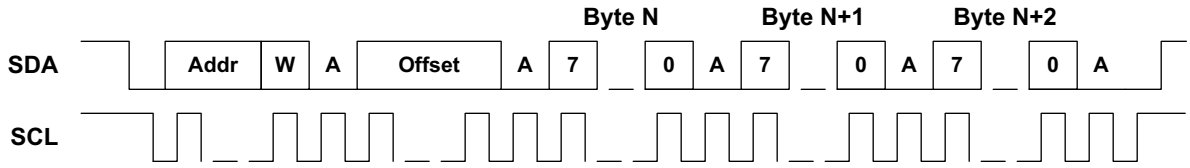


Figure 50. I2C Write Operation

9.9.9 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 Apollo3 Blue MCU but with the RW bit as a 1 indicating a read operation. Figure 51 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.

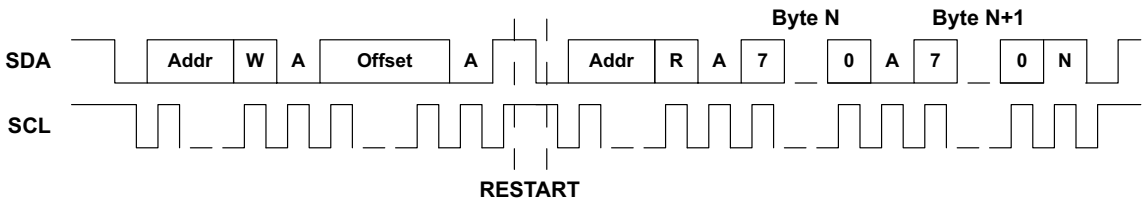


Figure 51. I2C Read Operation

9.9.10 General Address Detection

The I2C/SPI Slave may be configured to detect an I2C General Address (0x00) write. If this address is detected, the first data byte written is stored in the REG_IOSLAVE_GADATA Register and the GENAD interrupt flag is set. This allows software to create the appropriate response, which is typically to reset the I2C/SPI Slave.

9.10 SPI Interface

The I2C/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 I2C/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 I2C/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 I2C/SPI Slave supports clock frequencies up to 12 MHz, and responds to 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.

9.10.1 Write Operation

Figure 52 shows a SPI write operation. The operation is initiated when the nCE signal to the Apollo3 Blue MCU 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 I2C/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 I2C/SPI Slave. Note also that if the Address Pointer is 0x7F, it does not increment on the read.

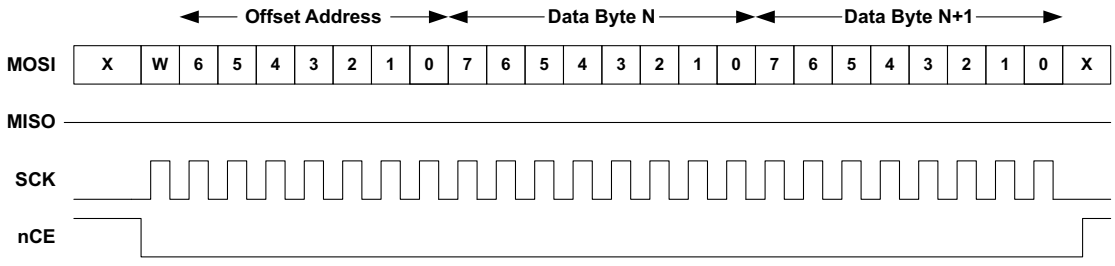


Figure 52. SPI Write Operation

9.10.2 Read Operation

Figure 53 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 I2C/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.

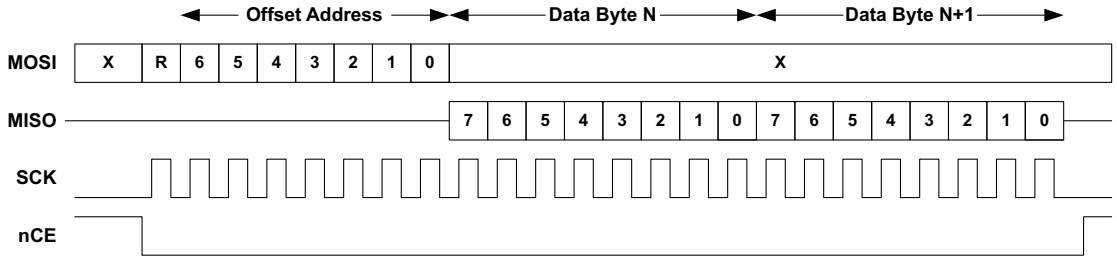


Figure 53. SPI Read Operation

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

The I2C/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 I2C/SPI Slave itself.

9.10.4 SPI Polarity and Phase

The I2C/SPI Slave supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode. Figure 54 shows how these two bits affect the interface signal behavior.

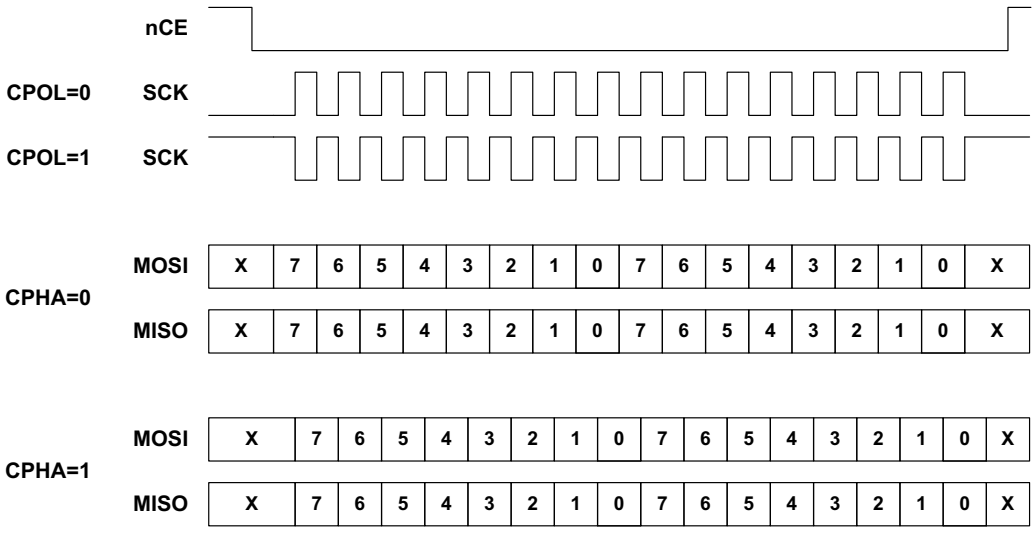


Figure 54. SPI CPOL and CPHA

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, REG_IOSLAVE_IOSCFG_SPOL must be set to 0. If CPOL ≠ CPHA, SPOL must be set to 1.

9.11 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 REG_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.

9.12 Wakeup Using the I^2C/SPI Slave

The I^2C/SPI Slave can continue to operate even if the Apollo3 Blue MCU CPU is in Sleep or Deep Sleep mode. The hardware will enable and disable the I^2C/SPI Slave clock and oscillators as necessary. The only consideration in this environment is when the MCU is in a deep sleep mode, such that the HFRC Oscillator is powered down, and a master attempts to access the I^2C/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 I^2C mode.

For I^2C 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 Apollo3 Blue MCU prior to initiating any SPI transfers.

There is no delay restriction if the MCU is in normal Sleep mode. In that case the HFRC is not powered down and the I^2C/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 Apollo3 Blue MCU 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.

9.13 IOSLAVE Registers

I2C/SPI Slave

INSTANCE 0 BASE ADDRESS:0x50000000

9.13.1 Register Memory Map

Table 470: IOSLAVE Register Map

Address(s)	Register 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

9.13.2 IOSLAVE Registers

9.13.2.1 FIFOPTR Register

Current FIFO Pointer

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x50000100

Current FIFO Pointer

Table 471: 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 472: 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.

9.13.2.2 FIFOCFG Register

FIFO Configuration

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x50000104

FIFO Configuration

Table 473: 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 474: 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 (FIFOBASE*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).

9.13.2.3 FIFOTHR Register

FIFO Threshold Configuration

OFFSET: 0x00000108

INSTANCE 0 ADDRESS: 0x50000108

FIFO Threshold Configuration

Table 475: 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 476: FIFOTHR Register Bits

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

9.13.2.4 FUPD Register

FIFO Update Status

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x5000010C

FIFO Update Status

Table 477: 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 478: FUPD Register Bits

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

9.13.2.5 FIFOCTR Register

Overall FIFO Counter

OFFSET: 0x00000110

INSTANCE 0 ADDRESS: 0x50000110

Overall FIFO Counter

Table 479: 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 480: FIFOCTR Register Bits

Bit	Name	Reset	RW	Description
31:10	RSVD	0x0	RO	RESERVED

Table 480: FIFOCTR Register Bits

Bit	Name	Reset	RW	Description
9:0	FIFOCTR	0x0	RW	Virtual FIFO byte count

9.13.2.6 FIFOINC Register

Overall FIFO Counter Increment

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x50000114

Overall FIFO Counter Increment

Table 481: 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 482: 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

9.13.2.7 CFG Register

I/O Slave Configuration

OFFSET: 0x00000118

INSTANCE 0 ADDRESS: 0x50000118

I/O Slave Configuration

Table 483: 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 484: 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
19:8	I2CADDR	0x0	RW	7-bit or 10-bit I2C device address.
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 I2C interface for the IO Slave. SPI = 0x1 - Selects SPI interface for the IO Slave.

9.13.2.8 PRENC Register

I/O Slave Interrupt Priority Encode

OFFSET: 0x0000011C

INSTANCE 0 ADDRESS: 0x5000011C

I/O Slave Interrupt Priority Encode

Table 485: 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 486: 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.

9.13.2.9 IOINTCTL Register

I/O Interrupt Control

OFFSET: 0x00000120

INSTANCE 0 ADDRESS: 0x50000120

I/O Interrupt Control

Table 487: 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 488: 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.

9.13.2.10GENADD Register

General Address Data

OFFSET: 0x00000124

INSTANCE 0 ADDRESS: 0x50000124

General Address Data

Table 489: 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 490: 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.

9.13.2.11INTEN 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 491: 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 492: 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.

Table 492: INTEN Register Bits

Bit	Name	Reset	RW	Description
5	IOINTW	0x0	RW	IO Write interrupt.
4	GENAD	0x0	RW	I2C 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.

9.13.2.12INTSTAT 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 493: 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 494: 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 494: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
4	GENAD	0x0	RW	I2C 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.

9.13.2.13INTCLR 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 495: 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 496: 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	I2C General Address interrupt.

Table 496: INTCLR Register Bits

Bit	Name	Reset	RW	Description
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.

9.13.2.14INTSET 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 497: 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 498: 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	I2C General Address interrupt.

Table 498: INTSET Register Bits

Bit	Name	Reset	RW	Description
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.

9.13.2.15REGACCINTEN 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 499: 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 500: REGACCINTEN Register Bits

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

9.13.2.16REGACCINTSTAT 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 501: 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 502: REGACCINTSTAT Register Bits

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

9.13.2.17REGACCINTCLR 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 503: 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 504: REGACCINTCLR Register Bits

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

9.13.2.18REGACCINTSET 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 505: 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 506: REGACCINTSET Register Bits

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

9.14 Host Side Address Space and Register

9.14.1 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 refers to accesses from the Host.

9.14.1.1 HOST_IER Register

Host Interrupt Enable

OFFSET: 0x78

This register enables the FIFO read interrupts.

Table 507: HOST_IER Register

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

Table 508: 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

9.14.1.2 HOST_ISR Register

Host Interrupt Status Register

OFFSET: 0x79

The host uses this register to read interrupt status.

Table 509: HOST_ISR Register

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

Table 510: 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.

9.14.1.3 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 511: HOST_WCR Register

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

Table 512: 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
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

9.14.1.4 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 513: HOST_WCS Register

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

Table 514: 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

9.14.1.5 FIFOCTRLO Register

FIFOCTR Low Byte

OFFSET: 0x7C

This register allows the host to read the lower eight bits of the FIFOCTR register.

Table 515: FIFOCTRLO Register

07	06	05	04	03	02	01	00
FIFOCTRLO							

Table 516: FIFOCTRLO Register Bits

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

9.14.1.6 FIFOCTRUP Register

FIFOCTR Upper Byte

OFFSET: 0x7D

This register allows the host to read the upper two bits of the FIFOCTR register.

Table 517: FIFOCTRUP Register

07	06	05	04	03	02	01	00
RSVD						FIFOCTRUP	

Table 518: FIFOCTRUP Register Bits

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

9.14.1.7 FIFO Register

FIFO Read Data

OFFSET: 0x7F

Read this register for FIFO data.

Table 519: FIFO Register

07	06	05	04	03	02	01	00
FIFO							

Table 520: FIFO Register Bits

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

10. PDM/I2S Module

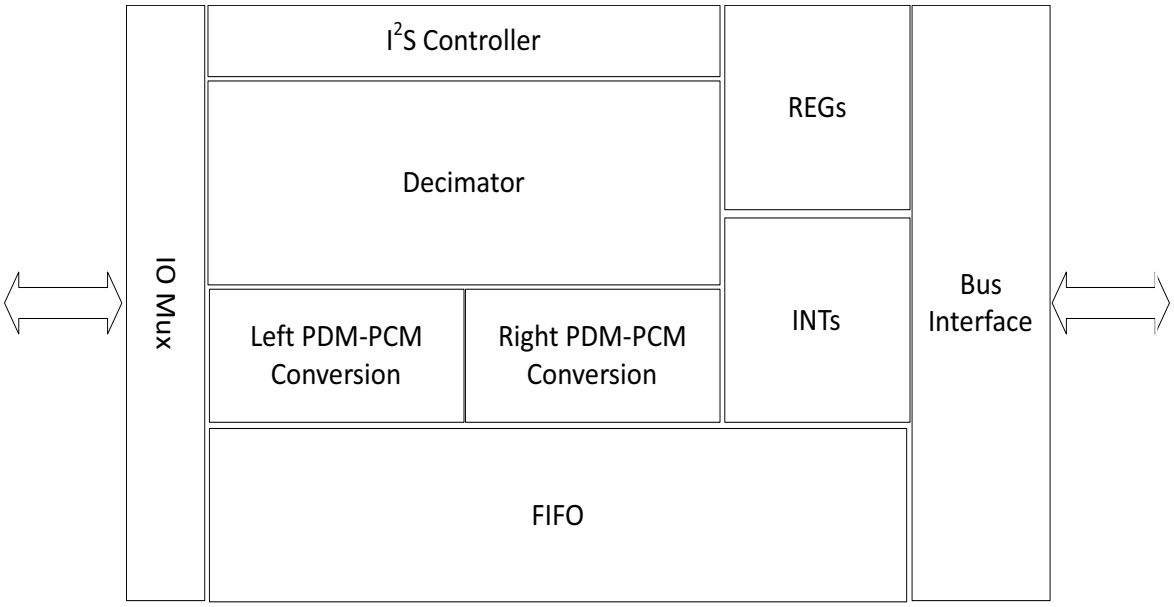


Figure 55. Block Diagram for PDM Module

10.1 Features

The PDM module provides support for low power Pulse-Density Modulated (PDM) to Pulse-Code Modulated (PCM) conversion and optional I2S 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 8/16kHz. 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 MCU CPU via the AHB slave interface.

The PDM controller includes the following features:

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

10.2 Functional Overview

The Apollo3 Blue MCU 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

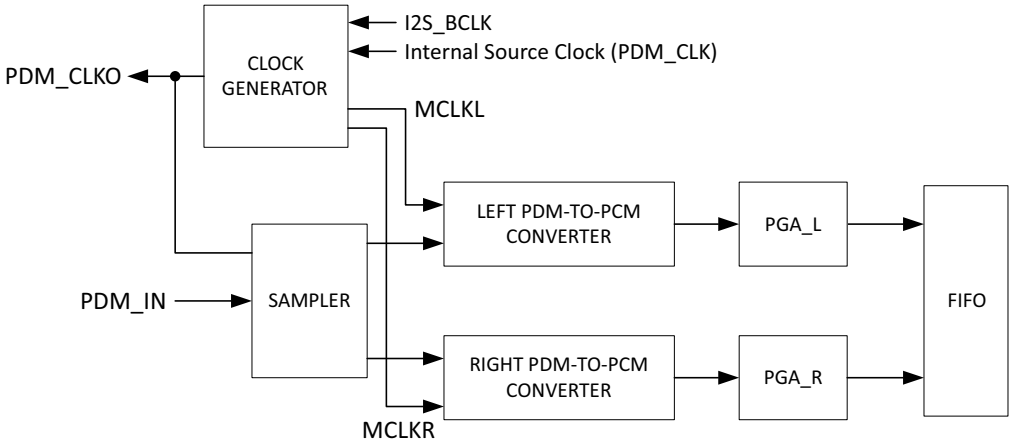
frequency PDM clock is generated to the microphone (requires digital microphone that supports low power operation). Once a keyword is detected, the MCU generates a wake event to enter normal mode. In normal mode, higher PDM frequencies are supported to process audio/voice as needed for voice recording, voice calls, etc.

10.2.1 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 56.

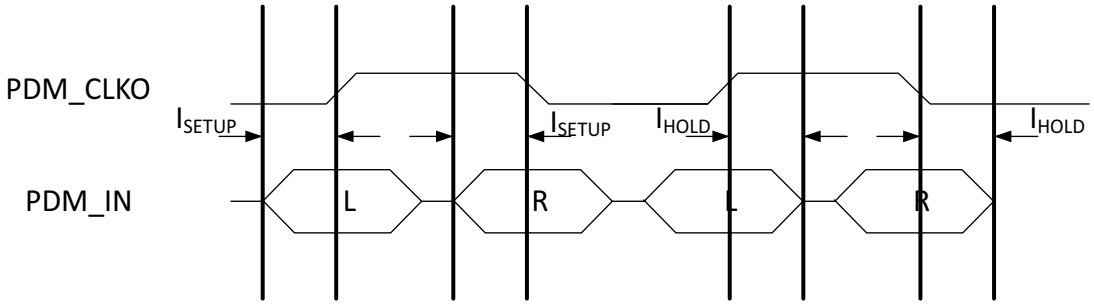
Figure 56. Stereo PDM to PCM Conversion Path



10.2.2 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 57 below.

Figure 57. 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 521 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

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 521: PDM Clock Output Reference Table

F_S (kHz)	Duty Cycle (%)	F_{PDM_CLKO}(kHz)	OSR	MCLKDIV	SYNC_ 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)
16	50	3072	192	MCKDIV4	96	12.288MHz (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.

10.2.3 Clock Switching

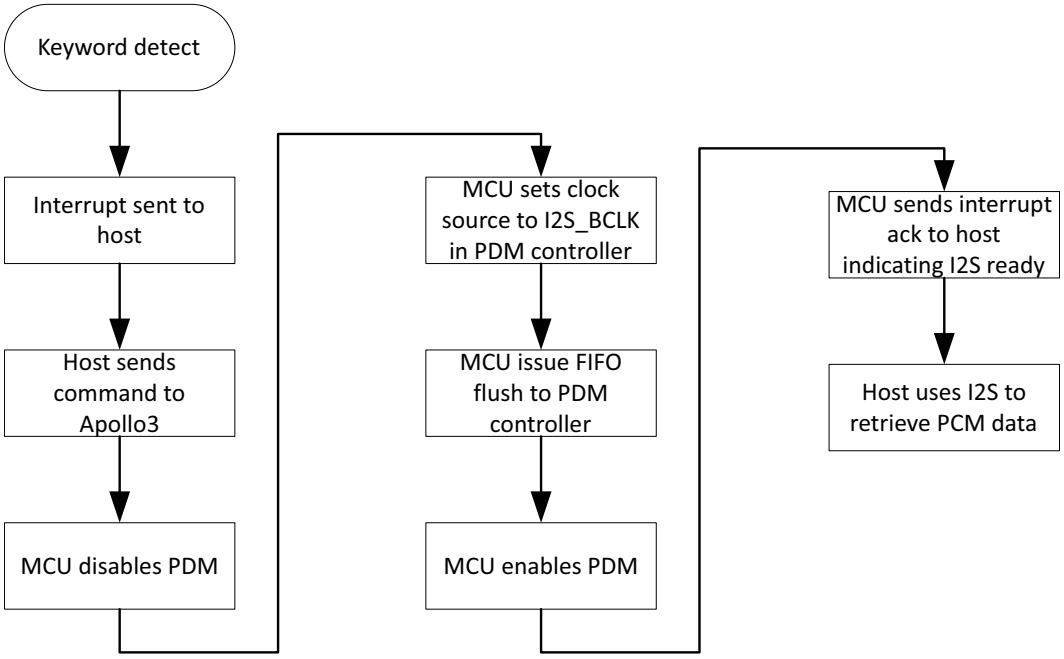
The Apollo3 Blue MCU supports dual-mode clock sourcing for PDM microphone operation. The first mode is clock sourcing from the MCU directly (via divided down HFRC reference). The second mode is clock sourcing from an external host via the I2S_BCLK.

The scenario for switching clock sources is if a higher accuracy clock is required based on the audio sampling requirements. The MCU 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 Apollo3 Blue MCU is used for voice/keyword detect, upon detection, the Apollo3 Blue MCU can generate notification to the external host. The external host can then send a command to the Apollo3 Blue MCU 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 58. PDM Clock Source Switching Flow



10.2.4 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 522 below. 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 522: 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		

Table 522: PDM Operating Modes and Data Formats

Mode	CHSET	PCMPACK	LRSWAP	31 - FIFO Data Format - 0		MCLKL	MCLKR
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	En	En
				0000	R0		
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.

10.2.5 FIFO Control and Interrupts

The PCM data is retrieved from the PDM module through a 32-word FIFO, read at the FRD Register. The number of words currently in the FIFO (0 to 32) 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 32. 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.

10.2.6 Digital Volume Gain

The PDM controller supports digital volume control with a range from -6 dB to +40.5 dB in steps of 1.5 dB. It is programmed by register PGA_L and PGA_R for both left and right channels.

Port Name	Default	Description
PGA_L[4:0]	0000	Left Channel PGA Gain: +1.5dB/step, -6dB to +40.5dB 00000 = -6 dB 00001 = -4.5 dB \u2026 11110 = +39 dB 11111 = +40.5 dB

Port Name	Default	Description
PGA_R[4:0]	0000	Right Channel PGA Gain: +1.5dB/step, -6dB to +40.5dB 00000 = -6 dB 00001 = -4.5 dB \u2026 11110 = +39 dB 11111 = +40.5 dB

10.2.7 Low Pass Filter (LPF)

The controller's internal low pass filters attenuate the out-of-band noise at predefined bandwidth and corners.

Table 524: 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

10.2.8 High Pass Filter

The filter response for high pass filter is characterized as:

$$H(Z) = (1 - Z^{1}) / [1 - (1 - 2^{-HPGAIN}) Z^{-1}]$$

In default mode, HPGAIN = 1011, so the high pass filter can be formulated by the polynomial:

$$H(Z) = (1 - Z^{1}) / [1 - 0.99951Z^{-1}]$$

10.3 I2S Slave Interface

The PDM controller supports an optional I^2S 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 I^2S slave mode, the MSB of I2S_DAT PCM data is available on the second rising edge of I2S_BCLK following an I2S_WDCLK 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_WDCLK is always 32 clock cycles/phase.

Figure 59. I2S Interface Data Format Timing

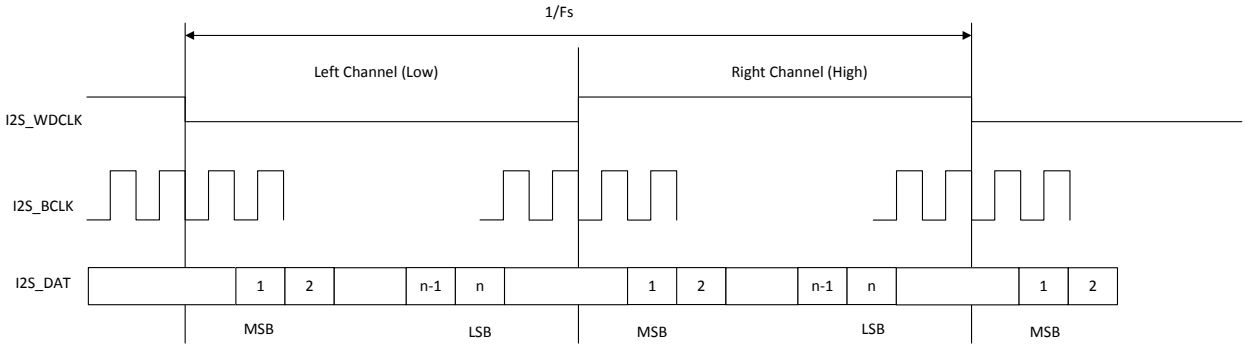
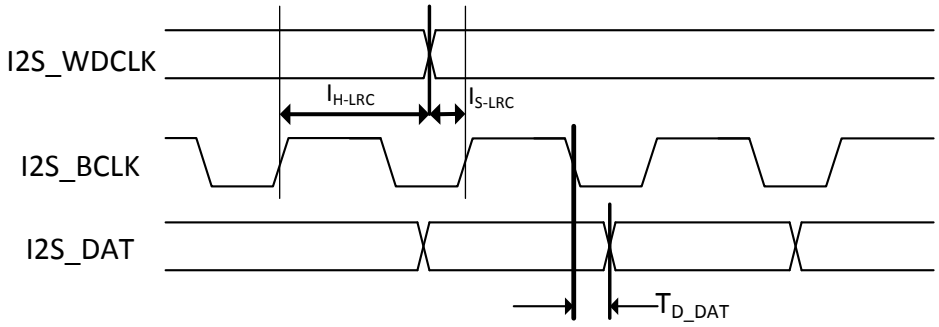


Figure 60. I2S Interface Setup and Hold Timing Diagram



10.4 PDM Registers

PDM Audio

INSTANCE 0 BASE ADDRESS:0x50011000

10.4.1 Register Memory Map

Table 525: PDM Register Map

Address(s)	Register Name	Description
0x50011000	PCFG	PDM Configuration Register
0x50011004	VCFG	Voice Configuration Register
0x50011008	VOICESTAT	Voice Status Register
0x5001100C	FIFOREAD	FIFO Read
0x50011010	FIFOFLUSH	FIFO Flush
0x50011014	FIFOTHR	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
0x50011240	DMATRIGEN	DMA Trigger Enable Register
0x50011244	DMATRIGSTAT	DMA Trigger Status Register
0x50011280	DMACFG	DMA Configuration Register
0x50011288	DMATOTCOUNT	DMA Total Transfer Count
0x5001128C	DMATARGADDR	DMA Target Address Register
0x50011290	DMASTAT	DMA Status Register

10.4.2 PDM Registers

10.4.2.1 PCFG Register

PDM Configuration Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x50011000

PDM Configuration Register

Table 526: 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	PDMCOREEN

Table 527: 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).

Table 527: PCFG Register Bits

Bit	Name	Reset	RW	Description
30:26	PGARIGHT	0x0	RW	Right channel PGA gain. P405DB = 0x1F - 40.5 db gain. P390DB = 0x1E - 39.0 db gain. P375DB = 0x1D - 37.5 db gain. P360DB = 0x1C - 36.0 db gain. P345DB = 0x1B - 34.5 db gain. P330DB = 0x1A - 33.0 db gain. P315DB = 0x19 - 31.5 db gain. P300DB = 0x18 - 30.0 db gain. P285DB = 0x17 - 28.5 db gain. P270DB = 0x16 - 27.0 db gain. P255DB = 0x15 - 25.5 db gain. P240DB = 0x14 - 24.0 db gain. P225DB = 0x13 - 22.5 db gain. P210DB = 0x12 - 21.0 db gain. P195DB = 0x11 - 19.5 db gain. P180DB = 0x10 - 18.0 db gain. P165DB = 0xF - 16.5 db gain. P150DB = 0xE - 15.0 db gain. P135DB = 0xD - 13.5 db gain. P120DB = 0xC - 12.0 db gain. P105DB = 0xB - 10.5 db gain. P90DB = 0xA - 9.0 db gain. P75DB = 0x9 - 7.5 db gain. P60DB = 0x8 - 6.0 db gain. P45DB = 0x7 - 4.5 db gain. P30DB = 0x6 - 3.0 db gain. P15DB = 0x5 - 1.5 db gain. 0DB = 0x4 - 0.0 db gain. M15DB = 0x3 - -1.5 db gain. M300DB = 0x2 - -3.0 db gain. M45DB = 0x1 - -4.5 db gain. M60DB = 0x0 - -6.0 db gain.

Table 527: PCFG Register Bits

Bit	Name	Reset	RW	Description
25:21	PGALEFT	0x0	RW	Left channel PGA gain. P405DB = 0x1F - 40.5 db gain. P390DB = 0x1E - 39.0 db gain. P375DB = 0x1D - 37.5 db gain. P360DB = 0x1C - 36.0 db gain. P345DB = 0x1B - 34.5 db gain. P330DB = 0x1A - 33.0 db gain. P315DB = 0x19 - 31.5 db gain. P300DB = 0x18 - 30.0 db gain. P285DB = 0x17 - 28.5 db gain. P270DB = 0x16 - 27.0 db gain. P255DB = 0x15 - 25.5 db gain. P240DB = 0x14 - 24.0 db gain. P225DB = 0x13 - 22.5 db gain. P210DB = 0x12 - 21.0 db gain. P195DB = 0x11 - 19.5 db gain. P180DB = 0x10 - 18.0 db gain. P165DB = 0xF - 16.5 db gain. P150DB = 0xE - 15.0 db gain. P135DB = 0xD - 13.5 db gain. P120DB = 0xC - 12.0 db gain. P105DB = 0xB - 10.5 db gain. P90DB = 0xA - 9.0 db gain. P75DB = 0x9 - 7.5 db gain. P60DB = 0x8 - 6.0 db gain. P45DB = 0x7 - 4.5 db gain. P30DB = 0x6 - 3.0 db gain. P15DB = 0x5 - 1.5 db gain. 0DB = 0x4 - 0.0 db gain. M15DB = 0x3 - -1.5 db gain. M300DB = 0x2 - -3.0 db gain. M45DB = 0x1 - -4.5 db gain. M60DB = 0x0 - -6.0 db gain.
20:19	RSVD	0x0	RO	This bitfield 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.

Table 527: PCFG Register Bits

Bit	Name	Reset	RW	Description
0	PDMCOREEN	0x1	RW	Data Streaming Control. EN = 0x1 - Enable Data Streaming. DIS = 0x0 - Disable Data Streaming.

10.4.2.2 VCFG Register

Voice Configuration Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x50011004

Voice Configuration Register

Table 528: 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			PDMCLKEN	RSVD					I2SEN	BCLKINV	RSVD	DMICKDEL	SELAP	RSVD							PCMPACK	RSVD			CHSET	RSVD			

Table 529: 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	PDMCLKEN	0x0	RW	Enable the serial clock. DIS = 0x0 - Disable serial clock. EN = 0x1 - Enable serial clock.

Table 529: VCFG Register Bits

Bit	Name	Reset	RW	Description
25:21	RSVD	0x0	RO	This bitfield is reserved for future use.
20	I2SEN	0x0	RW	I2S interface enable. DIS = 0x0 - Disable I2S interface. EN = 0x1 - Enable I2S interface.
19	BCLKINV	0x0	RW	I2S BCLK input inversion. INV = 0x0 - BCLK inverted. NORM = 0x1 - BCLK not inverted.
18	RSVD	0x0	RO	This bitfield 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. I2S = 0x1 - Clock source from I2S BCLK. INTERNAL = 0x0 - Clock source from internal clock generator.
15:9	RSVD	0x0	RO	This bitfield 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 bitfield is reserved for future use.
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.

10.4.2.3 VOICESTAT Register

Voice Status Register

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x50011008

Voice Status Register

Table 530: VOICESTAT Register

31	30	29	28	27	26	25	24	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 531: VOICESTAT Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	This bitfield is reserved for future use.
5:0	FIFOCNT	0x0	RO	Valid 32-bit entries currently in the FIFO.

10.4.2.4 FIFOREAD Register

FIFO Read

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x5001100C

FIFO Read

Table 532: FIFOREAD Register

31	30	29	28	27	26	25	24	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 533: FIFOREAD Register Bits

Bit	Name	Reset	RW	Description
31:0	FIFOREAD	0x0	RO	FIFO read data.

10.4.2.5 FIFOFLUSH Register

FIFO Flush

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x50011010

FIFO Flush

Table 534: FIFOFLUSH Register

31	30	29	28	27	26	25	24	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 535: FIFOFLUSH Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.
0	FIFOFLUSH	0x0	WO	FIFO FLUSH.

10.4.2.6 FIFOTHR Register

FIFO Threshold

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x50011014

FIFO Threshold

Table 536: 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 537: FIFOTHR Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	This bitfield is reserved for future use.
4:0	FIFOTHR	0x10	RW	FIFO Threshold value. When the FIFO count is equal to, or larger than this value (in words), a THR interrupt is generated (if enabled)

10.4.2.7 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 538: 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																											DERR	DCMP	UNDFL	OVF	THR

Table 539: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4	DERR	0x0	RW	DMA Error receieved
3	DCMP	0x0	RW	DMA completed a transfer
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.

10.4.2.8 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 540: 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																											DERR	DCMP	UNDFL	OVF	THR

Table 541: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4	DERR	0x0	RW	DMA Error receieved
3	DCMP	0x0	RW	DMA completed a transfer
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.

10.4.2.9 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 542: 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																											DERR	DCMP	UNDFL	OVF	THR

Table 543: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4	DERR	0x0	RW	DMA Error receieved
3	DCMP	0x0	RW	DMA completed a transfer
2	UNDFL	0x0	RW	This is the FIFO underflow interrupt.
1	OVF	0x0	RW	This is the FIFO overflow interrupt.

Table 543: INTCLR Register Bits

Bit	Name	Reset	RW	Description
0	THR	0x0	RW	This is the FIFO threshold interrupt.

10.4.2.10INTSET 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 544: 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																											DERR	DCMP	UNDFL	OVF	THR

Table 545: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:5	RSVD	0x0	RO	RESERVED
4	DERR	0x0	RW	DMA Error receieved
3	DCMP	0x0	RW	DMA completed a transfer
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.

10.4.2.11DMATRIGEN Register

DMA Trigger Enable Register

OFFSET: 0x00000240

INSTANCE 0 ADDRESS: 0x50011240

DMA Trigger Enable Register

Table 546: DMATRIGEN Register

31	30	29	28	27	26	25	24	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																														DTHR90	DTHR

Table 547: DMATRIGEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DTHR90	0x0	RW	Trigger DMA at FIFO 90 percent full. This signal is also used internally for AUTOHIP function
0	DTHR	0x0	RW	Trigger DMA upon when FIFO iss filled to level indicated by the FIFO THRESHOLD,at granularity of 16 bytes only

10.4.2.12DMATRIGSTAT Register

DMA Trigger Status Register

OFFSET: 0x00000244

INSTANCE 0 ADDRESS: 0x50011244

DMA Trigger Status Register

Table 548: DMATRIGSTAT Register

31	30	29	28	27	26	25	24	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																														DTHR90STAT	DTHRSTAT

Table 549: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DTHR90STAT	0x0	RO	Triggered DMA from FIFO reaching 90 percent full

Table 549: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
0	DTHRSTAT	0x0	RO	Triggered DMA from FIFO reaching threshold

10.4.2.13DMACFG Register

DMA Configuration Register

OFFSET: 0x00000280

INSTANCE 0 ADDRESS: 0x50011280

DMA Configuration Register

Table 550: DMACFG Register

31	30	29	28	27	26	25	24	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																					DPWROFF	DAUTOHIP	DMAPRI	RSVD					DMADIR	RSVD	DMAEN

Table 551: DMACFG Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED.
10	DPWROFF	0x0	RW	Power Off the ADC System upon DMACPL.
9	DAUTOHIP	0x0	RW	Raise priority to high on fifo full, and DMAPRI set to low
8	DMAPRI	0x0	RW	Sets the Priority of the DMA request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
7:3	RSVD	0x0	RO	RESERVED.
2	DMADIR	0x0	RO	Direction P2M = 0x0 - Peripheral to Memory (SRAM) transaction. THe PDM module will only DMA to memory. M2P = 0x1 - Memory to Peripheral transaction. Not available for PDM module
1	RSVD	0x0	RO	RESERVED.
0	DMAEN	0x0	RW	DMA Enable DIS = 0x0 - Disable DMA Function EN = 0x1 - Enable DMA Function

10.4.2.14DMATOTCOUNT Register

DMA Total Transfer Count

OFFSET: 0x00000288

INSTANCE 0 ADDRESS: 0x50011288

DMA Total Transfer Count

Table 552: DMATOTCOUNT Register

31	30	29	28	27	26	25	24	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												TOTCOUNT																			

Table 553: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	RESERVED.
19:0	TOTCOUNT	0x0	RW	Total Transfer Count. The transfer count must be a multiple of the THR setting to avoid DMA overruns.

10.4.2.15DMATARGADDR Register

DMA Target Address Register

OFFSET: 0x0000028C

INSTANCE 0 ADDRESS: 0x5001128C

DMA Target Address Register

Table 554: DMATARGADDR Register

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

Table 555: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
31:20	UTARGADDR	0x100	RO	SRAM Target

Table 555: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
19:0	LTARGADDR	0x0	RW	DMA Target Address. This register is not updated with the current address of the DMA, but will remain static with the original address during the DMA transfer.

10.4.2.16DMASTAT Register

DMA Status Register

OFFSET: 0x00000290

INSTANCE 0 ADDRESS: 0x50011290

DMA Status Register

Table 556: DMASTAT Register

31	30	29	28	27	26	25	24	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																													DMAERR	DMACPL	DMATIP

Table 557: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	DMAERR	0x0	RW	DMA Error
1	DMACPL	0x0	RW	DMA Transfer Complete
0	DMATIP	0x0	RW	DMA Transfer In Progress

11. GPIO and Pad Configuration Module

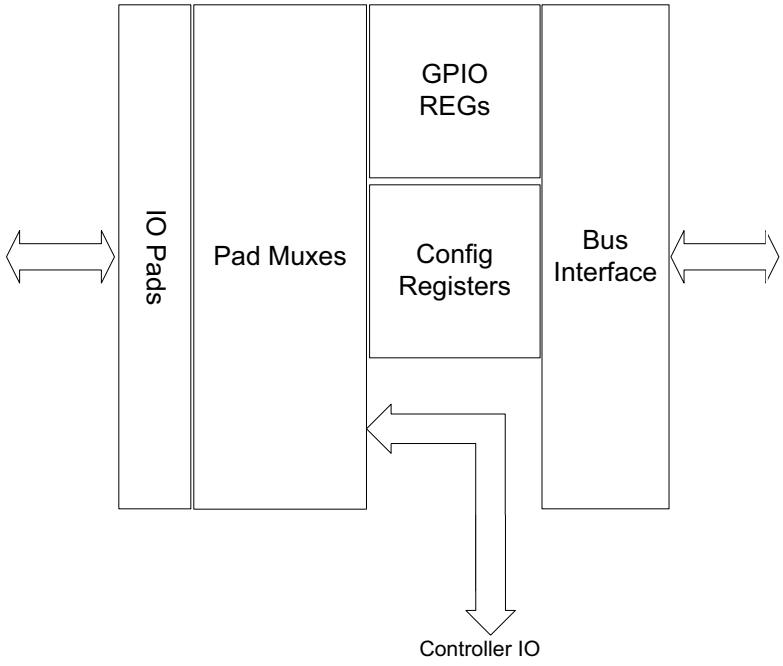


Figure 61. Block diagram for the General Purpose I/O (GPIO) Module

11.1 Functional Overview

The General Purpose I/O and Pad Configuration (GPIO) Module, shown in Figure 61, 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.

11.2 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 Table 559. Functions are grouped by module, with the color coding shown in Table 561. This table also defines the pad type for each configuration. The Special pad types are defined in Table 562. 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 Table 559.

The REG_GPIO_PADREGy_PADnSTRNG bit and the REG_GPIO_ALTPADCFGy_PADn_DS1 bit 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 558.

Table 558: 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

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 3 and 36 have selectable high side power switch transistors to provide ~1 Ω switches to VDDH. Pads 37 and 41 have a selectable low side power switch transistors to provide ~1 Ω switches to VSS. The high side power switches are enabled by setting the REG_GPIO_PADREGF_PAD3PWRUP or REG_GPIO_PADREGK_PAD36PWRUP bits, and the low side switch is enabled by setting the REG_GPIO_PADREGB_PAD37PWRDN or REG_GPIO_PADREGB_PAD41PWRDN 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 563.

Table 559: Apollo3 Blue MCU Pad Function Mapping

Pad	PADnFNCSEL								CSP PKG
	0	1	2	3	4	5	6	7	
0	SLSCL	SLSCK	CLKOUT	GPIO00		MSPI4		NCE0	X
1	SLSDAWIR3	SLMOSI	UART0TX	GPIO01		MSPI5		NCE1	X
2	UART1RX	SLMISO	UART0RX	GPIO02		MSPI6		NCE2	X
3	UA0RTS	SLnCE	NCE3	GPIO03		MSPI7	TRIG1	I2SWCLK	X
4	UA0CTS	SLINT	NCE4	GPIO04		UART1RX	CT17	MSPI2	X
5	M0SCL	M0SCK	UA0RTS	GPIO05		-		CT8	X
6	M0SDAWIR3	M0MISO	UA0CTS	GPIO06		CT10		I2SDAT	X
7	NCE7	M0MOSI	CLKOUT	GPIO07	TRIG0	UART0TX		CT19	X
8	M1SCL	M1SCK	NCE8	GPIO08	SCCCLK		UART1TX		X
9	M1SDAWIR3	M1MISO	NCE9	GPIO09	SCCIO		UART1RX		X
10	UART1TX	M1MOSI	NCE10	GPIO10	PDMCLK	UA1RTS			X
11	ADCSE2	NCE11	CT31	GPIO11	SLINT	UA1CTS	UART0RX	PDMDATA	X
12	ADCD0NSE9	NCE12	CT0	GPIO12	SLnCE	PDMCLK	UA0CTS	UART1TX	X
13	ADCD0PSE8	NCE13	CT2	GPIO13	I2SBCLK	-	UA0RTS	UART1RX	X
14	ADCD1P	NCE14	UART1TX	GPIO14	PDMCLK	-	SWDCK	32KHzXT	X
15	ADCD1N	NCE15	UART1RX	GPIO15	PDMDATA	-	SWDIO	SWO	X
16	ADCSE0	NCE16	TRIG0	GPIO16	SCCRST	CMPIN0	UART0TX	UA1RTS	X
17	CMPRF1	NCE17	TRIG1	GPIO17	SCCCLK		UART0RX	UA1CTS	X
18	CMPIN1	NCE18	CT4	GPIO18	UA0RTS	-	UART1TX	SCCIO	X
19	CMPRF0	NCE19	CT6	GPIO19	SCCCLK	-	UART1RX	I2SBCLK	X
20	SWDCK	NCE20		GPIO20	UART0TX	UART1TX	I2SBCLK	UA1RTS	X
21	SWDIO	NCE21		GPIO21	UART0RX	UART1RX	SCCRST	UA1CTS	X
22	UART0TX	NCE22	CT12	GPIO22	PDMCLK	-	MSPI0	SWO	X
23	UART0RX	NCE23	CT14	GPIO23	I2SWCLK	CMPOUT	MSPI3	-	X
24	UART1TX	NCE24	MSPI8	GPIO24	UA0CTS	CT21	32KHzXT	SWO	X
25	UART1RX	NCE25	CT1	GPIO25	M2SDAWIR3	M2MISO			X
26	-	NCE26	CT3	GPIO26	SCCRST	MSPI1	UART0TX	UA1CTS	X
27	UART0RX	NCE27	CT5	GPIO27	M2SCL	M2SCK			X
28	I2SWCLK	NCE28	CT7	GPIO28		M2MOSI	UART0TX		X
29	ADCSE1	NCE29	CT9	GPIO29	UA0CTS	UA1CTS	UART0RX	PDMDATA	X
30	-	NCE30	CT11	GPIO30	UART0TX	UA1RTS	BLEIF_SCK	I2SDAT	
31	ADCSE3	NCE31	CT13	GPIO31	UART0RX	SCCCLK	BLEIF_MISO	UA1RTS	
32	ADCSE4	NCE32	CT15	GPIO32	SCCIO	-	BLEIF_MOSI	UA1CTS	
33	ADCSE5	NCE33	32KHzXT	GPIO33	BLEIF_CSN	UA0CTS	CT23	SWO	
34	ADCSE6	NCE34	UA1RTS	GPIO34	CMPRF2	UA0RTS	UART0RX	PDMDATA	
35	ADCSE7	NCE35	UART1TX	GPIO35	I2SDAT	CT27	UA0RTS	BLEIF_STATUS	
36	TRIG1	NCE36	UART1RX	GPIO36	32KHzXT	UA1CTS	UA0CTS	PDMDATA	
37	TRIG2	NCE37	UA0RTS	GPIO37	SCCIO	UART1TX	PDMCLK	CT29	
38	TRIG3	NCE38	UA0CTS	GPIO38		M3MOSI	UART1RX		
39	UART0TX	UART1TX	CT25	GPIO39	M4SCL	M4SCK			X
40	UART0RX	UART1RX	TRIG0	GPIO40	M4SDAWIR3	M4MISO			X
41	NCE41	BLEIF_IRQ	SWO	GPIO41	I2SWCLK	UA1RTS	UART0TX	UA0RTS	X
42	UART1TX	NCE42	CT16	GPIO42	M3SCL	M3SCK			
43	UART1RX	NCE43	CT18	GPIO43	M3SDAWIR3	M3MISO			
44	UA1RTS	NCE44	CT20	GPIO44		M4MOSI	UART0TX		X
45	UA1CTS	NCE45	CT22	GPIO45	I2SDAT	PDMDATA	UART0RX	SWO	
46	I2SBCLK	NCE46	CT24	GPIO46	SCCRST	PDMCLK	UART1TX	SWO	
47	32KHzXT	NCE47	CT26	GPIO47		M5MOSI	UART1RX		X
48	UART0TX	NCE48	CT28	GPIO48	M5SCL	M5SCK			X
49	UART0RX	NCE49	CT30	GPIO49	M5SDAWIR3	M5MISO			X

Table 561: 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
	MSPI Signals	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	Global IOM/MSPI	Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	SCARD	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
	Clock Output Signals	Push-pull Output
	Debug Signals	Input Special or Push-pull output, as indicated by [I], [S] or [O] respectively
	High-side power switch	Pads 3 and 36 have selectable high side power switch transistors to provide ~1 Ω switches to VDDH.
	Low-side power switch	Pads 37 and 41 have selectable low side power switch transistors to provide ~1 Ω switches to VSS.

Table 562: Special Pad Types

Pad	PADnFNCSEL	Name	Pad Type
0	5	MSPI4	Bidirectional Tri-state

Table 562: Special Pad Types

Pad	PADnFNCSEL	Name	Pad Type
1	0	SLSDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
1	5	MSPI5	Bidirectional Tri-state
2	5	MSPI6	Bidirectional Tri-state
3	5	MSPI7	Bidirectional Tri-state
4	7	MSPI2	Bidirectional Tri-state
5	0	M0SCL	Open Drain
6	0	M0SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
8	0	M1SCL	Open Drain
9	0	M1SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
9	4	SCCIO	Bidirectional Tri-state
15	6	SWDIO	Bidirectional Tri-state
18	7	SCCIO	Bidirectional Tri-state
21	0	SWDIO	Bidirectional Tri-state
22	6	MSPI0	Bidirectional Tri-state
23	6	MSPI3	Bidirectional Tri-state
25	4	M2SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
26	5	MSPI1	Bidirectional Tri-state
27	4	M2SCL	Open Drain
32	4	SCCIO	Bidirectional Tri-state
37	4	SCCIO	Bidirectional Tri-state
39	4	M4SCL	Open Drain
40	4	M4SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
42	4	M3SCL	Open Drain
43	4	M3SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state
48	4	M5SCL	Open Drain
49	4	M5SDAWIR3	Bidirectional Open Drain Bidirectional Tri-state

Table 563: I^2C Pullup Resistor Selection

RSEL[1:0]	Pullup Resistor
00	1.5 kΩ
01	6 kΩ
10	12 kΩ
11	24 kΩ

Table 564: NCE Encoding Table

	GPIOxOUTCFG			
	0	1	2	3
NCE0	IOM3.2	IOM4.2	IOM5.2	IOM1.3
NCE1	IOM0.2	IOM1.2	IOM2.2	MSPI.0
NCE2	IOM3.3	IOM4.3	IOM5.3	IOM2.1
NCE3	IOM3.0	IOM4.0	IOM5.0	IOM2.0
NCE4	IOM3.1	IOM4.1	IOM5.1	IOM1.1
-	-	-	-	-
-	-	-	-	-
NCE7	IOM3.1	IOM4.1	IOM5.1	MSPI.0
NCE8	IOM3.0	IOM4.0	IOM5.0	IOM0.0
NCE9	IOM3.3	IOM4.3	IOM5.3	IOM2.3
NCE10	IOM3.2	IOM4.2	IOM5.2	MSPI.0
NCE11	IOM0.0	IOM1.0	IOM2.0	IOM3.0
NCE12	IOM3.0	IOM4.0	IOM5.0	MSPI.1
NCE13	IOM3.1	IOM4.1	IOM5.1	IOM0.1
NCE14	IOM0.2	IOM1.2	IOM2.2	IOM4.2
NCE15	IOM0.3	IOM1.3	IOM2.3	MSPI.0
NCE16	IOM0.0	IOM1.0	IOM2.0	IOM5.0
NCE17	IOM0.1	IOM1.1	IOM2.1	IOM4.1
NCE18	IOM0.2	IOM1.2	IOM2.2	IOM3.2
NCE19	IOM0.3	IOM1.3	IOM3.3	MSPI.0
NCE20	IOM3.1	IOM4.1	IOM5.1	IOM2.1
NCE21	IOM3.2	IOM4.2	IOM5.2	IOM2.2
NCE22	IOM3.3	IOM4.3	IOM5.3	IOM0.3
NCE23	IOM0.0	IOM1.0	IOM2.0	IOM4.0
NCE24	IOM0.1	IOM1.1	IOM2.1	IOM5.1
NCE25	IOM3.2	IOM4.2	IOM5.2	IOM0.2
NCE26	IOM3.3	IOM4.3	IOM5.3	IOM1.3
NCE27	IOM3.0	IOM4.0	IOM5.0	IOM1.0
NCE28	IOM3.1	IOM4.1	IOM5.1	MSPI.0
NCE29	IOM3.2	IOM4.2	IOM5.2	IOM1.2
NCE30	IOM3.3	IOM4.3	IOM5.3	IOM0.3
NCE31	IOM0.0	IOM1.0	IOM2.0	IOM4.0
NCE32	IOM0.1	IOM1.1	IOM2.1	MSPI.1
NCE33	IOM0.2	IOM1.2	IOM2.2	IOM5.2
NCE34	IOM0.3	IOM1.3	IOM2.3	IOM3.3
NCE35	IOM0.0	IOM1.0	IOM2.0	IOM3.0
NCE36	IOM3.1	IOM4.1	IOM5.1	MSPI.1
NCE37	IOM3.2	IOM4.2	IOM5.2	IOM0.2
NCE38	IOM0.3	IOM1.3	IOM2.3	IOM5.3
-	-	-	-	-
-	-	-	-	-
NCE41	IOM0.1	IOM1.1	IOM2.1	MSPI.1
NCE42	IOM0.0	IOM1.0	IOM2.0	IOM5.0
NCE43	IOM0.1	IOM1.1	IOM2.1	MSPI.1
NCE44	IOM0.2	IOM1.2	IOM2.2	IOM5.2
NCE45	IOM3.3	IOM4.3	IOM5.3	IOM1.3
NCE46	IOM3.0	IOM4.0	IOM5.0	MSPI.1
NCE47	IOM0.1	IOM1.1	IOM2.1	IOM3.1
NCE48	IOM0.2	IOM1.2	IOM2.2	IOM3.2
NCE49	IOM0.3	IOM1.3	IOM2.3	IOM4.3

11.3 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.

11.3.1 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. 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.

For Apollo3 Blue MCU, if the PADxFNCSEL is set to a “NCE” signal group, the additional NCE encoding is applied as shown in Table 564 based on GPIOxOUTCFG and the pad is automatically configured for push-pull output. If the PADxFNCSEL is set to a “MSPI” signal group, the additional MSPI encoding is applied as shown in Section 11.5.2.

11.3.2 Reading from a GPIO Pad

All GPIO inputs are readable at all times unless the interrupt configuration (determined by REG_GPIO_CFGy_GPIOnINCFG and REG_GPIO_CFGy_GPIOnINTD) is set to “disabled”, 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 if the interrupt configuration is set to “disabled”.

11.3.3 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 or REG_GPIO_WTSB 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 or REG_GPIO_WTCB 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. ENA and ENB 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.

11.3.4 GPIO Interrupts

Each GPIO pad can be configured to generate an interrupt on a high-to-low transition or a low-to-high or either transition, as selected by the REG_GPIO_CFGy_GPIOnINTD and REG_GPIO_CFGy_GPIOnINCFG bits. 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 GPIOA_IER, GPIOA_ISR, GPIOA_WCR and GPIOB_WSR for GPIO pads 0 to 31, and Registers GPIOB_IER, GPIOB_ISR, GPIOB_WCR and GPIOB_WSR for GPIO pads 32 to 49.

Below describes the interrupt configuration.

Table 565: Interrupt Configuration

INCFG	INTD	Interrupt
0	0	Low -> high transition
0	1	High -> low transition
1	0	Disabled
1	1	Either low -> high or high -> low transition

11.4 Pad Connection Summary

Figure 62 shows the detailed implementation of each pad. Each element will be described in detail.

11.4.1 Output Selection

There is a multiplexer which selects the module signal to be driven to the output based on REG_GPIO_PADREGy_PADnFNCSEL field. This implements the multiplexing shown in Table 559 for output pads. For all pads, a PADnFNCSEL value of 0x3 selects the value in the corresponding GPIO_WTy register bit.

Certain functional groups, Timer (CT), NCE and MSPI in particular, have additional pre-muxing configuration as noted.

NOTE: In order to achieve high data rates on the MSPI interface, some of the normal GPIO muxing has been bypassed. When MSPI-related pads (24, 22, 26, 4, 23, 0, 1, 2, 3) are used for non-MSPI functions, the MSPI module must be powered down or the related bit in the MSPI PADOUTEN register must be disabled to ensure that the MSPI does not drive data to the pad (which is logically OR'd with the data from the GPIO functional mux select).

11.4.2 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 562 and Table 563.

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.

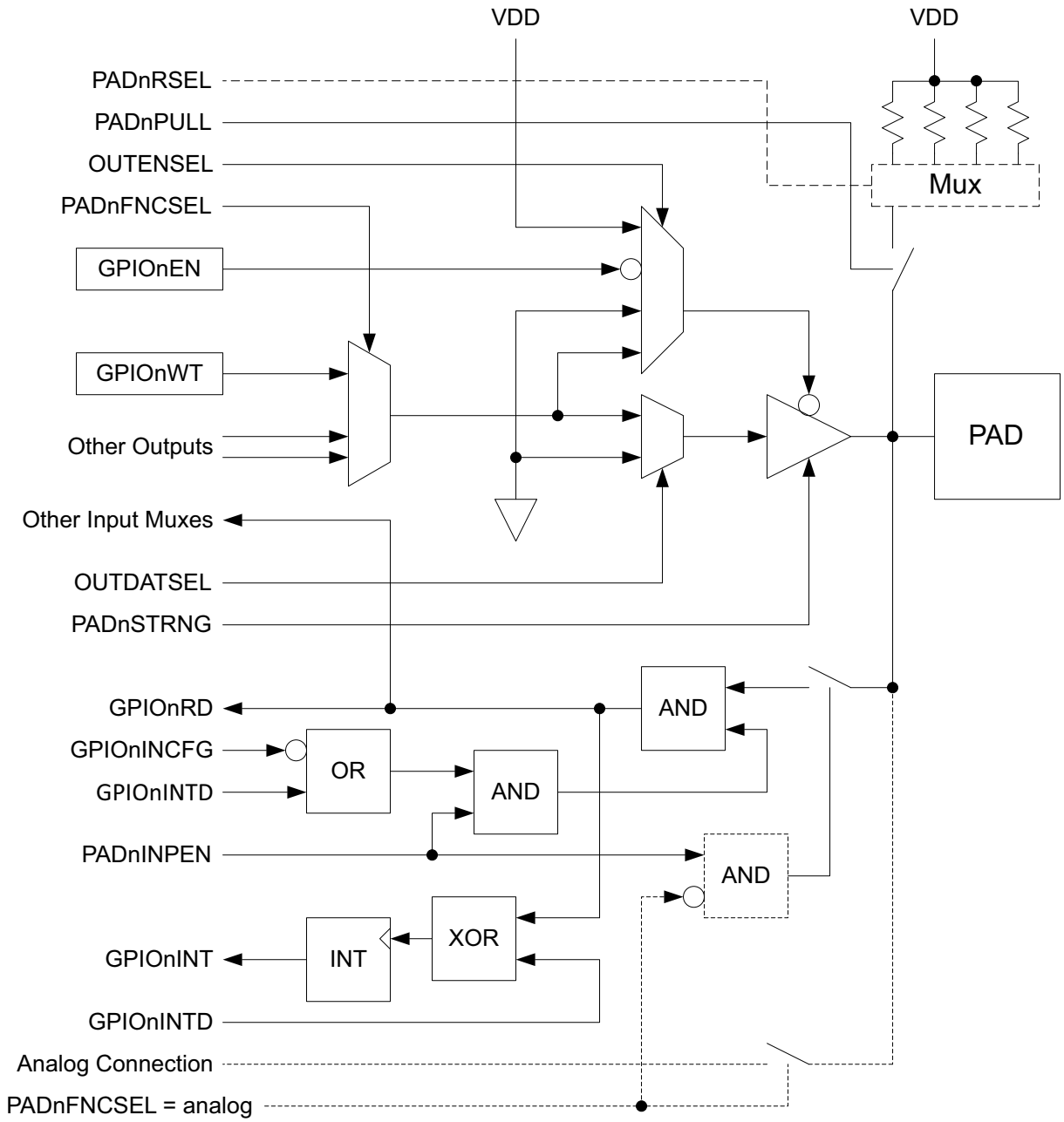


Figure 62. Pad Connection Details

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 GPIOEN bit. 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 11.2 on page 379.

11.4.3 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 Table 559. In addition, the pad input can always be read from the GPIORD register unless the pad is configured as a GPIO (PADnFNCSEL = 0x3) and GPIOnINCFG is high, which will force the GPIORD 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 corresponding 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.

11.4.4 Pull-up Control

If PADnPULL is high, a pullup 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 (pads 0, 1, 5, 6, 8, 9, 25, 27, 39, 40, 42, 43 48 and 49) contain the additional circuitry shown with the dashed lines. In this case four different pullup resistors are selected by the PADnRSEL field.

11.4.5 Analog Pad Configuration

Pads which may have analog connections (pads 11-19, 29 and 31-35) include the circuitry shown with the dotted lines. If the pad is configured in analog mode (reference the analog input function selections in Table 1: Pin List and Function Table, 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.

11.5 Module-specific Pad Configuration

The following sections describe in detail how to configure the pads for each module function.

11.5.1 Implementing IO Master Connections

The six IO Master modules must be correctly connected to the appropriate pads in order to operate.

11.5.1.1 IO Master 0 I^2C Connection

I^2C 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 566. The PAD5INPEN and PAD6INPEN bits must be set. If the internal I^2C pullup resistors are to be used, PAD5PULL and PAD6PULL should be set, and the PAD5RSEL

and PAD6RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD5PULL and PAD6PULL should be cleared.

Table 566: IO Master 0 I2C Configuration

Field	Value
PAD5FNCSEL	0
PAD6FNCSEL	0

11.5.1.2 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 567. The PAD8INPEN and PAD9INPEN bits must be set. If the internal I2C pullup resistors are to be used, PAD8PULL and PAD9PULL should be set, and the PAD8RSEL and PAD9RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD8PULL and PAD9PULL should be cleared.

Table 567: IO Master 1 I2C Configuration

Field	Value
PAD8FNCSEL	0
PAD9FNCSEL	0

11.5.1.3 IO Master 2 I2C Connection

I2C mode of IO Master 2 uses pad 27 as SCL and pad 25 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 568. The PAD27INPEN and PAD25INPEN bits must be set. If the internal I2C pullup resistors are to be used, PAD27PULL and PAD25PULL should be set, and the PAD27RSEL and PAD25RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD27PULL and PAD25PULL should be cleared.

Table 568: IO Master 2 I2C Configuration

Field	Value
PAD27FNCSEL	4
PAD25FNCSEL	4

11.5.1.4 IO Master 3 I2C Connection

I2C 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 569. The PAD42INPEN and PAD43INPEN bits must be set. If the internal I2C pullup resistors are to be used, PAD42PULL and PAD43PULL should be set, and the

PAD42RSEL and PAD43RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD42PULL and PAD43PULL should be cleared.

Table 569: IO Master 3 I2C Configuration

Field	Value
PAD42FNCSEL	4
PAD43FNCSEL	4

11.5.1.5 IO Master 4 I2C Connection

I2C 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 570. The PAD39INPEN and PAD40INPEN bits must be set. If the internal I2C pullup resistors are to be used, PAD39PULL and PAD40PULL should be set, and the PAD39RSEL and PAD40RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD39PULL and PAD40PULL should be cleared.

Table 570: IO Master I2C Configuration

Field	Value
PAD39FNCSEL	4
PAD40FNCSEL	4

11.5.1.6 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 571. The PAD48INPEN and PAD49INPEN bits must be set. If the internal I2C pullup resistors are to be used, PAD48PULL and PAD49PULL should be set, and the PAD48RSEL and PAD49RSEL fields should be set to select the desired pullup resistor size as shown in Table 563. If external pullup resistors are used, PAD48PULL and PAD49PULL should be cleared.

Table 571: IO Master 5 I2C Configuration

Field	Value
PAD48FNCSEL	4
PAD49FNCSEL	4

11.5.1.7 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 572. The PAD5INPEN and PAD6INPEN bits must be set. PAD5PULL, PAD6PULL and PAD7PULL should be cleared.

Table 572: IO Master 0 4-wire SPI Configuration

Field	Value
PAD5FNCSEL	1
PAD6FNCSEL	1
PAD7FNCSEL	1

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.8 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 573. The PAD8INPEN and PAD9INPEN bits must be set. PAD8PULL, PAD9PULL and PAD10PULL should be cleared.

Table 573: 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 four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.9 IO Master 2 4-wire SPI Connection

Four-wire SPI mode of IO Master 2 uses pad 27 as SCK, pad 28 as MOSI and pad 25 as MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 574. The PAD27INPEN and PAD28INPEN bits must be set. PAD27PULL, PAD28PULL and PAD25PULL should be cleared.

Table 574: IO Master 2 4-wire SPI Configuration

Field	Value
PAD27FNCSEL	5
PAD28FNCSEL	5
PAD25FNCSEL	5

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.10IO 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 575. The PAD42INPEN and PAD38INPEN bits must be set. PAD38PULL, PAD42PULL and PAD43PULL should be cleared.

Table 575: 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 four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.11IO 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 576. The PAD39INPEN and PAD44INPEN bits must be set. PAD39PULL, PAD40PULL and PAD44PULL should be cleared.

Table 576: 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 four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.12IO 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 577. The PAD48INPEN and PAD47INPEN bits must be set. PAD48PULL, PAD49PULL and PAD47PULL should be cleared.

Table 577: 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 four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.13IO Master 0 3-wire SPI Connection

Three-wire SPI mode of IO Master 0 uses pad 5 as SCK and pad 6 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 578. The PAD5INPEN and PAD6INPEN bits must be set. PAD5PULL and PAD6PULL should be cleared. Pad 7 may be used for other functions.

Table 578: IO Master 0 3-wire SPI Configuration

Field	Value
PAD5FNCSEL	1
PAD6FNCSEL	0

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.14IO Master 1 3-wire SPI Connection

Three-wire SPI mode of IO Master 1 uses pad 8 as SCK and pad 9 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 579. The PAD8INPEN and PAD9INPEN bits must be set. PAD8PULL and PAD9PULL should be cleared. Pad 10 may be used for other functions.

Table 579: IO Master 1 3-wire SPI Configuration

Field	Value
PAD8FNCSEL	1
PAD9FNCSEL	0

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.15IO Master 2 3-wire SPI Connection

Three-wire SPI mode of IO Master 2 uses pad 27 as SCK and pad 25 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 580. The PAD27INPEN and PAD25INPEN bits must be set. PAD27PULL and PAD25PULL should be cleared. Pad 28 may be used for other functions.

Table 580: IO Master 2 3-wire SPI Configuration

Field	Value
PAD27FNCSEL	5
PAD25FNCSEL	4

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.16IO Master 3 3-wire SPI Connection

Three-wire SPI mode of IO Master 3 uses pad 42 as SCK and pad 43as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 581. The PAD42INPENand PAD43INPENbits must be set. PAD42PULL and PAD43PULLshould be cleared. Pad 38 may be used for other functions.

Table 581: IO Master 3 3-wire SPI Configuration

Field	Value
PAD42FNCSEL	5
PAD43FNCSEL	4

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.17IO Master 4 3-wire SPI Connection

Three-wire SPI mode of IO Master 4 uses pad 39 as SCK and pad 40 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 582. The PAD39INPEN and PAD40INPEN bits must be set. PAD39PULL and PAD40PULL should be cleared. Pad 44 may be used for other functions.

Table 582: IO Master 4 3-wire SPI Configuration

Field	Value
PAD39FNCSEL	5
PAD40FNCSEL	4

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.18IO Master 5 3-wire SPI Connection

Three-wire SPI mode of IO Master 5 uses pad 48 as SCK and pad 49 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 583. The PAD48INPEN and PAD49INPEN bits must be set. PAD48PULL and PAD49PULL should be cleared. Pad 47 may be used for other functions.

Table 583: IO Master 3-wire SPI Configuration

Field	Value
PAD48FNCSEL	5
PAD49FNCSEL	4

A variety of pads may be used for up to four nCE signals to select up to four separate slaves. The nCE signals are pre-muxed into a signal group called NCE. The muxing configuration is shown in Table 564. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

11.5.1.19SPI 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).

11.5.2 MSPI Connection

The MSPI interface has various device configurations. These are mainly handled within the MSPI controller configuration. However, there are some additional pad muxing options to provide more flexibility for system integration. These mux configurations are listed below.

Table 584: MSPI REG_MSPI_PADCFG Input Mux Configuration

Signal	IN0[1:0]				IN1		IN2		IN3	
	3	2	1	0	1	0	1	0	1	0
MSPI0	MSPI D[5]	MSPI D[1]	MSPI D[4]	MSPI D[0]	-	-	-	-	-	
MSPI1	-	-	-	-	MSPI D[5]	MSPI D[1]	-	-	-	-
MSPI2	-	-	-	-	-	-	MSPI D[6]	MSPI D[2]	-	-
MSPI3	-	-	-	-	-	-	-	-	MSPI D[7]	MSPI D[3]

Table 585: MSPI REG_MSPI_PADCFG Output Mux Configuration

Signal	OUT7		OUT6		OUT5		OUT4		OUT3	
	1	0	1	0	1	0	1	0	1	0
MSPI0	MSPI D[0]									
MSPI1	MSPI D[1]									
MSPI2	MSPI D[2]									
MSPI3	-	-	-	-	-	-	-	-	MSPI CLK	MSPI D[3]
MSPI4	-	-	-	-	-	-	MSPI D[0]	MSPI D[4]	-	-
MSPI5	-	-	-	-	MSPI D[1]	MSPI D[5]	-	-	-	-
MSPI6	-	-	MSPI D[2]	MSPI D[6]	-	-	-	-	-	-
MSPI7	MSPI D[3]	MSPI D[7]	-	-	-	-	-	-	-	-
MSPI8	MSPI CLK									

11.5.3 Implementing IO Slave Connections

The IO Master module must be correctly connected to the appropriate pads in order to operate.

11.5.3.1 IO Slave I2C Connection

I2C 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 586. The PAD0INPEN and PAD1INPEN bits must be set. PAD0PULL and PAD1PULL should be cleared.

Table 586: IO Slave I2C Configuration

Field	Value
PAD0FNCSEL	0
PAD1FNCSEL	0

11.5.3.2 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 587. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD1PULL, PAD2PULL and PAD3PULL should be cleared.

Table 587: IO Slave 4-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD1FNCSEL	1
PAD2FNCSEL	1
PAD3FNCSEL	1

11.5.3.3 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 588. 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 588: IO Slave 3-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD2FNCSEL	0
PAD3FNCSEL	1

11.5.3.4 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.

11.5.4 Implementing Counter/Timer Connections

Each Counter/Timer can optionally count pulses from an input pad, or generate pulses on an output pad. Table 589 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 589: Counter/Timer Pad Configuration

Pad (FNCSEL)	ctimer output signal	Output Selection (REG_CTIMER_INCFG)							
		0	1	2	3	4	5	6	7
PAD4 (6)	CT17	Force to 0	Force to 1	A4OUT2	B7OUT	A4OUT	A1OUT2	A6OUT2	A7OUT2
PAD5 (7)	CT8	Force to 0	Force to 1	A2OUT	A3OUT2	A4OUT2	B6OUT	A6OUT2	A7OUT2
PAD6 (5)	CT10	Force to 0	Force to 1	B2OUT	B3OUT2	B4OUT2	A6OUT	A6OUT2	A7OUT2
PAD7 (7)	CT19	Force to 0	Force to 1	B4OUT2	A2OUT	B4OUT	B1OUT2	A6OUT2	A7OUT2
PAD11 (2)	CT31	Force to 0	Force to 1	B7OUT2	A6OUT	B7OUT	B3OUT2	A6OUT2	A7OUT2
PAD12 (2)	CT0	Force to 0	Force to 1	A0OUT	B2OUT2	A5OUT2	A6OUT	A6OUT2	A7OUT2
PAD13 (2)	CT2	Force to 0	Force to 1	B0OUT	B1OUT2	B6OUT2	A7OUT	A6OUT2	A7OUT2
PAD18 (2)	CT4	Force to 0	Force to 1	A1OUT	A2OUT2	A5OUT2	B5OUT	A6OUT2	A7OUT2
PAD19 (2)	CT6	Force to 0	Force to 1	B1OUT	A1OUT	B5OUT2	B7OUT	A6OUT2	A7OUT2
PAD22 (2)	CT12	Force to 0	Force to 1	A3OUT	B1OUT	B0OUT2	B6OUT2	A6OUT2	A7OUT2
PAD23 (2)	CT14	Force to 0	Force to 1	B3OUT	B1OUT	B7OUT2	A7OUT	A6OUT2	A7OUT2
PAD24 (5)	CT21	Force to 0	Force to 1	A5OUT2	A1OUT	B5OUT	A0OUT2	A6OUT2	A7OUT2
PAD25 (2)	CT1	Force to 0	Force to 1	A0OUT2	A0OUT	A5OUT	B7OUT2	A6OUT2	A7OUT2
PAD26 (2)	CT3	Force to 0	Force to 1	B0OUT2	B0OUT	A1OUT	A6OUT	A6OUT2	A7OUT2
PAD27 (2)	CT5	Force to 0	Force to 1	A1OUT2	A1OUT	B6OUT	A7OUT	A6OUT2	A7OUT2
PAD28 (2)	CT7	Force to 0	Force to 1	B1OUT2	B1OUT	B5OUT	A7OUT	A6OUT2	A7OUT2
PAD29 (2)	CT9	Force to 0	Force to 1	A2OUT2	A2OUT	A4OUT	B0OUT	A6OUT2	A7OUT2
PAD30 (2)	CT11	Force to 0	Force to 1	B2OUT2	B2OUT	B4OUT	B5OUT2	A6OUT2	A7OUT2
PAD31 (2)	CT13	Force to 0	Force to 1	A3OUT2	A3OUT	A6OUT	B4OUT2	A6OUT2	A7OUT2
PAD32 (2)	CT15	Force to 0	Force to 1	B3OUT2	B3OUT	A7OUT	A4OUT2	A6OUT2	A7OUT2
PAD33 (6)	CT23	Force to 0	Force to 1	B5OUT2	A7OUT	A5OUT	B0OUT2	A6OUT2	A7OUT2

Table 589: Counter/Timer Pad Configuration

Pad (FNCSEL)	ctimer output signal	Output Selection (REG_CTIMER_INCFG)							
		0	1	2	3	4	5	6	7
PAD35 (5)	CT27	Force to 0	Force to 1	B6OUT2	A1OUT	B6OUT	B2OUT2	A6OUT2	A7OUT2
PAD37 (7)	CT29	Force to 0	Force to 1	B5OUT2	A1OUT	A7OUT	A3OUT2	A6OUT2	A7OUT2
PAD39 (2)	CT25	Force to 0	Force to 1	B4OUT2	B2OUT	A6OUT	A2OUT2	A6OUT2	A7OUT2
PAD42 (2)	CT16	Force to 0	Force to 1	A4OUT	A0OUT	A0OUT2	B3OUT2	A6OUT2	A7OUT2
PAD43 (2)	CT18	Force to 0	Force to 1	B4OUT	B0OUT	A0OUT	A3OUT2	A6OUT2	A7OUT2
PAD44 (2)	CT20	Force to 0	Force to 1	A5OUT	A1OUT	A1OUT2	B2OUT2	A6OUT2	A7OUT2
PAD45 (2)	CT22	Force to 0	Force to 1	B5OUT	B1OUT	A6OUT	A2OUT2	A6OUT2	A7OUT2
PAD46 (2)	CT24	Force to 0	Force to 1	A6OUT	A2OUT	A1OUT	B1OUT2	A6OUT2	A7OUT2
PAD47 (2)	CT26	Force to 0	Force to 1	B6OUT	B2OUT	A5OUT	A1OUT2	A6OUT2	A7OUT2
PAD48 (2)	CT28	Force to 0	Force to 1	A7OUTB	A3OUT	A5OUT2	B0OUT2	A6OUT2	A7OUT2
PAD49 (2)	CT30	Force to 0	Force to 1	B7OUT	B3OUT	A4OUT2	A0OUT2	A6OUT2	A7OUT2

11.5.5 Implementing UART Connections

The UART signals can be connected to a variety of pads.

11.5.5.1 UART0 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 590 shows the connections for TX, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 591 shows the connections for RX, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 590: UART0 TX Configuration

Field	Value	Pad
PAD1FNCSEL	2	1
PAD7FNCSEL	5	7
PAD16FNCSEL	6	16
PAD20FNCSEL	4	20
PAD22FNCSEL	0	22
PAD26FNCSEL	6	26
PAD28FNCSEL	6	28
PAD30FNCSEL	4	30
PAD39FNCSEL	0	39
PAD41FNCSEL	6	41
PAD44FNCSEL	6	44
PAD48FNCSEL	0	48

Table 591: UART0 RX Configuration

Field	Value	Pad
PAD2FNCSEL	2	2
PAD11FNCSEL	6	11
PAD17FNCSEL	6	17
PAD21FNCSEL	4	21
PAD23FNCSEL	0	23
PAD27FNCSEL	0	27
PAD29FNCSEL	6	29
PAD31FNCSEL	4	31
PAD40FNCSEL	0	40
PAD45FNCSEL	6	45
PAD49FNCSEL	0	49

11.5.5.2 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 592 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 593 shows the connections for CTS,

which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 592: UART0 RTS Configuration

Field	Value	Pad
PAD3FNCSEL	0	3
PAD5FNCSEL	2	5
PAD13FNCSEL	6	13
PAD18FNCSEL	4	18
PAD34FNCSEL	5	34
PAD35FNCSEL	6	35
PAD37FNCSEL	2	37
PAD41FNCSEL	7	41

Table 593: UART0 CTS Configuration

Field	Value	Pad
PAD4FNCSEL	0	4
PAD6FNCSEL	2	6
PAD12FNCSEL	6	12
PAD24FNCSEL	4	24
PAD29FNCSEL	4	29
PAD33FNCSEL	5	33
PAD36FNCSEL	6	36
PAD38FNCSEL	2	38

11.5.5.3 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 594 shows the connections for TX, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 595 shows the connections for RX, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 594: UART1 TX Configuration

Field	Value	Pad
PAD8FNCSEL	6	8
PAD10FNCSEL	0	10
PAD12FNCSEL	7	12
PAD14FNCSEL	2	14
PAD18FNCSEL	6	18
PAD20FNCSEL	5	20
PAD24FNCSEL	0	24
PAD35FNCSEL	2	35
PAD37FNCSEL	5	37
PAD39FNCSEL	1	39
PAD42FNCSEL	0	42
PAD47FNCSEL	6	47

Table 595: UART1 RX Configuration

Field	Value	Pad
PAD2FNCSEL	0	2
PAD4FNCSEL	5	4
PAD9FNCSEL	6	9
PAD13FNCSEL	7	13
PAD15FNCSEL	2	15
PAD19FNCSEL	6	19
PAD21FNCSEL	5	21
PAD25FNCSEL	0	25
PAD36FNCSEL	2	36
PAD40FNCSEL	1	40
PAD43FNCSEL	0	43
PAD47FNCSEL	6	47

11.5.5.4 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 596 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 597 shows the connections for CTS,

which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 596: UART1 RTS Configuration

Field	Value	Pad
PAD10FNCSEL	5	10
PAD16FNCSEL	7	16
PAD20FNCSEL	7	20
PAD30FNCSEL	5	30
PAD31FNCSEL	7	31
PAD34FNCSEL	2	34
PAD41FNCSEL	5	41
PAD44FNCSEL	0	44

Table 597: UART1 CTS Configuration

Field	Value	Pad
PAD11FNCSEL	5	11
PAD17FNCSEL	7	17
PAD21FNCSEL	7	21
PAD26FNCSEL	7	26
PAD29FNCSEL	5	29
PAD32FNCSEL	7	32
PAD36FNCSEL	5	36
PAD45FNCSEL	0	45

11.5.6 Implementing Audio Connections

The Audio signals can be connected to a variety of pads.

11.5.6.1 PDM Connections

The PDM CLK and DATA signals may each be connected to several pads. Note that CLK and DATA are selected independently. Table 598 shows the connections for PDM CLK, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 599 shows the connections for PDM DATA, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 598: PDM CLK Configuration

Field	Value	Pad
PAD10FNCSEL	4	10
PAD12FNCSEL	5	12
PAD14FNCSEL	4	14
PAD22FNCSEL	4	22
PAD37FNCSEL	6	37
PAD46FNCSEL	5	46

Table 599: PDM DATA Configuration

Field	Value	Pad
PAD11FNCSEL	7	11
PAD15FNCSEL	4	15
PAD29FNCSEL	7	29
PAD34FNCSEL	7	34
PAD36FNCSEL	7	36
PAD45FNCSEL	5	45

11.5.6.2 I2S Connections

The I2S BCLK, WCLK and DAT signalsmay each be connected to several pads. Note that BCLK, WCLK and DAT are selected independentlyTable 600 shows the connections for I2S BCLK, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 601 shows the connections for I2S WCLK, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 602 shows the connections for I2S DAT, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 600: I2S BCLK Configuration

Field	Value	Pad
PAD13FNCSEL	4	13
PAD19FNCSEL	7	19
PAD20FNCSEL	6	20
PAD46FNCSEL	0	46

Table 601: I2S WCLK Configuration

Field	Value	Pad
PAD3FNCSEL	7	3
PAD23FNCSEL	4	23
PAD28FNCSEL	0	28
PAD41FNCSEL	4	41

Table 602: I2S DAT Configuration

Field	Value	Pad
PAD6FNCSEL	7	6
PAD30FNCSEL	7	30
PAD35FNCSEL	4	35
PAD45FNCSEL	4	45

11.5.7 Implementing Secure Card Connections

The Secure Card signals can be connected to a variety of pads. Table 603 shows the connections for Secure Card CLK (SCCCLK), which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 604 shows the connections for Secure Card IO (SCCIO), which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.Table 605 shows the connections for Secure Card RSTIO (SCCRST), which must have the corresponding PADnINPEN and PADnPULL bits clear.

Table 603: Secure Card Clock Configuration

Field	Value	Pad
PAD8FNCSEL	4	8
PAD17FNCSEL	4	17
PAD19FNCSEL	4	19
PAD31FNCSEL	5	31

Table 604: Secure Card IO Configuration

Field	Value	Pad
PAD9FNCSEL	4	9
PAD18FNCSEL	7	18
PAD32FNCSEL	4	32
PAD37FNCSEL	4	37

Table 605: Secure Card RST Configuration

Field	Value	Pad
PAD16FNCSEL	4	16
PAD21FNCSEL	6	21
PAD26FNCSEL	4	26
PAD46FNCSEL	4	46

11.5.8 Implementing GPIO Connections

Each pad of the Apollo3 Blue MCU 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.

11.5.9 Implementing CLKOUT Connections

The flexible clock output of the Clock Generator module, CLKOUT, may be configured on several pads as shown in . PADnINPEN and PADnPULL should be cleared in each case.

Table 606: CLKOUT Configuration

Field	Value	Pad
PAD0FNCSEL	2	0
PAD7FNCSEL	2	7

11.5.10 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 Apollo3 Blue MCU. This clock output may be configured on several pads as shown in Table 607. PADnINPEN and PADnPULL should be cleared in each case.

Table 607: 32kHz CLKOUT Configuration

Field	Value	Pad
PAD14FNCSEL	7	14
PAD24FNCSEL	6	24
PAD33FNCSEL	2	33
PAD36FNCSEL	4	36
PAD47FNCSEL	0	47

11.5.11 Implementing ADC Connections

Three types of pad connections may be made for the ADC module. Up to ten pads may be configured as the analog inputs, as shown in Table 608. The ADCREF reference voltage input supplied on a dedicated input pin. If an external digital trigger is desired, up to four selectable pad choices may be configured, as shown in Table 609. 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 608: 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

Table 608: ADC Analog Input Configuration

Field	Value	Input	Pad
PAD12FNCSEL	0	ADCD0P/ SE9	12
PAD14FNCSEL	0	ADCD1P	14
PAD15FNCSEL	0	ADCD1M	15

I

Table 609: 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

11.5.12mplementing 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 610. The voltage to be applied to the comparator positive input are shown in . In each case PADnINPEN and PADnPULL should be clearedNote 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.

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 610: Voltage Comparator Reference Configuration

Field	Value	Input	Pad
PAD19FNCSEL	0	CMPRF0	19
PAD17FNCSEL	0	CMPRF1	17
PAD34FNCSEL	4	CMPRF2	34

Table 611: Voltage Comparator Input Configuration

Field	Value	Input	Pad
PAD16FNCSEL	5	CMPIN0	16

Table 611: Voltage Comparator Input Configuration

Field	Value	Input	Pad
PAD18FNCSEL	0	CMPIN1	18

11.5.13Implementing 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 612, and PADnINPEN and PADnPULL should be cleared for the selected pad.

Table 612: SWO Configuration

Field	Value	Pad
PAD15FNCSEL	7	15
PAD22FNCSEL	7	22
PAD24FNCSEL	7	24
PAD33FNCSEL	7	33
PAD41FNCSEL	2	41
PAD45FNCSEL	7	45
PAD46FNCSEL	7	46

11.5.14Fast GPIO

11.5.14.1Description

Access to GPIO pin registers on the Apollo3 Blue MCU can be multiple CPU cycles to complete. To support certain functions that require shorter latency access, a fast GPIO interface is supported. The fast GPIO is accessed via the fast GPIO registers shown in the next section.

11.6 FASTGPIO Registers

APB DMA Register Interfaces

INSTANCE 0 BASE ADDRESS:0x40011000

11.6.1 Register Memory Map

Table 613: FASTGPIO Register Map

Address(s)	Register Name	Description
0x40011000	BBVALUE	Control Register
0x40011004	BBSETCLEAR	Set/Clear Register
0x40011008	BBINPUT	PIO Input Values
0x40011020	DEBUGDATA	PIO Input Values
0x40011040	DEBUG	PIO Input Values

11.6.2 FASTGPIO Registers

11.6.2.1 BBVALUE Register

Control Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40011000

Control Register

Table 614: BBVALUE Register

31	30	29	28	27	26	25	24	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								PIN								RSVD								DATAOUT							

Table 615: BBVALUE Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:16	PIN	0x0	RO	PIO values
15:8	RSVD	0x0	RO	RESERVED
7:0	DATAOUT	0x0	RW	Data Output Values

11.6.2.2 BBSETCLEAR Register

Set/Clear Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40011004

Set/Clear Register

Table 616: BBSETCLEAR Register

31	30	29	28	27	26	25	24	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								CLEAR								RSVD								SET							

Table 617: BBSETCLEAR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:16	CLEAR	0x0	WO	Write 1 to Clear PIO value
15:8	RSVD	0x0	RO	RESERVED
7:0	SET	0x0	WO	Write 1 to Set PIO value (set hier priority than clear if both bit set)

11.6.2.3 BBINPUT Register

PIO Input Values

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40011008

PIO Input Values

Table 618: BBINPUT Register

31	30	29	28	27	26	25	24	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																								DATAIN							

Table 619: BBINPUT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7:0	DATAIN	0x0	RO	PIO values

11.6.2.4 DEBUGDATA Register

PIO Input Values

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40011020

PIO Input Values

Table 620: DEBUGDATA Register

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

Table 621: DEBUGDATA Register Bits

Bit	Name	Reset	RW	Description
31:0	DEBUGDATA	0x0	RO	Debug Data

11.6.2.5 DEBUG Register

PIO Input Values

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x40011040

PIO Input Values

Table 622: DEBUG Register

31	30	29	28	27	26	25	24	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																												DEBUGEN			

Table 623: DEBUG Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED
3:0	DEBUGEN	0x0	RW	Debug Enable OFF = 0x0 - Debug Disabled ARB = 0x1 - Debug Arb values

11.7 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 multiplexor. 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.

11.7.1 Register Memory Map

Table 624: GPIO Register Map

Address(s)	Register Name	Description
0x40010000	PADREGA	Pad Configuration Register A (Pads 0-3)
0x40010004	PADREGB	Pad Configuration Register B (Pads 4-7)
0x40010008	PADREGC	Pad Configuration Register C (Pads 8-11)
0x4001000C	PADREGD	Pad Configuration Register D (Pads 12-15)
0x40010010	PADREGE	Pad Configuration Register E (Pads 16-19)
0x40010014	PADREGF	Pad Configuration Register F (Pads 20-23)
0x40010018	PADREGG	Pad Configuration Register G (Pads 24-27)
0x4001001C	PADREGH	Pad Configuration Register H (Pads 28-31)
0x40010020	PADREGI	Pad Configuration Register I (Pads 32-25)
0x40010024	PADREGJ	Pad Configuration Register J (Pads 36-39)
0x40010028	PADREGK	Pad Configuration Register K (Pads 40-43)
0x4001002C	PADREGL	Pad Configuration Register L (Pads 44-47)
0x40010030	PADREGM	Pad Configuration Register M (Pads 47-48)
0x40010040	CFGA	GPIO Configuration Register A (Pads 0-7)
0x40010044	CFGB	GPIO Configuration Register B (Pads 8-15)
0x40010048	CFGC	GPIO Configuration Register C (Pads 16-23)
0x4001004C	CFGD	GPIO Configuration Register D (Pads 24-31)
0x40010050	CFGE	GPIO Configuration Register E (Pads 32-39)
0x40010054	CFGF	GPIO Configuration Register F (Pads 40 -47)
0x40010058	CFGG	GPIO Configuration Register G (Pads 48-49)
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
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

Table 624: GPIO Register Map

Address(s)	Register Name	Description
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	BLEIFIRQ	BLEIF Flow Control IRQ Select
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 (Pads 15,14,13,12)
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)
0x40010114	SCDET	SCARD Card Detect select
0x40010118	CTENCFG	Counter/Timer Enable Config
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

11.7.2 GPIO Registers

11.7.2.1 PADREGA Register

Pad Configuration Register A (Pads 0-3)

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 625: 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	PAD3PWRUP	PAD3FNCSEL			PAD3STRNG	PAD3INPEN	PAD3PULL	RSVD		PAD2FNCSEL			PAD2STRNG	PAD2INPEN	PAD2PULL	PAD1RSEL		PAD1FNCSEL			PAD1STRNG	PAD1INPEN	PAD1PULL	PAD0RSEL		PAD0FNCSEL			PAD0STRNG	PAD0INPEN	PAD0PULL

Table 626: PADREGA Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	PAD3PWRUP	0x0	RW	Pad 3 VDD power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (switched to VDD)
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 NCE3 = 0x2 - IOM/MSPI nCE group 3 GPIO3 = 0x3 - Configure as GPIO3 RSVD = 0x4 - Reserved MSPI7 = 0x5 - MSPI data connection 7 TRIG1 = 0x6 - Configure as the ADC Trigger 1 signal I2S_WCLK = 0x7 - Configure as the PDM I2S 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 626: PADREGA Register Bits

Bit	Name	Reset	RW	Description
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD2FNCSEL	0x3	RW	Pad 2 function select UART1RX = 0x0 - Configure as the UART1 RX input SLMISO = 0x1 - Configure as the IOSLAVE SPI MISO signal UART0RX = 0x2 - Configure as the UART0 RX input GPIO2 = 0x3 - Configure as GPIO2 RSVD4 = 0x4 - Reserved MSPI6 = 0x5 - CMSPI data connection 6 RSVD6 = 0x6 - Reserved NCE2 = 0x7 - IOM/MSPI nCE group 2
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	PAD1RSEL	0x0	RW	Pad 1 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
13:11	PAD1FNCSEL	0x3	RW	Pad 1 function select SLSDAWIR3 = 0x0 - Configure as the IOSLAVE I2C SDA or SPI WIR3 signal SLMOSI = 0x1 - Configure as the IOSLAVE SPI MOSI signal UART0TX = 0x2 - Configure as the UART0 TX output signal GPIO1 = 0x3 - Configure as GPIO1 RSVD4 = 0x4 - Reserved MSPI5 = 0x5 - MSPI data connection 5 RSVD6 = 0x6 - Reserved NCE1 = 0x7 - IOM/MSPI nCE group 1
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 626: PADREGA Register Bits

Bit	Name	Reset	RW	Description
7:6	PAD0RSEL	0x0	RW	Pad 0 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
5:3	PAD0FNCSEL	0x3	RW	Pad 0 function select SLSCL = 0x0 - Configure as the IOSLAVE I2C SCL signal SLSCK = 0x1 - Configure as the IOSLAVE SPI SCK signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO0 = 0x3 - Configure as GPIO0 RSVD4 = 0x4 - Reserved MSPI4 = 0x5 - MSPI data connection 4 RSVD6 = 0x6 - Reserved NCE0 = 0x7 - IOM/MSPI nCE group 0
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.2 PADREGB Register

Pad Configuration Register B (Pads 4-7)

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 627: 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	RSVD		PAD4FNCSEL			PAD4STRNG	PAD4INPEN	PAD4PULL

Table 628: PADREGB Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD7FNCSEL	0x3	RW	Pad 7 function select NCE7 = 0x0 - IOM/MSPI nCE group 7 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 RSVD = 0x6 - Reserved CT19 = 0x7 - CTIMER connection 19
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 M0SDAWIR3 = 0x0 - Configure as the IOMSTR0 I2C SDA or SPI WIR3 signal M0MISO = 0x1 - Configure as the IOMSTR0 SPI MISO signal UA0CTS = 0x2 - Configure as the UART0 CTS input signal GPIO6 = 0x3 - Configure as GPIO6 RSVD4 = 0x4 - Reserved CT10 = 0x5 - CTIMER connection 10 RSVD6 = 0x6 - Reserved I2S_DAT = 0x7 - Configure as the PDM I2S 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 628: PADREGB Register Bits

Bit	Name	Reset	RW	Description
15:14	PAD5RSEL	0x0	RW	Pad 5 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
13:11	PAD5FNCSEL	0x3	RW	Pad 5 function select M0SCL = 0x0 - Configure as the IOMSTR0 I2C 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 RSVD4 = 0x4 - Reserved RSVD6 = 0x6 - Reserved CT8 = 0x7 - CTIMER connection 8
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	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 NCE4 = 0x2 - IOM/SPI nCE group 4 GPIO4 = 0x3 - Configure as GPIO4 RSVD4 = 0x4 - Reserved UART0RX = 0x5 - Configure as the UART0 RX input CT17 = 0x6 - CTIMER connection 17 MSPI2 = 0x7 - MSPI data connection 2
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.3 PADREGC Register

Pad Configuration Register C (Pads 8-11)

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 629: 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		PAD10FNC-SEL			PAD10STRNG	PAD10INPEN	PAD10PULL	PAD9RSEL		PAD9FNCSEL			PAD9STRNG	PAD9INPEN	PAD9PULL	PAD8RSEL		PAD8FNCSEL			PAD8STRNG	PAD8INPEN	PAD8PULL

Table 630: 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 NCE11 = 0x1 - IOM/MSPI nCE group 11 CT31 = 0x2 - CTIMER connection 31 GPIO11 = 0x3 - Configure as GPIO11 SLINT = 0x4 - Configure as the IOSLAVE interrupt out signal 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED

Table 630: PADREGC Register Bits

Bit	Name	Reset	RW	Description
21:19	PAD10FNCSEL	0x3	RW	Pad 10 function select RSVD0 = 0x0 - Reserved M1MOSI = 0x1 - Configure as the IOMSTR1 SPI MOSI signal NCE10 = 0x2 - IOM/MSPI nCE group 10 GPIO10 = 0x3 - Configure as GPIO10 PDMCLK = 0x4 - PDM serial clock out UA1RTS = 0x5 - Configure as the UART1 RTS output signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - REserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	PAD9RSEL	0x0	RW	Pad 9 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
13:11	PAD9FNCSEL	0x3	RW	Pad 9 function select M1SDAWIR3 = 0x0 - Configure as the IOMSTR1 I2C SDA or SPI WIR3 signal M1MISO = 0x1 - Configure as the IOMSTR1 SPI MISO signal NCE9 = 0x2 - IOM/MSPI nCE group 9 GPIO9 = 0x3 - Configure as GPIO9 SCCIO = 0x4 - SCARD data I/O connection RSVD5 = 0x5 - Reserved UART1RX = 0x6 - Configure as UART1 RX input signal RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 630: PADREGC Register Bits

Bit	Name	Reset	RW	Description
7:6	PAD8RSEL	0x0	RW	Pad 8 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
5:3	PAD8FNCSEL	0x3	RW	Pad 8 function select M1SCL = 0x0 - Configure as the IOMSTR1 I2C SCL signal M1SCK = 0x1 - Configure as the IOMSTR1 SPI SCK signal NCE8 = 0x2 - IOM/MSPI nCE group 8 GPIO8 = 0x3 - Configure as GPIO8 SCCLK = 0x4 - SCARD serial clock output RSVD5 = 0x5 - Reserved UART1TX = 0x6 - Configure as the UART1 TX output signal RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.4 PADREGD Register

Pad Configuration Register D (Pads 12-15)

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 631: 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		PAD15FNC-SEL			PAD15STRNG	PAD15INPEN	PAD15PULL	RSVD		PAD14FNC-SEL			PAD14STRNG	PAD14INPEN	PAD14PULL	RSVD		PAD13FNC-SEL			PAD13STRNG	PAD13INPEN	PAD13PULL	RSVD		PAD12FNC-SEL			PAD12STRNG	PAD12INPEN	PAD12PULL

Table 632: 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 NCE15 = 0x1 - IOM/MSPI nCE group 15 UART1RX = 0x2 - Configure as the UART1 RX signal GPIO15 = 0x3 - Configure as GPIO15 PDMDATA = 0x4 - PDM serial data input 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE14 = 0x1 - IOM/MSPI nCE group 14 UART1TX = 0x2 - Configure as the UART1 TX output signal GPIO14 = 0x3 - Configure as GPIO14 PDMCLK = 0x4 - PDM serial clock output SWDCK = 0x6 - Configure as the alternate input for the SWDCK input signal 32kHzXT = 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED

Table 632: PADREGD Register Bits

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 NCE13 = 0x1 - IOM/MSPI nCE group 13 CT2 = 0x2 - CTIMER connection 2 GPIO13 = 0x3 - Configure as GPIO13 I2SBCLK = 0x4 - I2C interface bit clock 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE12 = 0x1 - IOM/MSPI nCE group 12 CT0 = 0x2 - CTIMER connection 0 GPIO12 = 0x3 - Configure as GPIO12 SLnCE = 0x4 - Configure as the IOSLAVE SPI nCE signal PDMCLK = 0x5 - PDM serial clock output 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.5 PADREGE Register

Pad Configuration Register E (Pads 16-19)

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 633: 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		PAD19FNC-SEL			PAD19STRNG	PAD19INPEN	PAD19PULL	RSVD		PAD18FNC-SEL			PAD18STRNG	PAD18INPEN	PAD18PULL	RSVD		PAD17FNC-SEL			PAD17STRNG	PAD17INPEN	PAD17PULL	RSVD		PAD16FNC-SEL			PAD16STRNG	PAD16INPEN	PAD16PULL

Table 634: 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 NCE19 = 0x1 - IOM/MSPI nCE group 19 CT6 = 0x2 - CTIMER conenction 6 GPIO19 = 0x3 - Configure as GPIO19 SCCLK = 0x4 - SCARD serial clock ANATEST1 = 0x5 - Configure as the ANATEST1 I/O signal UART1RX = 0x6 - Configure as the UART1 RX input signal I2SBCLK = 0x7 - Configure as the PDM I2S bit 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

Table 634: PADREGE Register Bits

Bit	Name	Reset	RW	Description
21:19	PAD18FNCSEL	0x3	RW	Pad 18 function select CMPIN1 = 0x0 - Configure as the analog comparator input 1 signal NCE18 = 0x1 - IOM/MSPI nCE group 18 CT4 = 0x2 - CTIMER connection 4 GPIO18 = 0x3 - Configure as GPIO18 UA0RTS = 0x4 - Configure as UART0 RTS output signal ANATEST2 = 0x5 - Configure as ANATEST2 I/O signal UART1TX = 0x6 - Configure as UART1 TX output signal SCCIO = 0x7 - SCARD data input/output connectin
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE17 = 0x1 - IOM/MSPI nCE group 17 TRIG1 = 0x2 - Configure as the ADC Trigger 1 signal GPIO17 = 0x3 - Configure as GPIO17 SCCCLK = 0x4 - SCARD serial clock output RSVD = 0x5 - Reserved 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	RSVD	0x0	RO	RESERVED

Table 634: PADREGE Register Bits

Bit	Name	Reset	RW	Description
5:3	PAD16FNCSEL	0x3	RW	Pad 16 function select ADCSE0 = 0x0 - Configure as the analog ADC single ended port 0 input signal NCE16 = 0x1 - IOM/MSPI nCE group 16 TRIG0 = 0x2 - Configure as the ADC Trigger 0 signal GPIO16 = 0x3 - Configure as GPIO16 SCCRST = 0x4 - SCARD reset output 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
1	PAD16INPEN	0x0	RW	Pad 16 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD16PULL	0x0	RW	Pad 16 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.6 PADREGF Register

Pad Configuration Register F (Pads 20-23)

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 635: 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		PAD23FNC-SEL			PAD23STRNG	PAD23INPEN	PAD23PULL	RSVD		PAD22FNC-SEL			PAD22STRNG	PAD22INPEN	PAD22PULL	RSVD		PAD21FNC-SEL			PAD21STRNG	PAD21INPEN	PAD21PULL	RSVD		PAD20FNC-SEL			PAD20STRNG	PAD20INPEN	PAD20PULL

Table 636: PADREGF Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED

Table 636: PADREGF Register Bits

Bit	Name	Reset	RW	Description
29:27	PAD23FNCSEL	0x3	RW	Pad 23 function select UART0RX = 0x0 - Configure as the UART0 RX signal NCE23 = 0x1 - IOM/MSPI nCE group 23 CT14 = 0x2 - CTIMER connection 14 GPIO23 = 0x3 - Configure as GPIO23 I2SWCLK = 0x4 - I2S word clock input CMPOUT = 0x5 - Configure as voltage comparitor output MSPI3 = 0x6 - MSPI data connection 3
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD22FNCSEL	0x3	RW	Pad 22 function select UART0TX = 0x0 - Configure as the UART0 TX signal NCE22 = 0x1 - IOM/MSPI nCE group 22 CT12 = 0x2 - CTIMER connection 12 GPIO22 = 0x3 - Configure as GPIO22 PDM_CLK = 0x4 - Configure as the PDM CLK output MSPI0 = 0x6 - MSPI data connection 0 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED

Table 636: PADREGF Register Bits

Bit	Name	Reset	RW	Description
13:11	PAD21FNCSEL	0x0	RW	Pad 21 function select SWDIO = 0x0 - Configure as the serial wire debug data signal NCE21 = 0x1 - IOM/MSPI nCE group 21 RSVD = 0x2 - Reserved GPIO21 = 0x3 - Configure as GPIO21 UART0RX = 0x4 - Configure as UART0 RX input signal UART1RX = 0x5 - Configure as UART1 RX input signal I2SBCLK = 0x6 - I2S byte clock input UA1CTS = 0x7 - Configure as UART1 CTS input signal
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD20FNCSEL	0x0	RW	Pad 20 function select SWDCK = 0x0 - Configure as the serial wire debug clock signal NCE20 = 0x1 - IOM/MSPI nCE group 20 RSVD = 0x2 - Reserved GPIO20 = 0x3 - Configure as GPIO20 UART0TX = 0x4 - Configure as UART0 TX output signal UART1TX = 0x5 - Configure as UART1 TX output signal I2SBCLK = 0x6 - I2S byte clock input UA1RTS = 0x7 - Configure as UART1 RTS output signal
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 pulldown enable DIS = 0x0 - Pulldown disabled EN = 0x1 - Pulldown enabled

11.7.2.7 PADREGG Register

Pad Configuration Register G (Pads 24-27)

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 637: 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		PAD27FNC-SEL			PAD27STRNG	PAD27INPEN	PAD27PULL	RSVD		PAD26FNC-SEL			PAD26STRNG	PAD26INPEN	PAD26PULL	PAD25RSEL		PAD25FNC-SEL			PAD25STRNG	PAD25INPEN	PAD25PULL	RSVD		PAD24FNC-SEL			PAD24STRNG	PAD24INPEN	PAD24PULL

Table 638: PADREGG Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD27RSEL	0x0	RW	Pad 27 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
29:27	PAD27FNCSEL	0x3	RW	Pad 27 function select UART0RX = 0x0 - Configure as UART0 RX input signal NCE27 = 0x1 - IOM/MSPI nCE group 27 CT5 = 0x2 - CTIMER connection 5 GPIO27 = 0x3 - Configure as GPIO27 M2SCL = 0x4 - Configure as I2C clock I/O signal from IOMSTR2 M2SCK = 0x5 - Configure as SPI clock output signal from IOMSTR2 RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED

Table 638: PADREGG Register Bits

Bit	Name	Reset	RW	Description
21:19	PAD26FNCSEL	0x3	RW	Pad 26 function select NCE26 = 0x1 - IOM/MSPI nCE group 26 CT3 = 0x2 - CTIMER connection 3 GPIO26 = 0x3 - Configure as GPIO26 SCCRST = 0x4 - SCARD reset output MSPI1 = 0x5 - MSPI data connection 1 UART0TX = 0x6 - Configure as UART0 TX output signal UA1CTS = 0x7 - Configure as UART1 CTS input signal
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	PAD25RSEL	0x0	RW	Pad 25 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
13:11	PAD25FNCSEL	0x3	RW	Pad 25 function select UART1RX = 0x0 - Configure as UART1 RX input signal NCE25 = 0x1 - IOM/MSPI nCE group 25 CT1 = 0x2 - CTIMER connection 1 GPIO25 = 0x3 - Configure as GPIO25 M2SDAWIR3 = 0x4 - Configure as the IOMSTR2 I2C SDA or SPI WIR3 signal M2MISO = 0x5 - Configure as the IOMSTR2 SPI MISO input signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	RSVD	0x0	RO	RESERVED

Table 638: PADREGG Register Bits

Bit	Name	Reset	RW	Description
5:3	PAD24FNCSEL	0x3	RW	Pad 24 function select UART1TX = 0x0 - Configure as UART1 TX output signal NCE24 = 0x1 - IOM/MSPI nCE group 24 MSPI8 = 0x2 - MSPI data connection 8 GPIO24 = 0x3 - Configure as GPIO24 UA0CTS = 0x4 - Configure as UART0 CTS input signal CT21 = 0x5 - CTIMER connection 21 32kHzXT = 0x6 - Configure as the 32kHz crystal output 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.8 PADREGH Register

Pad Configuration Register H (Pads 28-31)

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 639: 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		PAD31FNC-SEL			PAD31STRNG	PAD31INPEN	PAD31PULL	RSVD		PAD30FNC-SEL			PAD30STRNG	PAD30INPEN	PAD30PULL	RSVD		PAD29FNC-SEL			PAD29STRNG	PAD29INPEN	PAD29PULL	RSVD		PAD28FNC-SEL			PAD28STRNG	PAD28INPEN	PAD28PULL

Table 640: PADREGH Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED

Table 640: PADREGH Register Bits

Bit	Name	Reset	RW	Description
29:27	PAD31FNCSEL	0x3	RW	Pad 31 function select ADCSE3 = 0x0 - Configure as the analog input for ADC single ended input 3 NCE31 = 0x1 - IOM/MSPI nCE group 31 CT13 = 0x2 - CTIMER connection 13 GPIO31 = 0x3 - Configure as GPIO31 UART0RX = 0x4 - Configure as the UART0 RX input signal SCCCLK = 0x5 - SCARD serial clock output RSVD = 0x6 - Reserved UA1RTS = 0x7 - Configure as UART1 RTS output signal
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD30FNCSEL	0x3	RW	Pad 30 function select ANATEST1 = 0x0 - Configure as the ANATEST1 I/O signal NCE30 = 0x1 - IOM/MSPI nCE group 30 CT11 = 0x2 - CTIMER connection 11 GPIO30 = 0x3 - Configure as GPIO30 UART0TX = 0x4 - Configure as UART0 TX output signal UA1RTS = 0x5 - Configure as UART1 RTS output signal RSVD = 0x6 - Reserved I2S_DAT = 0x7 - Configure as the PDM I2S Data output signal
18	PAD30STRNG	0x0	RW	Pad 30 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD30INPEN	0x0	RW	Pad 30 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD30PULL	0x0	RW	Pad 30 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED

Table 640: PADREGH Register Bits

Bit	Name	Reset	RW	Description
13:11	PAD29FNCSEL	0x3	RW	Pad 29 function select ADCSE1 = 0x0 - Configure as the analog input for ADC single ended input 1 NCE29 = 0x1 - IOM/MSPI nCE group 29 CT9 = 0x2 - CTIMER connection 9 GPIO29 = 0x3 - Configure as GPIO29 UA0CTS = 0x4 - Configure as the UART0 CTS input signal UA1CTS = 0x5 - Configure as the UART1 CTS input signal UART0RX = 0x6 - Configure as the UART0 RX input signal 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD28FNCSEL	0x3	RW	Pad 28 function select I2S_WCLK = 0x0 - Configure as the PDM I2S Word Clock input NCE28 = 0x1 - IOM/MSPI nCE group 28 CT7 = 0x2 - CTIMER connection 7 GPIO28 = 0x3 - Configure as GPIO28 RSVD4 = 0x4 - Reserved M2MOSI = 0x5 - Configure as the IOMSTR2 SPI MOSI output signal UART0TX = 0x6 - Configure as the UART0 TX output signal RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.9 PADREGI Register

Pad Configuration Register I (Pads 32-25)

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 641: 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		PAD35FNC-SEL			PAD35STRNG	PAD35INPEN	PAD35PULL	RSVD		PAD34FNC-SEL			PAD34STRNG	PAD34INPEN	PAD34PULL	RSVD		PAD33FNC-SEL			PAD33STRNG	PAD33INPEN	PAD33PULL	RSVD		PAD32FNC-SEL			PAD32STRNG	PAD32INPEN	PAD32PULL

Table 642: 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 NCE35 = 0x1 - IOM/MSPI nCE group 35 UART1TX = 0x2 - Configure as the UART1 TX signal GPIO35 = 0x3 - Configure as GPIO35 I2SDAT = 0x4 - I2S serial data output CT27 = 0x5 - CTIMER connection 27 UA0RTS = 0x6 - Configure as the UART0 RTS output RSVD = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE34 = 0x1 - IOM/MSPI nCE group 34 UA1RTS = 0x2 - Configure as the UART1 RTS output GPIO34 = 0x3 - Configure as GPIO34 CMPRF2 = 0x4 - Configure as the analog comparator reference 2 signal UA0RTS = 0x5 - Configure as the UART0 RTS output UART0RX = 0x6 - Configure as the UART0 RX input PDMDATA = 0x7 - PDM serial data input

Table 642: PADREGI Register Bits

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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE33 = 0x1 - IOM/MSPI nCE group 33 32kHzXT = 0x2 - Configure as the 32kHz crystal output signal GPIO33 = 0x3 - Configure as GPIO33 RSVD = 0x4 - Reserved UA0CTS = 0x5 - Configure as the UART0 CTS input CT23 = 0x6 - CTIMER connection 23 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE32 = 0x1 - IOM/MSPI nCE group 32 CT15 = 0x2 - CTIMER connection 15 GPIO32 = 0x3 - Configure as GPIO32 SCCIO = 0x4 - SCARD serial data input/output RSVD = 0x6 - Reserved UA1CTS = 0x7 - Configure as the UART1 CTS input
2	PAD32STRNG	0x0	RW	Pad 32 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 642: PADREGI Register Bits

Bit	Name	Reset	RW	Description
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

11.7.2.10PADREGJ Register

Pad Configuration Register J (Pads 36-39)

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 643: 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		PAD39FNC-SEL			PAD39STRNG	PAD39INPEN	PAD39PULL	RSVD		PAD38FNC-SEL			PAD38STRNG	PAD38INPEN	PAD38PULL	PAD37PWRDN	RSVD	PAD37FNC-SEL			PAD37STRNG	PAD37INPEN	PAD37PULL	RSVD	PAD36PWRUP	PAD36FNC-SEL			PAD36STRNG	PAD36INPEN	PAD36PULL

Table 644: PADREGJ Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD39RSEL	0x0	RW	Pad 39 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
29:27	PAD39FNCSEL	0x3	RW	Pad 39 function select UART0TX = 0x0 - Configure as the UART0 TX output signal UART1TX = 0x1 - Configure as the UART1 TX output signal CT25 = 0x2 - CTIMER connection 25 GPIO39 = 0x3 - Configure as GPIO39 M4SCL = 0x4 - Configure as the IOMSTR4 I2C SCL signal M4SCK = 0x5 - Configure as the IOMSTR4 SPI SCK signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved

Table 644: PADREGJ Register Bits

Bit	Name	Reset	RW	Description
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD38FNCSEL	0x3	RW	Pad 38 function select TRIG3 = 0x0 - Configure as the ADC Trigger 3 signal NCE38 = 0x1 - IOM/MSPI nCE group 38 UA0CTS = 0x2 - Configure as the UART0 CTS signal GPIO38 = 0x3 - Configure as GPIO38 RSVD4 = 0x4 - Reserved M3MOSI = 0x5 - Configure as the IOMSTR3 SPI MOSI output signal UART1RX = 0x6 - Configure as the UART1 RX input signal RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15	PAD37PWRDN	0x0	RW	Pad 37 VSS power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (switch to GND)
14	RSVD	0x0	RO	RESERVED
13:11	PAD37FNCSEL	0x3	RW	Pad 37 function select TRIG2 = 0x0 - Configure as the ADC Trigger 2 signal NCE37 = 0x1 - IOM/MSPI nCE group 37 UA0RTS = 0x2 - Configure as the UART0 RTS output signal GPIO37 = 0x3 - Configure as GPIO37 SCCIO = 0x4 - SCARD serial data input/output UART1TX = 0x5 - Configure as the UART1 TX output signal PDMCLK = 0x6 - Configure as the PDM CLK output signal CT29 = 0x7 - CTIMER connection 29

Table 644: PADREGJ Register Bits

Bit	Name	Reset	RW	Description
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7	RSVD	0x0	RO	RESERVED
6	PAD36PWRUP	0x0	RW	Pad 36 VDD power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (switched to VDD)
5:3	PAD36FNCSEL	0x3	RW	Pad 36 function select TRIG1 = 0x0 - Configure as the ADC Trigger 1 signal NCE36 = 0x1 - IOM/MSPI nCE group 36 UART1RX = 0x2 - Configure as the UART1 RX input signal GPIO36 = 0x3 - Configure as GPIO36 32kHzXT = 0x4 - Configure as the 32kHz output clock from the crystal UA1CTS = 0x5 - Configure as the UART1 CTS input signal UA0CTS = 0x6 - Configure as the UART0 CTS input signal PDMDATA = 0x7 - PDM serial data input
2	PAD36STRNG	0x0	RW	Pad 36 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD36INPEN	0x0	RW	Pad 36 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD36PULL	0x0	RW	Pad 36 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.11 PADREGK Register

Pad Configuration Register K (Pads 40-43)

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 645: 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		PAD43FNC-SEL			PAD43STRNG	PAD43INPEN	PAD43PULL	PAD42RSEL		PAD42FNC-SEL			PAD42STRNG	PAD42INPEN	PAD42PULL	PAD41PWRDN	RSVD	PAD41FNC-SEL			PAD41STRNG	PAD41INPEN	PAD41PULL	PAD40RSEL		PAD40FNC-SEL			PAD40STRNG	PAD40INPEN	PAD40PULL

Table 646: PADREGK Register Bits

Bit	Name	Reset	RW	Description
31:30	PAD43RSEL	0x0	RW	Pad 43 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
29:27	PAD43FNCSEL	0x3	RW	Pad 43 function select UART1RX = 0x0 - Configure as the UART1 RX input signal NCE43 = 0x1 - IOM/MSPI nCE group 43 CT18 = 0x2 - CTIMER connection 18 GPIO43 = 0x3 - Configure as GPIO43 M3SDAWIR3 = 0x4 - Configure as the IOMSTR3 I2C SDA or SPI WIR3 signal M3MISO = 0x5 - Configure as the IOMSTR3 SPI MISO signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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
24	PAD43PULL	0x0	RW	Pad 43 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	PAD42RSEL	0x0	RW	Pad 42 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

Table 646: PADREGK Register Bits

Bit	Name	Reset	RW	Description
21:19	PAD42FNCSEL	0x3	RW	Pad 42 function select UART1TX = 0x0 - Configure as the UART1 TX output signal NCE42 = 0x1 - IOM/MSPI nCE group 42 CT16 = 0x2 - CTIMER connection 16 GPIO42 = 0x3 - Configure as GPIO42 M3SCL = 0x4 - Configure as the IOMSTR3 I2C SCL clock I/O signal M3SCK = 0x5 - Configure as the IOMSTR3 SPI SCK output RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15	PAD41PWRDN	0x0	RW	Pad 41 power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled (Switch pad to VSS)
14	RSVD	0x0	RO	RESERVED
13:11	PAD41FNCSEL	0x3	RW	Pad 41 function select NCE41 = 0x0 - IOM/MSPI nCE group 41 RSVD = 0x1 - Reserved SWO = 0x2 - Configure as the serial wire debug SWO signal GPIO41 = 0x3 - Configure as GPIO41 I2SWCLK = 0x4 - I2S word clock input UA1RTS = 0x5 - Configure as the UART1 RTS output signal UART0TX = 0x6 - Configure as the UART0 TX output signal UA0RTS = 0x7 - Configure as the UART0 RTS output signal
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 646: PADREGK Register Bits

Bit	Name	Reset	RW	Description
7:6	PAD40RSEL	0x0	RW	Pad 40 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
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 M4SDAWIR3 = 0x4 - Configure as the IOMSTR4 I2C SDA or SPI WIR3 signal M4MISO = 0x5 - Configure as the IOMSTR4 SPI MISO input signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.12PADREGL Register

Pad Configuration Register L (Pads 44-47)

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 647: 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 648: PADREGL Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD47FNCSEL	0x3	RW	Pad 47 function select 32kHzXT = 0x0 - Configure as the 32kHz output clock from the crystal NCE47 = 0x1 - IOM/MSPI nCE group 47 CT26 = 0x2 - CTIMER connection 26 GPIO47 = 0x3 - Configure as GPIO47 RSVD4 = 0x4 - Reserved M5MOSI = 0x5 - Configure as the IOMSTR5 SPI MOSI output signal UART1RX = 0x6 - Configure as the UART1 RX input signal RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE46 = 0x1 - IOM/MSPI nCE group 46 CT24 = 0x2 - CTIMER connection 24 GPIO46 = 0x3 - Configure as GPIO46 SCCRST = 0x4 - SCARD reset output PDMCLK = 0x5 - PDM serial clock output UART1TX = 0x6 - Configure as the UART1 TX output signal 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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED

Table 648: PADREGL Register Bits

Bit	Name	Reset	RW	Description
13:11	PAD45FNCSEL	0x3	RW	Pad 45 function select UA1CTS = 0x0 - Configure as the UART1 CTS input signal NCE45 = 0x1 - IOM/MSPI nCE group 45 CT22 = 0x2 - CTIMER connection 22 GPIO45 = 0x3 - Configure as GPIO45 I2SDAT = 0x4 - I2S serial data output PDMDATA = 0x5 - PDM serial data input UART0RX = 0x6 - Configure as the SPI channel 5 nCE signal from IOM-STR5 SWO = 0x7 - Configure as the serial wire debug SWO signal
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup 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 NCE44 = 0x1 - IOM/MSPI nCE group 44 CT20 = 0x2 - CTIMER connection 20 GPIO44 = 0x3 - Configure as GPIO44 RSVD4 = 0x4 - Reserved M4MOSI = 0x5 - Configure as the IOMSTR4 SPI MOSI signal M5nCE6 = 0x6 - Configure as the SPI channel 6 nCE signal from IOMSTR5 RSVD = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.13PADREGM Register

Pad Configuration Register M (Pads 47-48)

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 649: 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		PAD49FNC-SEL			PAD49STRNG	PAD49INPEN	PAD49PULL	PAD48RSEL		PAD48FNC-SEL			PAD48STRNG	PAD48INPEN	PAD48PULL

Table 650: PADREGM Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15:14	PAD49RSEL	0x0	RW	Pad 49 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
13:11	PAD49FNCSEL	0x3	RW	Pad 49 function select UART0RX = 0x0 - Configure as the UART0 RX input signal NCE49 = 0x1 - IOM/MSPPI nCE group 49 CT30 = 0x2 - CTIMER connection 30 GPIO49 = 0x3 - Configure as GPIO49 M5SDAWIR3 = 0x4 - Configure as the IOMSTR5 I2C SDA or SPI WIR3 signal M5MISO = 0x5 - Configure as the IOMSTR5 SPI MISO input signal RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
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 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 650: PADREGM Register Bits

Bit	Name	Reset	RW	Description
7:6	PAD48RSEL	0x0	RW	Pad 48 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
5:3	PAD48FNCSEL	0x3	RW	Pad 48 function select UART0TX = 0x0 - Configure as the UART0 TX output signal NCE48 = 0x1 - IOM/MSPI nCE group 48 CT28 = 0x2 - CTIMER conenction 28 GPIO48 = 0x3 - Configure as GPIO48 M5SCL = 0x4 - Configure as the IOMSTR5 I2C SCL clock I/O signal M5SCK = 0x5 - Configure as the IOMSTR5 SPI SCK output RSVD6 = 0x6 - Reserved RSVD7 = 0x7 - Reserved
2	PAD48STRNG	0x0	RW	Pad 48 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD48INPEN	0x0	RW	Pad 48 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD48PULL	0x0	RW	Pad 48 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

11.7.2.14CFGA Register

GPIO Configuration Register A (Pads 0-7)

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 651: 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 652: CFGA Register Bits

Bit	Name	Reset	RW	Description
31	GPIO7INTD	0x0	RW	GPIO7 interrupt direction, nCE polarity. nCELOW = 0x0 - FNCSEL = 0x0 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x0 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x0, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x0, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x0, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x0, INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO7OUTCFG	0x0	RW	GPIO7 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x0 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x0 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x0 - IOM5 nCE, Channel 1 MSPInCE0 = 0x3 - FNCSEL = 0x0 - MSPI nCE, Channel 0
28	GPIO7INCFG	0x0	RW	GPIO7 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
27	GPIO6INTD	0x0	RW	GPIO6 interrupt direction. INTDIS = 0x0 - INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - INCFG = 0 - Interrupt on high to low GPIO transition
26:25	GPIO6OUTCFG	0x0	RW	GPIO6 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state
24	GPIO6INCFG	0x0	RW	GPIO6 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
23	GPIO5INTD	0x0	RW	GPIO5 interrupt direction. INTDIS = 0x0 - INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - INCFG = 0 - Interrupt on high to low GPIO transition

Table 652: CFGA Register Bits

Bit	Name	Reset	RW	Description
22:21	GPIO5OUTCFG	0x0	RW	GPIO5 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state
20	GPIO5INCFG	0x0	RW	GPIO5 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
19	GPIO4INTD	0x0	RW	GPIO4 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x2 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x2 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x2, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x2, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x2, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x2, INCFG = 0 - Interrupt on high to low GPIO transition
18:17	GPIO4OUTCFG	0x0	RW	GPIO4 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x2 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x2 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x2 - IOM5 nCE, Channel 1 M1nCE1 = 0x3 - FNCSEL = 0x2 - IOM1 nCE, Channel 1
16	GPIO4INCFG	0x0	RW	GPIO4 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
15	GPIO3INTD	0x0	RW	GPIO3 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x2 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x2 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x2, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x2, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x2, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x2, INCFG = 0 - Interrupt on high to low GPIO transition

Table 652: CFGA Register Bits

Bit	Name	Reset	RW	Description
14:13	GPIO3OUTCFG	0x0	RW	GPIO3 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE0 = 0x0 - FNCSEL = 0x2 - IOM3 nCE, Channel 0 M4nCE0 = 0x1 - FNCSEL = 0x2 - IOM4 nCE, Channel 0 M5nCE0 = 0x2 - FNCSEL = 0x2 - IOM5 nCE, Channel 0 M2nCE0 = 0x3 - FNCSEL = 0x2 - IOM2 nCE, Channel 0
12	GPIO3INCFG	0x0	RW	GPIO3 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
11	GPIO2INTD	0x0	RW	GPIO2 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x7 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x7 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x7, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x7, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x7, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x7, INCFG = 0 - Interrupt on high to low GPIO transition
10:9	GPIO2OUTCFG	0x0	RW	GPIO2 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x7 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x7 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x7 - IOM5 nCE, Channel 3 M2nCE1 = 0x3 - FNCSEL = 0x7 - IOM2 nCE, Channel 1
8	GPIO2INCFG	0x0	RW	GPIO2 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
7	GPIO1INTD	0x0	RW	GPIO1 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x7 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x7 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x7, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x7, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x7, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x7, INCFG = 0 - Interrupt on high to low GPIO transition

Table 652: CFGA Register Bits

Bit	Name	Reset	RW	Description
6:5	GPIO1OUTCFG	0x0	RW	GPIO1 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x7 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x7 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x7 - IOM2 nCE, Channel 2 MSPInCE0 = 0x3 - FNCSEL = 0x7 - MSPI nCE, Channel 0
4	GPIO1INCFG	0x0	RW	GPIO1 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
3	GPIO0INTD	0x0	RW	GPIO0 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x7 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x7 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x7, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x7, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x7, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x7, INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO0OUTCFG	0x0	RW	GPIO0 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x7 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x7 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x7 - IOM5 nCE, Channel 2 M1nCE3 = 0x3 - FNCSEL = 0x7 - IOM1 nCE, Channel 3
0	GPIO0INCFG	0x0	RW	GPIO0 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.15CFGB Register

GPIO Configuration Register B (Pads 8-15)

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 653: 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 654: CFGB Register Bits

Bit	Name	Reset	RW	Description
31	GPIO15INTD	0x0	RW	GPIO15 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO15OUT-CFG	0x0	RW	GPIO15 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE3 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 3 M1nCE3 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 3 M2nCE3 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 3 MSPInCE0 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 0
28	GPIO15INCFG	0x0	RW	GPIO15 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
27	GPIO14INTD	0x0	RW	GPIO14 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition

Table 654: CFGB Register Bits

Bit	Name	Reset	RW	Description
26:25	GPIO14OUT-CFG	0x0	RW	GPIO14 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 2 M4nCE2 = 0x3 - FNCSEL = 0x1 - IOM4 nCE, Channel 2
24	GPIO14INCFG	0x0	RW	GPIO14 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
23	GPIO13INTD	0x0	RW	GPIO13 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
22:21	GPIO13OUT-CFG	0x0	RW	GPIO13 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 1 M0nCE1 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 1
20	GPIO13INCFG	0x0	RW	GPIO13 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
19	GPIO12INTD	0x0	RW	GPIO12 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition

Table 654: CFGB Register Bits

Bit	Name	Reset	RW	Description
18:17	GPIO12OUT-CFG	0x0	RW	GPIO12 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE0 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 0 M4nCE0 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 0 M5nCE0 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 0 MSPInCE1 = 0x3 - FNCSEL = 0x1 - MPSI nCE, Channel 1
16	GPIO12INCFG	0x0	RW	GPIO12 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
15	GPIO11INTD	0x0	RW	GPIO11 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
14:13	GPIO11OUT-CFG	0x0	RW	GPIO11 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M3nCE0 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 0
12	GPIO11INCFG	0x0	RW	GPIO11 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
11	GPIO10INTD	0x0	RW	GPIO10 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x2 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x2 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x2, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x2, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x2, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x2, INCFG = 0 - Interrupt on high to low GPIO transition

Table 654: CFGB Register Bits

Bit	Name	Reset	RW	Description
10:9	GPIO10OUT-CFG	0x0	RW	GPIO10 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 2 MSPInCE0 = 0x3 - FNCSEL = 0x1 - MPSI nCE, Channel 0
8	GPIO10INCFG	0x0	RW	GPIO10 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
7	GPIO9INTD	0x0	RW	GPIO9 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x2 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x2 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x2, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x2, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x2, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x2, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO9OUTCFG	0x0	RW	GPIO9 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 3 M2nCE3 = 0x3 - FNCSEL = 0x1 - IOM2 nCE, Channel 3
4	GPIO9INCFG	0x0	RW	GPIO9 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
3	GPIO8INTD	0x0	RW	GPIO8 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x2 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x2 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x2, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x2, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x2, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x2, INCFG = 0 - Interrupt on high to low GPIO transition

Table 654: CFGB Register Bits

Bit	Name	Reset	RW	Description
2:1	GPIO8OUTCFG	0x0	RW	GPIO8 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE0 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 0 M4nCE0 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 0 M5nCE0 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 0 M0nCE0 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 0
0	GPIO8INCFG	0x0	RW	GPIO8 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.16CFGC Register

GPIO Configuration Register C (Pads 16-23)

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 655: 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 656: CFGC Register Bits

Bit	Name	Reset	RW	Description
31	GPIO23INTD	0x0	RW	GPIO23 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO23OUT-CFG	0x0	RW	GPIO23 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M4nCE0 = 0x3 - FNCSEL = 0x1 - IOM4 nCE, Channel 0
28	GPIO23INCFG	0x0	RW	GPIO23 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
27	GPIO22INTD	0x0	RW	GPIO22 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
26:25	GPIO22OUT-CFG	0x0	RW	GPIO22 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 3 M0nCE3 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 3
24	GPIO22INCFG	0x0	RW	GPIO22 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 656: CFGC Register Bits

Bit	Name	Reset	RW	Description
23	GPIO21INTD	0x0	RW	GPIO21 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
22:21	GPIO21OUT-CFG	0x0	RW	GPIO21 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 2 M2nCE2 = 0x3 - FNCSEL = 0x1 - IOM2 nCE, Channel 2
20	GPIO21INCFG	0x1	RW	GPIO21 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
19	GPIO20INTD	0x0	RW	GPIO20 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
18:17	GPIO20OUT-CFG	0x0	RW	GPIO20 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 1 M2nCE1 = 0x3 - FNCSEL = 0x1 - IOM2 nCE, Channel 1
16	GPIO20INCFG	0x1	RW	GPIO20 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 656: CFGC Register Bits

Bit	Name	Reset	RW	Description
15	GPIO19INTD	0x0	RW	GPIO19 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
14:13	GPIO19OUT-CFG	0x0	RW	GPIO19 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE3 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 3 M1nCE3 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 3 M2nCE3 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 3 MSPInCE0 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 0
12	GPIO19INCFG	0x0	RW	GPIO19 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
11	GPIO18INTD	0x0	RW	GPIO18 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
10:9	GPIO18OUT-CFG	0x0	RW	GPIO18 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 2 M3nCE2 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 2
8	GPIO18INCFG	0x0	RW	GPIO18 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 656: CFGC Register Bits

Bit	Name	Reset	RW	Description
7	GPIO17INTD	0x0	RW	GPIO17 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO17OUT-CFG	0x0	RW	GPIO17 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 1 M4nCE1 = 0x3 - FNCSEL = 0x1 - IOM4 nCE, Channel 1
4	GPIO17INCFG	0x0	RW	GPIO17 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
3	GPIO16INTD	0x0	RW	GPIO16 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO16OUT-CFG	0x0	RW	GPIO16 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M5nCE0 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 0
0	GPIO16INCFG	0x0	RW	GPIO16 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.17CFGD Register

GPIO Configuration Register D (Pads 24-31)

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 657: 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 658: CFGD Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31INTD	0x0	RW	GPIO31 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO31OUT-CFG	0x0	RW	GPIO31 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M4nCE0 = 0x3 - FNCSEL = 0x1 - IOM4 nCE, Channel 0
28	GPIO31INCFG	0x0	RW	GPIO31 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 658: CFGD Register Bits

Bit	Name	Reset	RW	Description
27	GPIO30INTD	0x0	RW	GPIO30 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
26:25	GPIO30OUTCFG	0x0	RW	GPIO30 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 3 M0nCE3 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 3
24	GPIO30INCFG	0x0	RW	GPIO30 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
23	GPIO29INTD	0x0	RW	GPIO29 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
22:21	GPIO29OUTCFG	0x0	RW	GPIO29 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 2 M1nCE2 = 0x3 - FNCSEL = 0x1 - IOM1 nCE, Channel 2
20	GPIO29INCFG	0x0	RW	GPIO29 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 658: CFGD Register Bits

Bit	Name	Reset	RW	Description
19	GPIO28INTD	0x0	RW	GPIO28 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
18:17	GPIO28OUTCFG	0x0	RW	GPIO28 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 1 MSPInCE0 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 0
16	GPIO28INCFG	0x0	RW	GPIO28 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
15	GPIO27INTD	0x0	RW	GPIO27 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
14:13	GPIO27OUTCFG	0x0	RW	GPIO27 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE0 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 0 M4nCE0 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 0 M5nCE0 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 0 M1nCE0 = 0x3 - FNCSEL = 0x1 - IOM1 nCE, Channel 0
12	GPIO27INCFG	0x0	RW	GPIO27 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 658: CFGD Register Bits

Bit	Name	Reset	RW	Description
11	GPIO26INTD	0x0	RW	GPIO26 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
10:9	GPIO26OUT-CFG	0x0	RW	GPIO26 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 3 M1nCE3 = 0x3 - FNCSEL = 0x1 - IOM1 nCE, Channel 3
8	GPIO26INCFG	0x0	RW	GPIO26 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
7	GPIO25INTD	0x0	RW	GPIO25 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO25OUT-CFG	0x0	RW	GPIO25 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 2 M0nCE2 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 2
4	GPIO25INCFG	0x0	RW	GPIO25 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 658: CFGD Register Bits

Bit	Name	Reset	RW	Description
3	GPIO24INTD	0x0	RW	GPIO24 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO24OUTCFG	0x0	RW	GPIO24 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 1 M5nCE1 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 1
0	GPIO24INCFG	0x0	RW	GPIO24 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.18CFGE Register

GPIO Configuration Register E (Pads 32-39)

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 659: 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 660: CFGE Register Bits

Bit	Name	Reset	RW	Description
31	GPIO39INTD	0x0	RW	GPIO39 interrupt direction. INTDIS = 0x0 - INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO39OUT-CFG	0x0	RW	GPIO39 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state
28	GPIO39INCFG	0x0	RW	GPIO39 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
27	GPIO38INTD	0x0	RW	GPIO38 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
26:25	GPIO38OUT-CFG	0x0	RW	GPIO38 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE3 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 3 M1nCE3 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 3 M2nCE3 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 3 M5nCE3 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 3
24	GPIO38INCFG	0x0	RW	GPIO38 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 660: CFGE Register Bits

Bit	Name	Reset	RW	Description
23	GPIO37INTD	0x0	RW	GPIO37 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
22:21	GPIO37OUT-CFG	0x0	RW	GPIO37 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE2 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 2 M4nCE2 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 2 M5nCE2 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 2 M0nCE2 = 0x3 - FNCSEL = 0x1 - IOM0 nCE, Channel 2
20	GPIO37INCFG	0x0	RW	GPIO37 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
19	GPIO36INTD	0x0	RW	GPIO36 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
18:17	GPIO36OUT-CFG	0x0	RW	GPIO36 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE1 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 1 M4nCE1 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 1 M5nCE1 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 1 MSPInCE1 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 1
16	GPIO36INCFG	0x0	RW	GPIO36 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 660: CFGE Register Bits

Bit	Name	Reset	RW	Description
15	GPIO35INTD	0x0	RW	GPIO35 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
14:13	GPIO35OUTCFG	0x0	RW	GPIO35 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M3nCE0 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 0
12	GPIO35INCFG	0x0	RW	GPIO35 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
11	GPIO34INTD	0x0	RW	GPIO34 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
10:9	GPIO34OUTCFG	0x0	RW	GPIO34 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE3 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 3 M1nCE3 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 3 M2nCE3 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 3 M3nCE3 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 3
8	GPIO34INCFG	0x0	RW	GPIO34 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 660: CFGE Register Bits

Bit	Name	Reset	RW	Description
7	GPIO33INTD	0x0	RW	GPIO33 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO33OUT-CFG	0x0	RW	GPIO33 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 2 M5nCE2 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 2
4	GPIO33INCFG	0x0	RW	GPIO33 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
3	GPIO32INTD	0x0	RW	GPIO32 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO32OUT-CFG	0x0	RW	GPIO32 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 1 MSPInCE1 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 1
0	GPIO32INCFG	0x0	RW	GPIO32 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.19CFGF Register

GPIO Configuration Register F (Pads 40 -47)

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 661: 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 662: CFGF Register Bits

Bit	Name	Reset	RW	Description
31	GPIO47INTD	0x0	RW	GPIO47 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
30:29	GPIO47OUT-CFG	0x0	RW	GPIO47 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 1 M3nCE1 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 1
28	GPIO47INCFG	0x0	RW	GPIO47 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 662: CFGF Register Bits

Bit	Name	Reset	RW	Description
27	GPIO46INTD	0x0	RW	GPIO46 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
26:25	GPIO46OUT-CFG	0x0	RW	GPIO46 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE0 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 0 M4nCE0 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 0 M5nCE0 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 0 MSPInCE1 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 1
24	GPIO46INCFG	0x0	RW	GPIO46 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
23	GPIO45INTD	0x0	RW	GPIO45 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
22:21	GPIO45OUT-CFG	0x0	RW	GPIO45 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M3nCE3 = 0x0 - FNCSEL = 0x1 - IOM3 nCE, Channel 3 M4nCE3 = 0x1 - FNCSEL = 0x1 - IOM4 nCE, Channel 3 M5nCE3 = 0x2 - FNCSEL = 0x1 - IOM5 nCE, Channel 3 M1nCE3 = 0x3 - FNCSEL = 0x1 - IOM1 nCE, Channel 3
20	GPIO45INCFG	0x0	RW	GPIO45 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 662: CFGF Register Bits

Bit	Name	Reset	RW	Description
19	GPIO44INTD	0x0	RW	GPIO44 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
18:17	GPIO44OUTCFG	0x0	RW	GPIO44 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 2 M5nCE2 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 2
16	GPIO44INCFG	0x0	RW	GPIO44 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
15	GPIO43INTD	0x0	RW	GPIO43 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
14:13	GPIO43OUTCFG	0x0	RW	GPIO43 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 1 MSPInCE1 = 0x3 - FNCSEL = 0x1 - MSPI nCE, Channel 1
12	GPIO43INCFG	0x0	RW	GPIO43 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 662: CFGF Register Bits

Bit	Name	Reset	RW	Description
11	GPIO42INTD	0x0	RW	GPIO42 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
10:9	GPIO42OUT-CFG	0x0	RW	GPIO42 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE0 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 0 M1nCE0 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 0 M2nCE0 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 0 M5nCE0 = 0x3 - FNCSEL = 0x1 - IOM5 nCE, Channel 0
8	GPIO42INCFG	0x0	RW	GPIO42 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
7	GPIO41INTD	0x0	RW	GPIO41 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x0 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x0 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x0, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x0, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x0, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x0, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO41OUT-CFG	0x0	RW	GPIO41 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE1 = 0x0 - FNCSEL = 0x0 - IOM0 nCE, Channel 1 M1nCE1 = 0x1 - FNCSEL = 0x0 - IOM1 nCE, Channel 1 M2nCE1 = 0x2 - FNCSEL = 0x0 - IOM2 nCE, Channel 1 MSPInCE1 = 0x3 - FNCSEL = 0x0 - MSPI nCE, Channel 1
4	GPIO41INCFG	0x0	RW	GPIO41 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

Table 662: CFGF Register Bits

Bit	Name	Reset	RW	Description
3	GPIO40INTD	0x0	RW	GPIO40 interrupt direction. INTDIS = 0x0 - INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO40OUT-CFG	0x0	RW	GPIO40 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state
0	GPIO40INCFG	0x0	RW	GPIO40 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.20CFGG Register

GPIO Configuration Register G (Pads 48-49)

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 663: 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 664: CFGG Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED

Table 664: CFGG Register Bits

Bit	Name	Reset	RW	Description
7	GPIO49INTD	0x0	RW	GPIO49 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
6:5	GPIO49OUTCFG	0x0	RW	GPIO49 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE3 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 3 M1nCE3 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 3 M2nCE3 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 3 M4nCE3 = 0x3 - FNCSEL = 0x1 - IOM4 nCE, Channel 3
4	GPIO49INCFG	0x0	RW	GPIO49 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data
3	GPIO48INTD	0x0	RW	GPIO48 interrupt direction. nCELOW = 0x0 - FNCSEL = 0x1 - nCE polarity active low nCEHIGH = 0x1 - FNCSEL = 0x1 - nCE polarity active high INTDIS = 0x0 - FNCSEL != 0x1, INCFG = 1 - No interrupt on GPIO transition INTBOTH = 0x1 - FNCSEL != 0x1, INCFG = 1 - Interrupt on either low to high or high to low GPIO transition INTLH = 0x0 - FNCSEL != 0x1, INCFG = 0 - Interrupt on low to high GPIO transition INTHL = 0x1 - FNCSEL != 0x1, INCFG = 0 - Interrupt on high to low GPIO transition
2:1	GPIO48OUTCFG	0x0	RW	GPIO48 output configuration. DIS = 0x0 - FNCSEL = 0x3 - Output disabled PUSHPULL = 0x1 - FNCSEL = 0x3 - Output is push-pull OD = 0x2 - FNCSEL = 0x3 - Output is open drain TS = 0x3 - FNCSEL = 0x3 - Output is tri-state M0nCE2 = 0x0 - FNCSEL = 0x1 - IOM0 nCE, Channel 2 M1nCE2 = 0x1 - FNCSEL = 0x1 - IOM1 nCE, Channel 2 M2nCE2 = 0x2 - FNCSEL = 0x1 - IOM2 nCE, Channel 2 M3nCE2 = 0x3 - FNCSEL = 0x1 - IOM3 nCE, Channel 2
0	GPIO48INCFG	0x0	RW	GPIO48 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - INTD = 0 - Readback will always be zero READEN = 0x1 - INTD = 1 - Read the GPIO pin data

11.7.2.21PADKEY Register

Key Register for all pad configuration registers

OFFSET: 0x00000060

INSTANCE 0 ADDRESS: 0x40010060

Key Register for all pad configuration registers

Table 665: 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 666: PADKEY Register Bits

Bit	Name	Reset	RW	Description
31:0	PADKEY	0x0	RW	Key register value. Key = 0x73 - Key

11.7.2.22RDA Register

GPIO Input Register A

OFFSET: 0x00000080

INSTANCE 0 ADDRESS: 0x40010080

GPIO Input Register A

Table 667: 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 668: RDA Register Bits

Bit	Name	Reset	RW	Description
31:0	RDA	0x0	RO	GPIO31-0 read data.

11.7.2.23RDB Register

GPIO Input Register B

OFFSET: 0x00000084

INSTANCE 0 ADDRESS: 0x40010084

GPIO Input Register B

Table 669: 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 670: RDB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	RDB	0x0	RO	GPIO49-32 read data.

11.7.2.24WTA Register

GPIO Output Register A

OFFSET: 0x00000088

INSTANCE 0 ADDRESS: 0x40010088

GPIO Output Register A

Table 671: 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 672: WTA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTA	0x0	RW	GPIO31-0 write data.

11.7.2.25WTB Register

GPIO Output Register B

OFFSET: 0x0000008C

INSTANCE 0 ADDRESS: 0x4001008C

GPIO Output Register B

Table 673: 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 674: WTB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	WTB	0x0	RW	GPIO49-32 write data.

11.7.2.26WTSA Register

GPIO Output Register A Set

OFFSET: 0x00000090

INSTANCE 0 ADDRESS: 0x40010090

GPIO Output Register A Set

Table 675: 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 676: WTSA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTSA	0x0	WO	Set the GPIO31-0 write data.

11.7.2.27WTSB Register

GPIO Output Register B Set

OFFSET: 0x00000094

INSTANCE 0 ADDRESS: 0x40010094

GPIO Output Register B Set

Table 677: 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 678: 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.

11.7.2.28WTCA Register

GPIO Output Register A Clear

OFFSET: 0x00000098

INSTANCE 0 ADDRESS: 0x40010098

GPIO Output Register A Clear

Table 679: 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 680: WTCA Register Bits

Bit	Name	Reset	RW	Description
31:0	WTCA	0x0	WO	Clear the GPIO31-0 write data.

11.7.2.29WTCB Register

GPIO Output Register B Clear

OFFSET: 0x0000009C

INSTANCE 0 ADDRESS: 0x4001009C

GPIO Output Register B Clear

Table 681: 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 682: 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.

11.7.2.30ENA Register

GPIO Enable Register A

OFFSET: 0x000000A0

INSTANCE 0 ADDRESS: 0x400100A0

GPIO Enable Register A

Table 683: 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 684: ENA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENA	0x0	RW	GPIO31-0 output enables

11.7.2.31ENB Register

GPIO Enable Register B

OFFSET: 0x000000A4

INSTANCE 0 ADDRESS: 0x400100A4

GPIO Enable Register B

Table 685: 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 686: ENB Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	ENB	0x0	RW	GPIO49-32 output enables

11.7.2.32ENSA Register

GPIO Enable Register A Set

OFFSET: 0x000000A8

INSTANCE 0 ADDRESS: 0x400100A8

GPIO Enable Register A Set

Table 687: 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 688: ENSA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENSA	0x0	RW	Set the GPIO31-0 output enables

11.7.2.33ENSB Register

GPIO Enable Register B Set

OFFSET: 0x000000AC

INSTANCE 0 ADDRESS: 0x400100AC

GPIO Enable Register B Set

Table 689: 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 690: 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

11.7.2.34ENCA Register

GPIO Enable Register A Clear

OFFSET: 0x000000B4

INSTANCE 0 ADDRESS: 0x400100B4

GPIO Enable Register A Clear

Table 691: 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 692: ENCA Register Bits

Bit	Name	Reset	RW	Description
31:0	ENCA	0x0	RW	Clear the GPIO31-0 output enables

11.7.2.35ENCB Register

GPIO Enable Register B Clear

OFFSET: 0x000000B8

INSTANCE 0 ADDRESS: 0x400100B8

GPIO Enable Register B Clear

Table 693: 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 694: 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

11.7.2.36STMRCAP Register

STIMER Capture Control

OFFSET: 0x000000BC

INSTANCE 0 ADDRESS: 0x400100BC

STIMER Capture trigger select and enable.

Table 695: 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 696: 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 696: STMRCAP Register Bits

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.

11.7.2.37IOM0IRQ Register

IOM0 Flow Control IRQ Select

OFFSET: 0x000000C0

INSTANCE 0 ADDRESS: 0x400100C0

IOMSTR0 IRQ select for flow control.

Table 697: 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 698: IOM0IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM0IRQ	0x3f	RW	IOMSTR0 IRQ pad select.

11.7.2.38IOM1IRQ Register

IOM1 Flow Control IRQ Select

OFFSET: 0x000000C4

INSTANCE 0 ADDRESS: 0x400100C4

IOMSTR1 IRQ select for flow control.

Table 699: 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 700: IOM1IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM1IRQ	0x3f	RW	IOMSTR1 IRQ pad select.

11.7.2.39IOM2IRQ Register

IOM2 Flow Control IRQ Select

OFFSET: 0x000000C8

INSTANCE 0 ADDRESS: 0x400100C8

IOMSTR2 IRQ select for flow control.

Table 701: 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 702: IOM2IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM2IRQ	0x3f	RW	IOMSTR2 IRQ pad select.

11.7.2.40IOM3IRQ Register

IOM3 Flow Control IRQ Select

OFFSET: 0x000000CC

INSTANCE 0 ADDRESS: 0x400100CC

IOMSTR3 IRQ select for flow control.

Table 703: 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 704: IOM3IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM3IRQ	0x3f	RW	IOMSTR3 IRQ pad select.

11.7.2.41IOM4IRQ Register

IOM4 Flow Control IRQ Select

OFFSET: 0x000000D0

INSTANCE 0 ADDRESS: 0x400100D0

IOMSTR4 IRQ select for flow control.

Table 705: 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 706: IOM4IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM4IRQ	0x3f	RW	IOMSTR4 IRQ pad select.

11.7.2.42IOM5IRQ Register

IOM5 Flow Control IRQ Select

OFFSET: 0x000000D4

INSTANCE 0 ADDRESS: 0x400100D4

IOMSTR5 IRQ select for flow control.

Table 707: 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 708: IOM5IRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	IOM5IRQ	0x3f	RW	IOMSTR5 IRQ pad select.

11.7.2.43BLEIFIRQ Register

BLEIF Flow Control IRQ Select

OFFSET: 0x000000D8

INSTANCE 0 ADDRESS: 0x400100D8

BLE IF IRQ select for flow control.

Table 709: BLEIFIRQ Register

31	30	29	28	27	26	25	24	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																										BLEIFIRQ					

Table 710: BLEIFIRQ Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	BLEIFIRQ	0x3f	RW	BLEIF IRQ pad select.

11.7.2.44GPIOOBS Register

GPIO Observation Mode Sample register

OFFSET: 0x000000DC

INSTANCE 0 ADDRESS: 0x400100DC

GPIO Observation mode sample register

Table 711: 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 712: 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 synronized to the read operation. Intended for debug purposes only

11.7.2.45ALTPADCFGA 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 713: 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 714: ALTPADCFGA Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED

Table 714: ALTPADCFGA Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.46ALTPADCFGB 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 715: 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 716: 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

Table 716: ALTPADCFGB Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.47ALTPADCFGC 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 717: 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 718: 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.

Table 718: ALTPADCFGC Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.48ALTPADCFGD 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 719: 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 720: 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

Table 720: ALTPADCFGD Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.49ALTPADCFGE 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 721: 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 722: 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

Table 722: ALTPADCFGE Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.50ALTPADCFGF 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 723: 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 724: 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.

Table 724: ALTPADCFGF Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.51ALTPADCFGG 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 725: 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 726: 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

Table 726: ALTPADCFGG Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.52ALTPADCFGH 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 727: 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 728: 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

Table 728: ALTPADCFGH Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.53ALTPADCFGI 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 729: 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 730: 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.

Table 730: ALTPADCFGI Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.54ALTPADCFGJ 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 731: 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 732: 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

Table 732: ALTPADCFGJ Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.55ALTPADCFGK 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 733: 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 734: 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

Table 734: ALTPADCFGK Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.56ALTPADCFGL 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 735: 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 736: 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.

Table 736: ALTPADCFGL Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.57ALTPADCFGM Register

Alternate Pad Configuration reg12 (Pads 49,48)

OFFSET: 0x00000110

INSTANCE 0 ADDRESS: 0x40010110

This register has additional configuration control for pads 49, 48

Table 737: 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 738: 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

Table 738: ALTPADCFGM Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.58SCDET Register

SCARD Card Detect select

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x40010114

Scard card detect select.

Table 739: SCDET Register

31	30	29	28	27	26	25	24	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																										SCDET					

Table 740: SCDET Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:0	SCDET	0x3f	RW	SCARD card detect pad select.

11.7.2.59CTENCFG Register

Counter/Timer Enable Config

OFFSET: 0x00000118

INSTANCE 0 ADDRESS: 0x40010118

Pad enable configuration.

Table 741: CTENCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0

Table 742: CTENCFG Register Bits

Bit	Name	Reset	RW	Description
31	EN31	0x1	RW	CT31 Enable DIS = 0x1 - Disable CT31 for output EN = 0x0 - Enable CT31 for output
30	EN30	0x1	RW	CT30 Enable DIS = 0x1 - Disable CT30 for output EN = 0x0 - Enable CT30 for output
29	EN29	0x1	RW	CT29 Enable DIS = 0x1 - Disable CT29 for output EN = 0x0 - Enable CT29 for output
28	EN28	0x1	RW	CT28 Enable DIS = 0x1 - Disable CT28 for output EN = 0x0 - Enable CT28 for output
27	EN27	0x1	RW	CT27 Enable DIS = 0x1 - Disable CT27 for output EN = 0x0 - Enable CT27 for output
26	EN26	0x1	RW	CT26 Enable DIS = 0x1 - Disable CT26 for output EN = 0x0 - Enable CT26 for output
25	EN25	0x1	RW	CT25 Enable DIS = 0x1 - Disable CT25 for output EN = 0x0 - Enable CT25 for output
24	EN24	0x1	RW	CT24 Enable DIS = 0x1 - Disable CT24 for output EN = 0x0 - Enable CT24 for output
23	EN23	0x1	RW	CT23 Enable DIS = 0x1 - Disable CT23 for output EN = 0x0 - Enable CT23 for output
22	EN22	0x1	RW	CT22 Enable DIS = 0x1 - Disable CT22 for output EN = 0x0 - Enable CT22 for output

Table 742: CTENCFG Register Bits

Bit	Name	Reset	RW	Description
21	EN21	0x1	RW	CT21 Enable DIS = 0x1 - Disable CT21 for output EN = 0x0 - Enable CT21 for output
20	EN20	0x1	RW	CT20 Enable DIS = 0x1 - Disable CT20 for output EN = 0x0 - Enable CT20 for output
19	EN19	0x1	RW	CT19 Enable DIS = 0x1 - Disable CT19 for output EN = 0x0 - Enable CT19 for output
18	EN18	0x1	RW	CT18 Enable DIS = 0x1 - Disable CT18 for output EN = 0x0 - Enable CT18 for output
17	EN17	0x1	RW	CT17 Enable DIS = 0x1 - Disable CT17 for output EN = 0x0 - Enable CT17 for output
16	EN16	0x1	RW	CT16 Enable DIS = 0x1 - Disable CT16 for output EN = 0x0 - Enable CT16 for output
15	EN15	0x1	RW	CT15 Enable DIS = 0x1 - Disable CT15 for output EN = 0x0 - Enable CT15 for output
14	EN14	0x1	RW	CT14 Enable DIS = 0x1 - Disable CT14 for output EN = 0x0 - Enable CT14 for output
13	EN13	0x1	RW	CT13 Enable DIS = 0x1 - Disable CT13 for output EN = 0x0 - Enable CT13 for output
12	EN12	0x1	RW	CT12 Enable DIS = 0x1 - Disable CT12 for output EN = 0x0 - Enable CT12 for output
11	EN11	0x1	RW	CT11 Enable DIS = 0x1 - Disable CT11 for output EN = 0x0 - Enable CT11 for output
10	EN10	0x1	RW	CT10 Enable DIS = 0x1 - Disable CT10 for output EN = 0x0 - Enable CT10 for output
9	EN9	0x1	RW	CT9 Enable DIS = 0x0 - Disable CT9 for output EN = 0x0 - Enable CT9 for output

Table 742: CTENCFG Register Bits

Bit	Name	Reset	RW	Description
8	EN8	0x1	RW	CT8 Enable DIS = 0x1 - Disable CT8 for output EN = 0x0 - Enable CT8 for output
7	EN7	0x1	RW	CT7 Enable DIS = 0x1 - Disable CT7 for output EN = 0x0 - Enable CT7 for output
6	EN6	0x1	RW	CT6 Enable DIS = 0x1 - Disable CT6 for output EN = 0x0 - Enable CT6 for output
5	EN5	0x1	RW	CT5 Enable DIS = 0x1 - Disable CT5 for output EN = 0x0 - Enable CT5 for output
4	EN4	0x1	RW	CT4 Enable DIS = 0x1 - Disable CT4 for output EN = 0x0 - Enable CT4 for output
3	EN3	0x1	RW	CT3 Enable DIS = 0x1 - Disable CT3 for output EN = 0x0 - Enable CT3 for output
2	EN2	0x1	RW	CT2 Enable DIS = 0x1 - Disable CT2 for output EN = 0x0 - Enable CT2 for output
1	EN1	0x1	RW	CT1 Enable DIS = 0x1 - Disable CT1 for output EN = 0x0 - Enable CT1 for output
0	EN0	0x1	RW	CT0 Enable DIS = 0x1 - Disable CT0 for output EN = 0x0 - Enable CT0 for output

11.7.2.60INT0EN 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 743: 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 744: 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.
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.

Table 744: INT0EN Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.61INT0STAT 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 745: 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 746: 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.
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.

Table 746: INT0STAT Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.62INT0CLR 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 747: 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 748: INT0CLR Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31	0x0	RW	GPIO31 interrupt.
30	GPIO30	0x0	RW	GPIO30 interrupt.

Table 748: INT0CLR Register Bits

Bit	Name	Reset	RW	Description
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.

Table 748: INT0CLR Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.63INT0SET 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 749: 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 750: 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.

Table 750: INT0SET Register Bits

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.

Table 750: INT0SET Register Bits

Bit	Name	Reset	RW	Description
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

11.7.2.64INT1EN 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 751: 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 752: 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.

Table 752: INT1EN Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.65INT1STAT 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 753: 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 754: INT1STAT Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED

Table 754: INT1STAT Register Bits

Bit	Name	Reset	RW	Description
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.

11.7.2.66INT1CLR 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 755: 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 756: 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.

Table 756: INT1CLR Register Bits

Bit	Name	Reset	RW	Description
0	GPIO32	0x0	RW	GPIO32 interrupt.

11.7.2.67INT1SET 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 757: 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 758: 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.
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.

Table 758: INT1SET Register Bits

Bit	Name	Reset	RW	Description
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.

12. Clock Generator and Real Time Clock Module

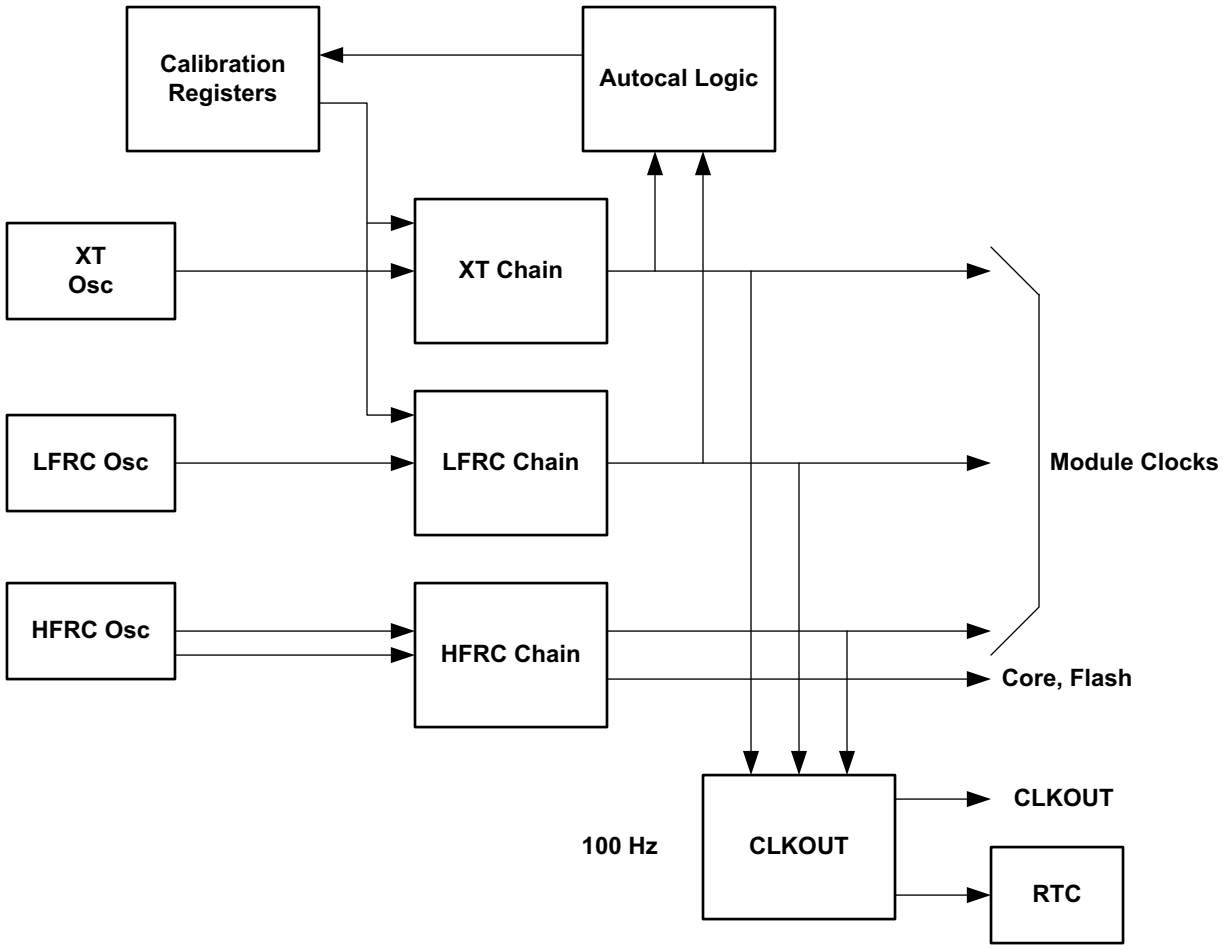


Figure 63. Block diagram for the Clock Generator and Real Time Clock Module

12.1 Clock Generator

12.1.1 Functional Overview

A high-level view of the Clock Generator Module, which supplies all clocks required by the Apollo3 Blue MCU, is shown in Figure 63. 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 48/96 MHz oscillator (HFRC).

A clock, CLKOUT, generated from any of the oscillators, may be configured and driven onto an external pin. 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.

Figure 64 shows the available clock sources, divisors and distribution to the various modules of the Apollo3 Blue MCU.

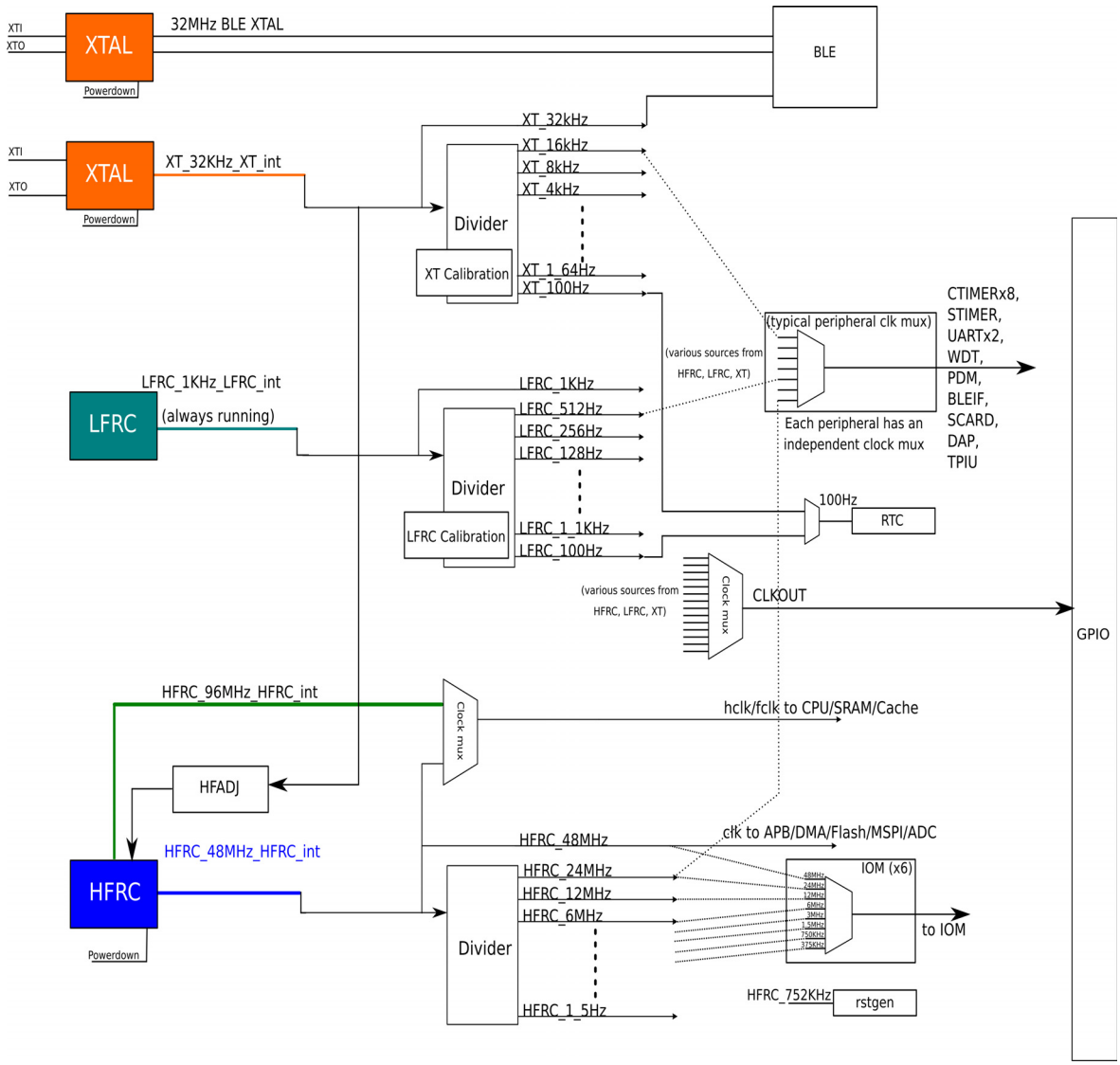


Figure 64. Apollo3 Blue Clock Tree

12.1.2 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 SIMO buck regulator in low power mode (32kHz) as well as some basic state machines and is always enabled. Calibration logic is included.

12.1.2.1 LFRC Oscillator Digital Calibration

The LFRC Oscillator includes a patented Distributed Digital Calibration function similar to that of the XT Oscillator (Section 12.1.3.2). 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%). This is enabled in the CALRC register, which is 18 bits wide. The most significant bit is the "sign" bit. A '1' on bit 17 would mean a subtraction, and a '0' would mean an addition. Bits 16 to 0 would be the number of cycles to be added to or subtracted from the 512Hz LFRC clock across a 1024 second period. The range of clocks that can be added or subtracted range from -131072 to 131071.

The pulses which are added to or subtracted from the 512 Hz clock are spread evenly over each 1024 second period using the Ambiq Micro 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 Apollo3 Blue MCU which use the LFRC.

Note that 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.

12.1.2.2 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 -131072 to 131071
9. If the adjustment value falls between these two values, set CALRC = Adj
10. Otherwise, the LFRC frequency is too low or too high to be calibrated

12.1.3 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 32.768kHz. This is handled within the Apollo3 Blue MCU.

NOTE: The XTAL is highly sensitive to external leakage on the XI pin. Therefore it is recommended to minimize the components on XI and to use extremely low leakage load capacitors.

12.1.3.1 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 Apollo3 Blue MCU which use the XT.

Note that 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.

12.1.3.2 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/(1000000/2^{19}) = 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.

If the 32 kHz XT Oscillator generates clocks at less than 8 kHz for a period of more than 32 ms, the Apollo3 Blue MCU detects an Oscillator Failure. The Oscillator Fail (OF) flag is set when an Oscillator Failure occurs, and is also set when the Apollo3 Blue MCU 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU 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.

12.1.4 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 Apollo3 Blue MCU 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. The Apollo3 Blue MCU supports high frequency TurboSPOT burst mode. The HFRC supplies both the 96MHz as well as the 48MHz clocks to support the high frequency core/ memory domains and the stable 48MHz clock for the remaining logic/IO controllers.

12.1.5 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 Apollo3 Blue MCU 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 21.5 on page 778.

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.

The following steps are recommended to enable the HFADJ functionality.

1. Write "0x47" to the CLKKEY register to enable access to CLK_GEN registers
2. Set the HFADJCK field in HFADJ register to set the target HFRC adjustment period. It can range from 4 seconds (0x0) to 1024 seconds (0x7).
3. Set the gain for the adjustment through HFADJGAIN field in HFADJ register.
4. Set the HFWARMUP field if XT is STOP (through the OCTRL register STOPXT field).
5. Enable HFADJEN field in HFADJ register to start the adjustment cycle.

12.1.6 Burst Mode Support

The Apollo3 Blue MCU supports the TurboSPOT burst operating mode. Under burst mode, the core clock runs at 96MHz. Burst mode is initiated when the BURSTREQ bit in REG_CLK_GEN_FREQCTRL is written with a '1'. Once the burst mode is available, the BURSTACK and BURSTSTATUS bit in REG_CLK_GEN_FREQCTRL register are updated.

When burst mode is no longer required, software will write a '0' to the BURSTREQ bit in REG_CLK_GEN_FREQCTRL. The BURSTSTATUS will be updated immediately, while the BURSTACK will only be updated once the nominal operating mode is available.

The following steps are necessary for enabling burst mode:

(Note that Burst can only be enabled if Burst is allowed - indicated via the SKU register)

1. This can be checked with REG_MCU_CTRL_SKU register in MCU_CTRL - ALLOWBURST bit
2. Set the BURSTREQ bit in REG_MCU_CTRL_FEATUREENABLE
3. Write "0x47" to the CLKKEY register to enable access to CLK_GEN registers
4. Set BURSTREQ bit in REG_CLK_GEN_FREQCTRL
5. Poll BURSTSTATUS bit and BURSTACK bit in REG_CLK_GEN_FREQCTRL register

   OR

   Wait for Event trigger (enabled through BURSTFEATUREEVEN bit in REG_PWRCTRL_DEVP-WREVENTEN register)

NOTE

In Burst Mode on the Apollo3 Blue MCU, the SYSTICK increments at twice the normal (48 MHz) clock rate. Some RTOSes may use SYSTICK for scheduler timing by default, in which case scheduler event timing will be wrong when using Burst Mode. It is recommended not to use SYSTICK and Burst Mode together unless proper compensation is made, e.g., ignoring every other SYSTICK interrupt during Burst Mode.

12.1.7 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.

12.1.8 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.

12.2 CLKGEN Registers

Clock Generator

INSTANCE 0 BASE ADDRESS:0x40004000

This is the register bank for the clock generator registers. It includes the RTC unit and the register control for BLE Ton Adjust unit

12.2.1 Register Memory Map

Table 759: CLKGEN Register Map

Address(s)	Register 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
0x40004014	CLKKEY	Key Register for Clock Control Register
0x40004018	CCTRL	HFRC Clock Control
0x4000401C	STATUS	Clock Generator Status
0x40004020	HFADJ	HFRC Adjustment
0x40004028	CLOCKENSTAT	Clock Enable Status
0x4000402C	CLOCKEN2STAT	Clock Enable Status
0x40004030	CLOCKEN3STAT	Clock Enable Status
0x40004034	FREQCTRL	HFRC Frequency Control register
0x4000403C	BLEBUCKTONADJ	BLE BUCK TON ADJUST
0x40004100	INTRPTEN	CLKGEN Interrupt Register: Enable
0x40004104	INTRPTSTAT	CLKGEN Interrupt Register: Status
0x40004108	INTRPTCLR	CLKGEN Interrupt Register: Clear
0x4000410C	INTRPTSET	CLKGEN Interrupt Register: Set

12.2.2 CLKGEN Registers

12.2.2.1 CALXT Register

XT Oscillator Control

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40004000

This is the XT Oscillator Calibration value. This value allows any derived XT clocks to be "calibrated". This means that the original 32KHz version of XT will not be changed, but a 16KHz version (divided down version) can be modified. This register value will add or subtract the number of cycles programmed in this register across a 32 seconds interval. For example, if a value of 100 is programmed in this register, then 100 additional clock cycles will be added into a 16KHz clock period across a 32 second interval.

Table 760: 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 761: CALXT Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:0	CALXT	0x0	RW	XT Oscillator calibration value. This register will enable the hardware to increase or decrease the number of cycles in a 16KHz clock derived from the original 32KHz version. The most significant bit is the sign. A '1' is a reduction, and a '0' is an addition. This calibration value will add or reduce the number of cycles programmed here across a 32 second interval. The maximum value that is effective is from -1024 to 1023.

12.2.2.2 CALRC Register

RC Oscillator Control

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40004004

This is the LFRC Calibration value. Similar to the XT calibration, it allows the derived LFRC clock to be calibrated. The original 1024Hz clock source will not change, but a 512Hz version (divided down version) can be modified. This register will add or subtract the number of cycles programmed in this register across a 1024 seconds interval. For example, if a value of 200 is programmed in this register, then 200 additional clocks will be added into the 512Hz derived clock across a 1024 seconds interval.

Table 762: 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 763: CALRC Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	CALRC	0x0	RW	LFRC Oscillator calibration value. This register will enable the hardware to increase or decrease the number of cycles in a 512 Hz clock derived from the original 1024 version. The most significant bit is the sign. A '1' is a reduction, and a '0' is an addition. This calibration value will add or reduce the number of cycles programmed here across a 32 second interval. The range is from -131072 (decimal) to 131071 (decimal). This register is normally used in conjuction with ACALCTR register. The CALRC register will load the ACALCTR register (bits 17:0) if the ACALCTR register is set to measure the LFRC with the XT clock.

12.2.2.3 ACALCTR Register

Autocalibration Counter

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40004008

This register can be used for 2 purposes. The first is to calibrate the LFRC clock using the XT clock source. The second is to measure an internal clock signal relative to the external clock. In that case, the ACALCTR will show the multiple of the external clock with respect to the internal clock signal. E.g. Fref = Fmeas x ACALCTR. Note that this register should not be confused with the HFRC Adjustment register, which is separately defined in CLKGEN_HFADJ register.

Table 764: 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 765: ACALCTR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED

Table 765: ACALCTR Register Bits

Bit	Name	Reset	RW	Description
23:0	ACALCTR	0x0	RO	Autocalibration Counter result. Bits 17 down to 0 of this is feed directly to the CALRC register if ACAL register in OCTRL register is set to 1024SEC or 512SEC.

12.2.2.4 OCTRL Register

Oscillator Control

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4000400C

This register includes controls for autocalibration in addition to the RTC oscillator controls.

Table 766: 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 767: OCTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:8	ACAL	0x0	RW	Autocalibration control. This selects the source to be used in the autocalibration flow. This flow can also be used to measure an internal clock against an external clock source, with the external clock normally used as the reference. DIS = 0x0 - Disable Autocalibration 1024SEC = 0x2 - Autocalibrate every 1024 seconds. Once autocalibration is done, an interrupt will be triggered at the end of 1024 seconds. 512SEC = 0x3 - Autocalibrate every 512 seconds. Once autocalibration is done, an interrupt will be trigged at the end of 512 seconds. XTFREQ = 0x6 - Frequency measurement using XT. The XT clock is normally considered much more accurate than the LFRC clock source. EXTFREQ = 0x7 - Frequency measurement using external clock.
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. If this is set, then LFRC clock source will switch from XT to RC. DIS = 0x0 - Disable the oscillator switch on failure function. EN = 0x1 - Enable the oscillator switch on failure function.

Table 767: OCTRL Register Bits

Bit	Name	Reset	RW	Description
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

12.2.2.5 CLKOUT Register

CLKOUT Frequency Select

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x40004010

This register enables the CLKOUT to the GPIOs, and selects the clock source to that.

Table 768: 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 769: 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
6	RSVD	0x0	RO	RESERVED

Table 769: CLKOUT Register Bits

Bit	Name	Reset	RW	Description
5:0	CKSEL	0x0	RW	CLKOUT signal select 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 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 LFRC_DIV1M = 0x31 - LFRC / 2^20 XTNE = 0x35 - XT (not autoenabled) XTNE_DIV16 = 0x36 - XT / 16 (not autoenabled) LFRCNE_DIV32 = 0x37 - LFRC / 32 (not autoenabled) LFRCNE = 0x39 - LFRC (not autoenabled) - Default for undefined values

12.2.2.6 CLKKEY Register

Key Register for Clock Control Register

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40004014

Key Register for Clock Control Register

Table 770: CLKKEY Register

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

Table 771: CLKKEY Register Bits

Bit	Name	Reset	RW	Description
31:0	CLKKEY	0x0	RW	Key register value. Key = 0x47 - Key

12.2.2.7 CCTRL Register

HFRC Clock Control

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40004018

This register controls the main divider for HFRC clock. If this is set, all internal HFRC clock sources are divided by 2.

Table 772: 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																															CORE-

Table 773: 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

12.2.2.8 STATUS Register

Clock Generator Status

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4000401C

This register provides status to the XT oscillator and the source of the RTC.

Table 774: 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 775: 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). After an RTC oscillator change, it may take up to 2 seconds for this field to reflect the new oscillator.

12.2.2.9 HFADJ Register

HFRC Adjustment

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40004020

This register controls the HFRC adjustment. The HFRC clock can change with temperature and process corners, and this register controls the HFRC adjustment logic which reduces the fluctuations to the clock.

Table 776: 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								HFADJGAIN			HFWARMUP	HFXTADJ												RSVD				HFADJCK			HFADJEN

Table 777: HFADJ Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED

Table 777: HFADJ Register Bits

Bit	Name	Reset	RW	Description
23:21	HFADJGAIN	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 warmup period for HFRC adjustment 1SEC = 0x0 - Autoadjust XT warmup period = 1-2 seconds 2SEC = 0x1 - Autoadjust XT warmup 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 - Autoadjust repeat period = 4 seconds 16SEC = 0x1 - Autoadjust repeat period = 16 seconds 32SEC = 0x2 - Autoadjust repeat period = 32 seconds 64SEC = 0x3 - Autoadjust repeat period = 64 seconds 128SEC = 0x4 - Autoadjust repeat period = 128 seconds 256SEC = 0x5 - Autoadjust repeat period = 256 seconds 512SEC = 0x6 - Autoadjust repeat period = 512 seconds 1024SEC = 0x7 - Autoadjust repeat period = 1024 seconds
0	HFADJEN	0x0	RW	HFRC adjustment control DIS = 0x0 - Disable the HFRC adjustment EN = 0x1 - Enable the HFRC adjustment

12.2.2.10CLOCKENSTAT Register

Clock Enable Status

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x40004028

This register provides the enable status to all the peripheral clocks.

Table 778: CLOCKENSTAT Register

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

Table 779: CLOCKENSTAT Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCKENSTAT	0x0	RO	Clock enable status ADC_CLKEN = 0x1 - Clock enable for the ADC. APBDMA_ACTIVITY_CLKEN = 0x2 - Clock enable for the APBDMA ACTIVITY APBDMA_AOH_CLKEN = 0x4 - Clock enable for the APBDMA AOH DOMAIN APBDMA_AOL_CLKEN = 0x8 - Clock enable for the APBDMA AOL DOMAIN APBDMA_APB_CLKEN = 0x10 - Clock enable for the APBDMA_APB APBDMA_BLEL_CLKEN = 0x20 - Clock enable for the APBDMA_BLEL APBDMA_HCPA_CLKEN = 0x40 - Clock enable for the APBDMA_HCPA APBDMA_HCPB_CLKEN = 0x80 - Clock enable for the APBDMA_HCPB APBDMA_HCPC_CLKEN = 0x100 - Clock enable for the APBDMA_HCPC APBDMA_MSPI_CLKEN = 0x200 - Clock enable for the APBDMA_MSPI APBDMA_PDM_CLKEN = 0x400 - Clock enable for the APBDMA_PDM BLEIF_CLK_CLKEN = 0x800 - Clock enable for the BLEIF BLEIF_CLK32K_CLKEN = 0x1000 - Clock enable for the BLEIF 32khZ CLOCK CTIMER_CLKEN = 0x2000 - Clock enable for the CTIMER BLOCK CTIMER0A_CLKEN = 0x4000 - Clock enable for the CTIMER0A CTIMER0B_CLKEN = 0x8000 - Clock enable for the CTIMER0B CTIMER1A_CLKEN = 0x10000 - Clock enable for the CTIMER1A CTIMER1B_CLKEN = 0x20000 - Clock enable for the CTIMER1B CTIMER2A_CLKEN = 0x40000 - Clock enable for the CTIMER2A CTIMER2B_CLKEN = 0x80000 - Clock enable for the CTIMER2B CTIMER3A_CLKEN = 0x100000 - Clock enable for the CTIMER3A CTIMER3B_CLKEN = 0x200000 - Clock enable for the CTIMER3B CTIMER4A_CLKEN = 0x400000 - Clock enable for the CTIMER4A CTIMER4B_CLKEN = 0x800000 - Clock enable for the CTIMER4B CTIMER5A_CLKEN = 0x1000000 - Clock enable for the CTIMER5A CTIMER5B_CLKEN = 0x2000000 - Clock enable for the CTIMER5B CTIMER6A_CLKEN = 0x4000000 - Clock enable for the CTIMER6A CTIMER6B_CLKEN = 0x8000000 - Clock enable for the CTIMER6B CTIMER7A_CLKEN = 0x10000000 - Clock enable for the CTIMER7A CTIMER7B_CLKEN = 0x20000000 - Clock enable for the CTIMER7B DAP_CLKEN = 0x40000000 - Clock enable for the DAP IOMSTRIFC0_CLKEN = 0x80000000 - Clock enable for the IOMSTRIFC0

12.2.2.11CLOCKEN2STAT Register

Clock Enable Status

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4000402C

This is a continuation of the clock enable status.

Table 780: CLOCKEN2STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CLOCKEN2STAT																															

Table 781: CLOCKEN2STAT Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCK-EN2STAT	0x0	RO	Clock enable status 2 IOMSTRIFC1_CLKEN = 0x1 - Clock enable for the IO MASTER 1 IFC INTERFACE IOMSTRIFC2_CLKEN = 0x2 - Clock enable for the IO MASTER 2 IFC INTERFACE IOMSTRIFC3_CLKEN = 0x4 - Clock enable for the IO MASTER 3 IFC INTERFACE IOMSTRIFC4_CLKEN = 0x8 - Clock enable for the IO MASTER 4 IFC INTERFACE IOMSTRIFC5_CLKEN = 0x10 - Clock enable for the IO MASTER 5 IFC INTERFACE PDM_CLKEN = 0x20 - Clock enable for the PDM PDMIFC_CLKEN = 0x40 - Clock enable for the PDM INTERFACE PWRCTRL_CLKEN = 0x80 - Clock enable for the PWRCTRL PWRCTRL_COUNT_CLKEN = 0x100 - Clock enable for the PWRCTRL counter RSTGEN_CLKEN = 0x200 - Clock enable for the RSTGEN SCARD_CLKEN = 0x400 - Clock enable for the SCARD SCARD_ALTAPB_CLKEN = 0x800 - Clock enable for the SCARD ALTAPB STIMER_CNT_CLKEN = 0x1000 - Clock enable for the STIMER_CNT_-CLKEN TPIU_CLKEN = 0x2000 - Clock enable for the TPIU_CLKEN UART0HF_CLKEN = 0x4000 - Clock enable for the UART0 HF UART1HF_CLKEN = 0x8000 - Clock enable for the UART1 HF WDT_CLKEN = 0x8000 - Clock enable for the Watchdog timer XT_32KHZ_EN = 0x40000000 - Clock enable for the XT 32KHZ FORCEHFRC = 0x80000000 - HFRC is forced on Status.

12.2.2.12CLOCKEN3STAT Register

Clock Enable Status

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x40004030

This is a continuation of the clock enable status.

Table 782: CLOCKEN3STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CLOCKEN3STAT																															

Table 783: CLOCKEN3STAT Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCK-EN3STAT	0x0	RO	Clock enable status 3 DAP_enabled = 0x20000 - DAP clock is enabled [17] VCOMP_enabled = 0x40000 - VCOMP powerdown indicator [18] XTAL_enabled = 0x1000000 - XTAL is enabled [24] HFRC_enabled = 0x2000000 - HFRC is enabled [25] HFADJEN = 0x4000000 - HFRC Adjust enabled [26] HFRC_en_out = 0x8000000 - HFRC Enabled out [27] RTC_XT = 0x10000000 - RTC use XT [28] clkout_xtal_en = 0x20000000 - XTAL clkout enabled [29] clkout_hfrc_en = 0x40000000 - HFRC clkout enabled [30] flashclk_en = 0x80000000 - Flash clk is enabled [31]

12.2.2.13FREQCTRL Register

HFRC Frequency Control register

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x40004034

This register provides the burst control and burst status.

Table 784: FREQCTRL Register

31	30	29	28	27	26	25	24	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																													BURSTSTATUS	BURSTACK	BURSTREQ

Table 785: FREQCTRL Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	BURSTSTATUS	0x0	RO	This represents frequency burst status.
1	BURSTACK	0x0	RO	Frequency Burst Request Acknowledge. Frequency burst requested is always acknowledged whether burst is granted or not depending on feature enable.
0	BURSTREQ	0x0	RW	Frequency Burst Enable Request DIS = 0x0 - Frequency for ARM core stays at 48MHz EN = 0x1 - Frequency for ARM core is increased to 96MHz

12.2.2.14BLEBUCKTONADJ Register

BLE BUCK TON ADJUST

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0x4000403C

This is the register control for BLE ton adjustment logic.

Table 786: BLEBUCKTONADJ Register

31	30	29	28	27	26	25	24	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				ZEROLENDETECTEN	ZEROLENDETECTTRIM				TONADJUSTEN	TONADJUSTPERIOD		TONHIGHTHRESHOLD										TONLOWTHRESHOLD										

Table 787: BLEBUCKTONADJ Register Bits

Bit	Name	Reset	RW	Description
31:28	RSVD	0x0	RO	RESERVED
27	ZEROLENDE-TECTEN	0x0	RW	BLEBUCK ZERO LENGTH DETECT ENABLE DIS = 0x0 - Disable Zero Length Detect EN = 0x1 - Enable Zero Length Detect

Table 787: BLEBUCKTONADJ Register Bits

Bit	Name	Reset	RW	Description
26:23	ZEROLENDE-TECTTRIM	0x0	RW	BLEBUCK ZERO LENGTH DETECT TRIM SetF = 0xF - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 81us (10 percent margin of error) or more SetE = 0xE - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 75.6us (10 percent margin of error) or more SetD = 0xD - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 70.2us (10 percent margin of error) or more SetC = 0xC - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 64.8us (10 percent margin of error) or more SetB = 0xB - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 59.4us (10 percent margin of error) or more SetA = 0xA - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 54.0us (10 percent margin of error) or more Set9 = 0x9 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 48.6us (10 percent margin of error) or more Set8 = 0x8 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 43.2us (10 percent margin of error) or more Set7 = 0x7 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 37.8us (10 percent margin of error) or more Set6 = 0x6 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 32.4us (10 percent margin of error) or more Set5 = 0x5 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 27.0us (10 percent margin of error) or more Set4 = 0x4 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 21.6us (10 percent margin of error) or more Set3 = 0x3 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 16.2us (10 percent margin of error) or more Set2 = 0x2 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 10.8us (10 percent margin of error) or more Set1 = 0x1 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 5.4us (10 percent margin of error) or more Set0 = 0x0 - Indicator send when the BLE BUCK asserts blebuck_comp1 for about 2.0us (10 percent margin of error) or more
22	TONADJUSTEN	0x0	RW	TON ADJUST ENABLE DIS = 0x0 - Disable Adjust for BLE BUCK TON trim EN = 0x1 - Enable Adjust for BLE BUCK TON trim
21:20	TONADJUST-PERIOD	0x0	RW	TON ADJUST PERIOD HFRC_94KHz = 0x0 - Adjust done for every 1 94KHz period
19:10	TONHIGH-THRESHOLD	0x0	RW	TON ADJUST HIGH THRESHOLD. Suggested values are #15(94KHz) #2A(47Khz) #A6(12Khz) #29A(3Khz)
9:0	TONLOW-THRESHOLD	0x0	RW	TON ADJUST LOW THRESHOLD. Suggested values are #A(94KHz) #15(47KHz) #53(12Khz) #14D(3Khz)

12.2.2.15INTRPTEN 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 788: INTRPTEN Register

31	30	29	28	27	26	25	24	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																													OF	ACC	ACF

Table 789: INTRPTEN Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

12.2.2.16INTRPTSTAT 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 790: INTRPTSTAT Register

31	30	29	28	27	26	25	24	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																													OF	ACC	ACF

Table 791: INTRPTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt

Table 791: INTRPTSTAT Register Bits

Bit	Name	Reset	RW	Description
0	ACF	0x0	RW	Autocalibration Fail interrupt

12.2.2.17INTRPTCLR 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 792: INTRPTCLR Register

31	30	29	28	27	26	25	24	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																													OF	ACC	ACF

Table 793: INTRPTCLR Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

12.2.2.18INTRPTSET 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 794: INTRPTSET Register

31	30	29	28	27	26	25	24	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																													OF	ACC	ACF

Table 795: INTRPTSET Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED
2	OF	0x0	RW	XT Oscillator Fail interrupt
1	ACC	0x0	RW	Autocalibration Complete interrupt
0	ACF	0x0	RW	Autocalibration Fail interrupt

12.3 Real Time Clock

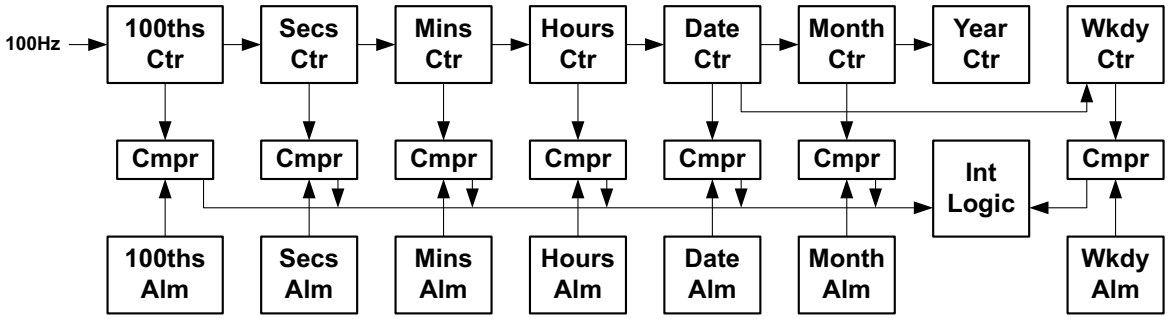


Figure 65. Block diagram for the Real Time Clock Module

12.3.1 RTC Functional Overview

The Real Time Clock (RTC) Module, shown in Figure 65, 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.

12.3.2 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.

12.3.3 Calendar Counter Reads

The RTC includes special logic to help insure that the Calendar Counters may be read reliably, i.e. 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.

12.3.4 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 796: 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.

12.3.5 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.

12.3.6 Century Control and Leap Year Management

The REG_CLK_GEN_CTRUP_CB bit indicates the current century. A value of 0 indicates the 20th century, and a value of 1 indicates the 19th or 21st 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.

12.3.7 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.

12.4 RTC Registers

Real Time Clock

INSTANCE 0 BASE ADDRESS:0x40004200

12.4.1 Register Memory Map

Table 797: RTC Register Map

Address(s)	Register Name	Description
0x40004240	CTRLOW	RTC Counters Lower
0x40004244	CTRUP	RTC Counters Upper
0x40004248	ALMLOW	RTC Alarms Lower
0x4000424C	ALMUP	RTC Alarms Upper
0x40004250	RTCCTL	RTC Control Register
0x40004300	INTEN	RTC Interrupt Register: Enable
0x40004304	INTSTAT	RTC Interrupt Register: Status
0x40004308	INTCLR	RTC Interrupt Register: Clear
0x4000430C	INTSET	RTC Interrupt Register: Set

12.4.2 RTC Registers

12.4.2.1 CTRLOW Register

RTC Counters Lower

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x40004240

This counter contains the values for hour, minutes, seconds and 100ths of a second Counter.

Table 798: 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 799: 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
14:8	CTRSEC	0x0	RW	Seconds Counter
7:0	CTR100	0x0	RW	100ths of a second Counter

12.4.2.2 CTRUP Register

RTC Counters Upper

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0x40004244

This register contains the day, month and year information. It contains which day in the week, and the century as well. The information of the century can also be derived from the year information. The 31st bit contains the error bit. See description in the register bit for condition when error is triggered.

Table 800: 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 801: CTRUP Register Bits

Bit	Name	Reset	RW	Description
31	CTERR	0x0	RO	Counter read error status. Error is triggered when software reads the lower word of the counters, and fails to read the upper counter within 1/100 second. This is because when the lower counter is read, the upper counter is held off from incrementing until it is read so that the full time stamp can be read. 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

12.4.2.3 ALMLOW Register

RTC Alarms Lower

OFFSET: 0x00000048

INSTANCE 0 ADDRESS: 0x40004248

This register is the Alarm settings for hours, minutes, second and 1/100th seconds settings.

Table 802: 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 803: 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

12.4.2.4 ALMUP Register

RTC Alarms Upper

OFFSET: 0x0000004C

INSTANCE 0 ADDRESS: 0x4000424C

This register is the alarm settings for week, month and day.

Table 804: 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 805: 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

12.4.2.5 RTCCTL Register

RTC Control Register

OFFSET: 0x00000050

INSTANCE 0 ADDRESS: 0x40004250

This is the register control for the RTC module. It sets the 12 or 24 hours mode, enables counter writes and sets the alarm repeat interval.

Table 806: 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 807: 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

Table 807: RTCCTL Register Bits

Bit	Name	Reset	RW	Description
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

12.4.2.6 INTEN Register

RTC Interrupt Register: Enable

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x40004300

Set bits in this register to allow this module to generate the corresponding interrupt.

Table 808: 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

Table 809: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	ALM	0x0	RW	RTC Alarm interrupt

12.4.2.7 INTSTAT Register

RTC Interrupt Register: Status

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x40004304

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

Table 810: 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

Table 811: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	ALM	0x0	RW	RTC Alarm interrupt

12.4.2.8 INTCLR Register

RTC Interrupt Register: Clear

OFFSET: 0x00000108

INSTANCE 0 ADDRESS: 0x40004308

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

Table 812: 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

Table 813: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	ALM	0x0	RW	RTC Alarm interrupt

12.4.2.9 INTSET Register

RTC Interrupt Register: Set

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x4000430C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 814: 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

Table 815: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	ALM	0x0	RW	RTC Alarm interrupt

13. Counter/Timer Module

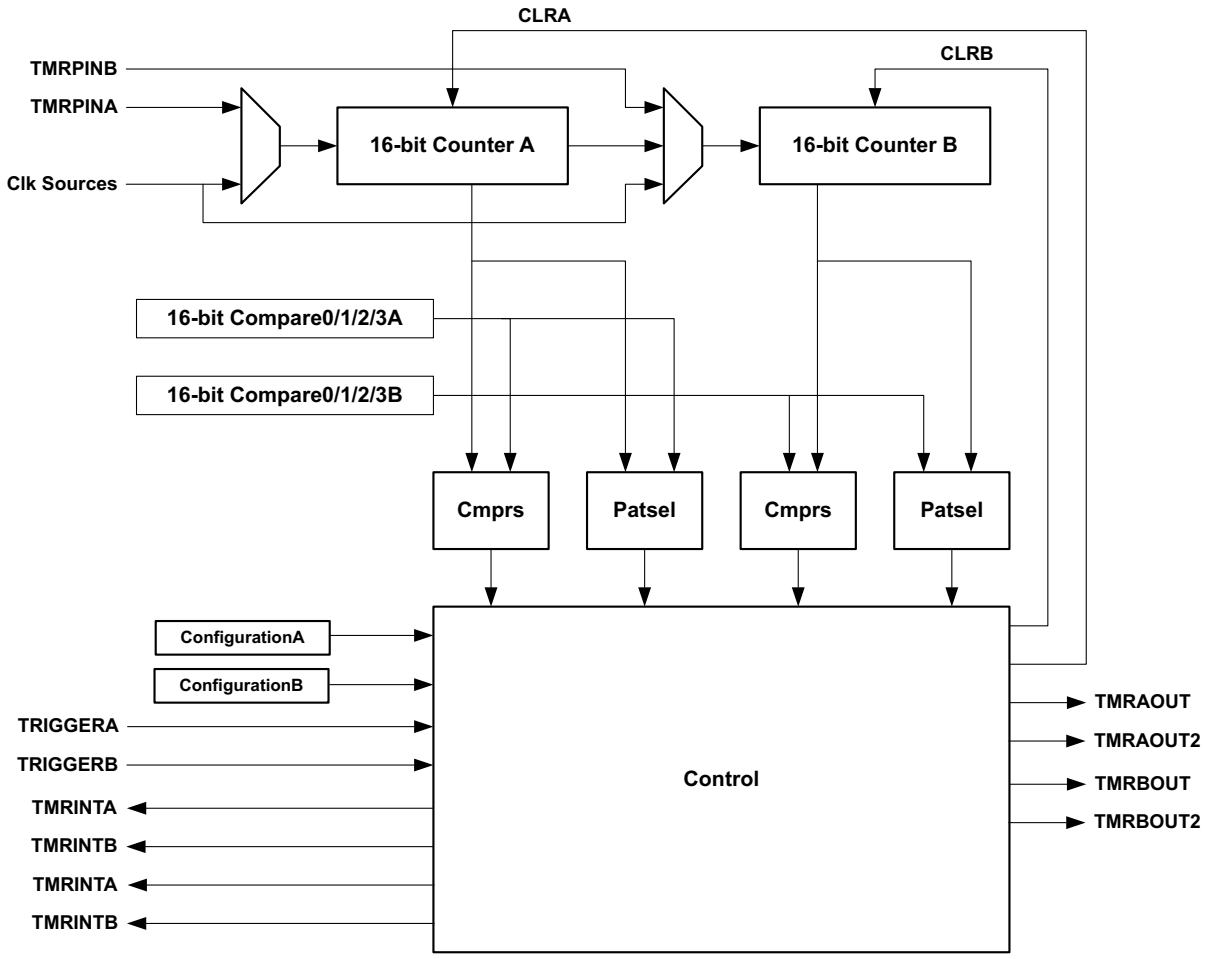


Figure 66. Block Diagram for One Counter/Timer Pair

13.1 Functional Overview

The Apollo3 Blue MCU Timer/Counter module includes eight Timer/Counter pairs, one of which is shown in Figure 66, as well as a system timer shown in Figure 84 in the System Timer chapter. Each Timer/Counter pair includes two very low power asynchronous 16-bit counters, which may be combined to provide a 32-bit counter. Eight registers contain reset values for the counters and/or comparison values to allow the generation of complex internal and external signals. Each Timer/Counter has an external pin connection, and can be configured to provide a variety of functions:

- 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
- Interrupt after a specified number of external pulses

- Generate outputs triggered or terminated by outputs of other Timer/Counters
- Generate a specified number of patterns
- Special inversion functions to support bidirectional stepper motor patterns

13.2 Counter/Timer Functions

Each Counter/Timer operates in a mode controlled by the REG_CTIMER_CTCTRLx_TMRxyFN bit field (x=0 to 7, y=A or B). The mode affects both the generation of interrupts and the control of the outputs. Each mode is described in the following sections. Note that for all functions except for Pattern Generation, a REG_CTIMER_CMPR0/1/2/3 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 eight modes:

0 => Single Count: Counts one time to the compare value, then the output changes polarity and stays at that level, with an optional interrupt.

1 => Repeated Count: Periodic 1-clock-cycle wide pulses with optional interrupts.

2 => Single Pulse (One Shot): A single pulse of programmed width, with an optional interrupt.

3 => Repeated Pulse: A rectangular (or square) waveform with programmed high and low widths, and optional interrupts on each cycle.

4 => Single Pattern: one burst of bits specified by the CMPR0/1/2/3 registers.

5 => Repeated Pattern: repeated burst of bits specified by the CMPR0/1/2/3 registers.

6 => Continuous: Free running timer with a single level change on the output and a single optional interrupt.

7 => Alternate Pulse: like Repeated Pulse but alternating between two different pulse width/spacing settings.

13.2.1 Single Count (FN = 0)

Operation in this mode is shown in Figure 67. 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.

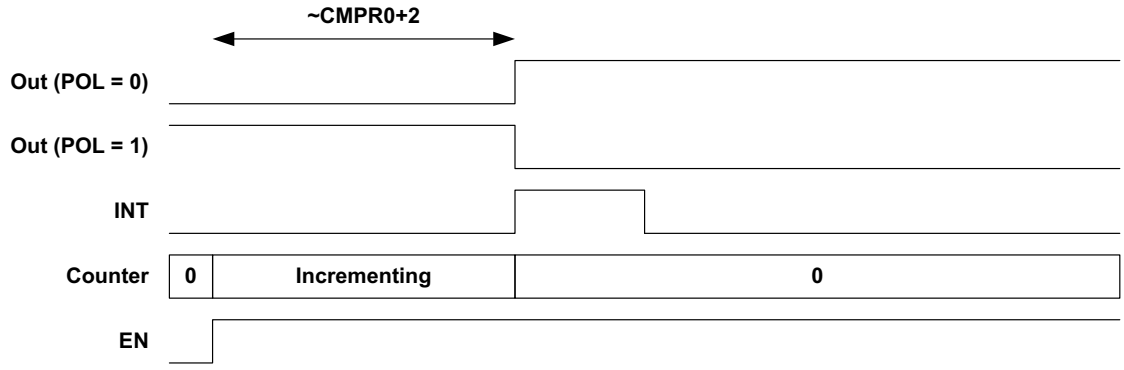


Figure 67. Counter/Timer Operation, FN = 0

13.2.2 Repeated Count (FN = 1)

Operation in this mode is shown in Figure 68. 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.

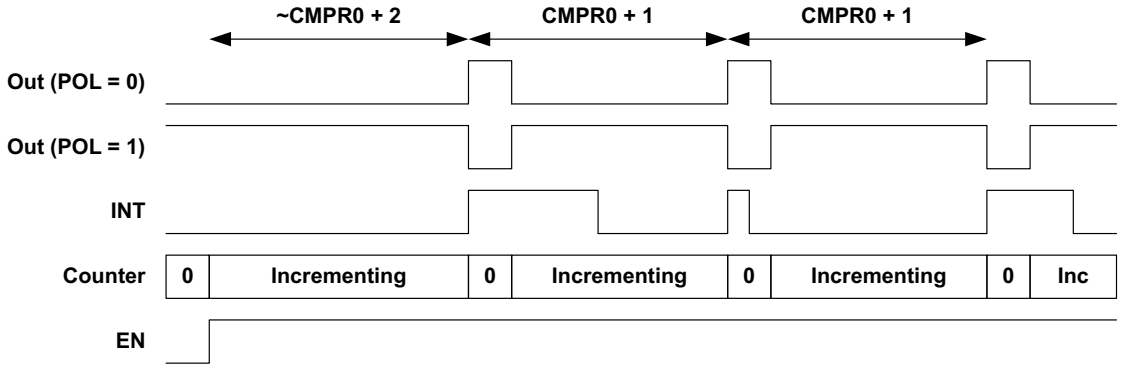


Figure 68. Counter/Timer Operation, FN = 1

13.2.3 Single Pulse (FN = 2)

Operation in this mode is shown in Figure 69. 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.

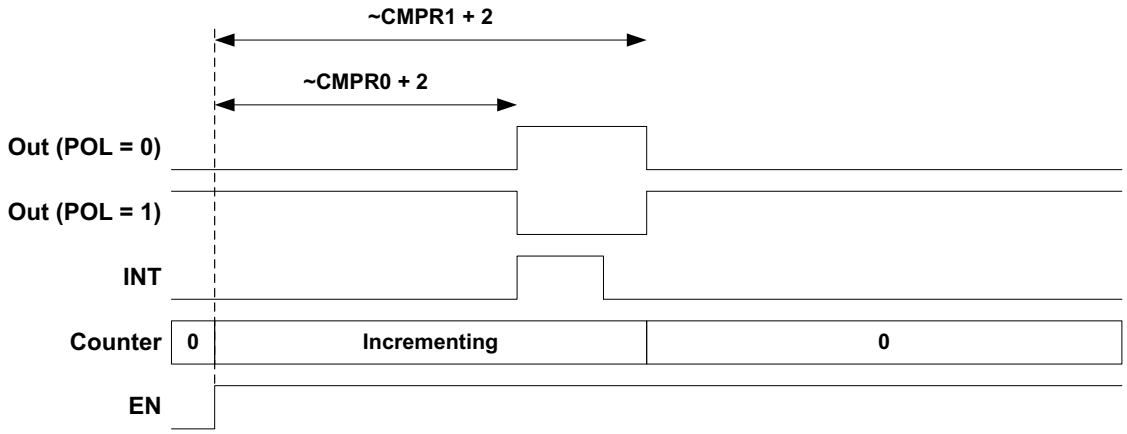


Figure 69. Counter/Timer Operation, FN = 2

The normal interrupt is generated on the rising edge of the output (before polarity is applied) if IE0 is set, as shown in Counter/Timer Operation, FN = 2Figure 69. The secondary interrupt is generated on the falling edge of the output if the REG_CTIMER_TMRxyIE1 bit is set.

13.2.4 Repeated Pulse (FN = 3)

Operation in this mode is shown in Figure 70. 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.

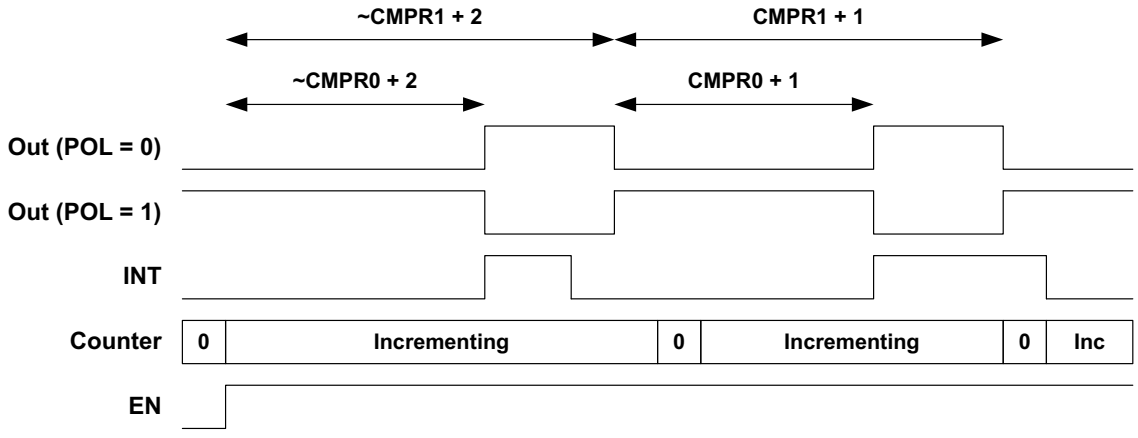


Figure 70. Counter/Timer Operation, FN = 3

The normal interrupt is generated on the rising edge of the output (before polarity is applied) if IE0 is set, as shown in Counter/Timer Operation, FN = 3Figure 70. The secondary interrupt is generated on the falling edge of the output if the IE1 bit is set.

13.2.5 Single Pattern (FN = 4)

In this mode the CTIMER outputs are generated from the pattern in the CMPR0/1/2/3 registers rather than from comparisons to the Counter. The Counter is still used to step through the pattern bits. Bit 0 of CMPR0 is output when the Counter is 0, bit 1 is output when the Counter is 1, and so on until the Counter reaches 16. At that point bit 0 from CMPR1 is output. Similarly, when the Counter reaches 32 the bits from CMPR2 will be output, and when the Counter reaches 48 the bits from CMPR3 will be output. When the Counter reaches the limit set by REG_CTIMER_CTAUXxy_LMT, the pattern generation stops and OUT returns to 0 as shown in Figure 71. Note that this results in LMT + 1 bits being generated. The pattern generation will begin 1 or 2 clock cycles after EN is asserted. The polarity of OUT is controlled by the POL bit as in other cases. If LMT is greater than 127, the pattern will repeat until LMT is reached and then stop.

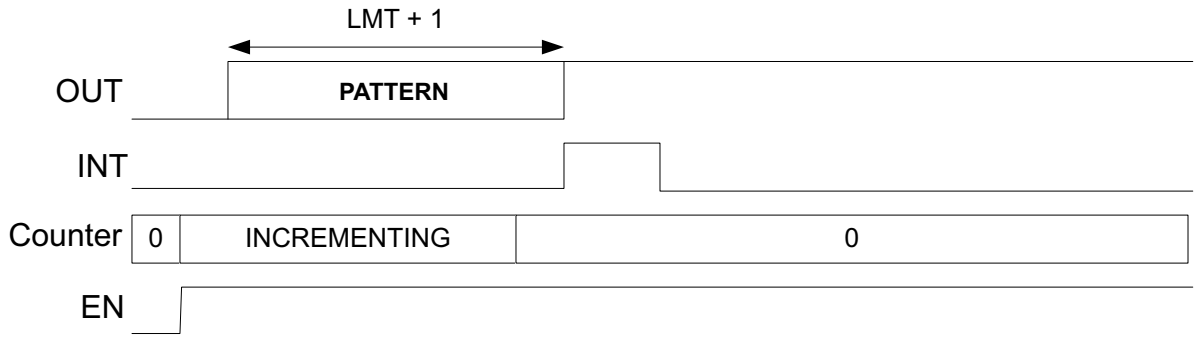


Figure 71. Counter/Timer Operation, FN = 4

The primary interrupt is generated when the pattern completes if IE0 is set. If LMT is greater than 31 and less than 63, the secondary interrupt will be generated when the Counter increments to 32 if IE1 is set. If LMT is 63, the secondary interrupt will be generated both when the Counter increments to 32, and when the Counter rolls over to 0.

If LMT is greater than 63, the pattern generation will use 128-bit mode. This mode is only available for the A CTIMER of a CTIMER pair, and the REG_CTIMER_CTCTRLx_CTRLINKx bit must be set. In this mode, the first 64 bits of the pattern are taken from the CMPRA0/1/2/3 registers as described above, and the remaining bits (up to a total of 128) are taken from the CMPRB0/1/2/3 registers. If IE1 is set, the secondary interrupt will be generated when the Counter increments to 64. If LMT is 127, the secondary interrupt will also be generated when the Counter rolls over to 0.

If LMT is greater than 127, the pattern will continue to repeat until the Counter reaches LMT. The secondary interrupt will be generated each time the Counter increments to a multiple of 64.

13.2.6 Repeat Pattern (FN = 5)

In this mode the CTIMER outputs the pattern from CMPR0/1/2/3 just as in the Single Pattern case, but the pattern repeats as soon as the LMT value is reached, as shown in Figure 72. The polarity is controlled by POL, and if LMT is greater than 63 the pattern will repeat within each pattern burst, although it will restart at the beginning once LMT is reached. The pattern generation will begin between 1 and 2 clock cycles after EN is asserted.

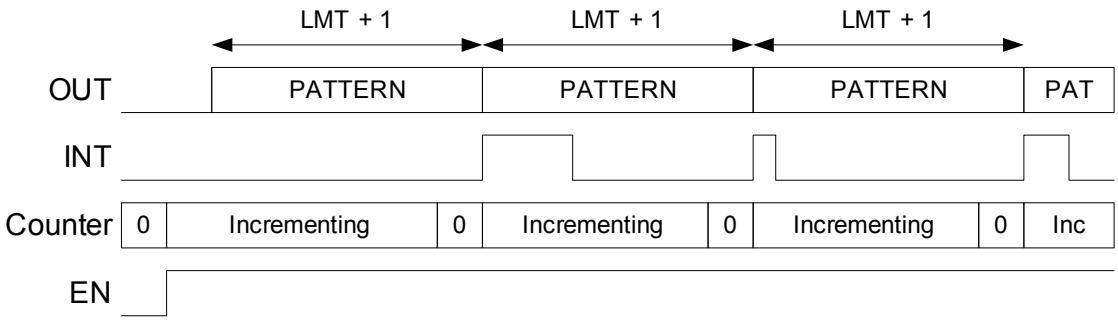


Figure 72. Counter/Timer Operation, FN = 5

The primary interrupt is generated when the pattern rolls over to 0 (if IE0 is set). If LMT is greater than 31 and less than 63, the secondary interrupt will be generated when the Counter increments to 32 (if IE1 is set). If LMT is 63, the secondary interrupt will be generated both when the Counter increments to 32, and when the Counter rolls over to 0.

If LMT is greater than 63, the pattern generation will use 128-bit mode. This mode is only available for the A CTIMER of a CTIMER pair, and the REG_CTIMER_CTCTRLx_CTRLINKx bit must be set. In this mode, the first 64 bits of the pattern are taken from the CMPRA0/1/2/3 registers as described above, and the remaining bits (up to a total of 128) are taken from the CMPRB0/1/2/3 registers. The secondary interrupt will be generated when the Counter increments to 64 (if IE1 is set). If LMT is 127, the secondary interrupt will also be generated when the Counter rolls over to 0.

If LMT is greater than 127, the pattern will continue to repeat until the Counter reaches LMT, and then it will repeat from the beginning. The secondary interrupt will be generated each time the Counter increments to a multiple of 64.

13.2.7 Continuous (FN = 6)

Operation in this mode is shown in Figure 73. 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.

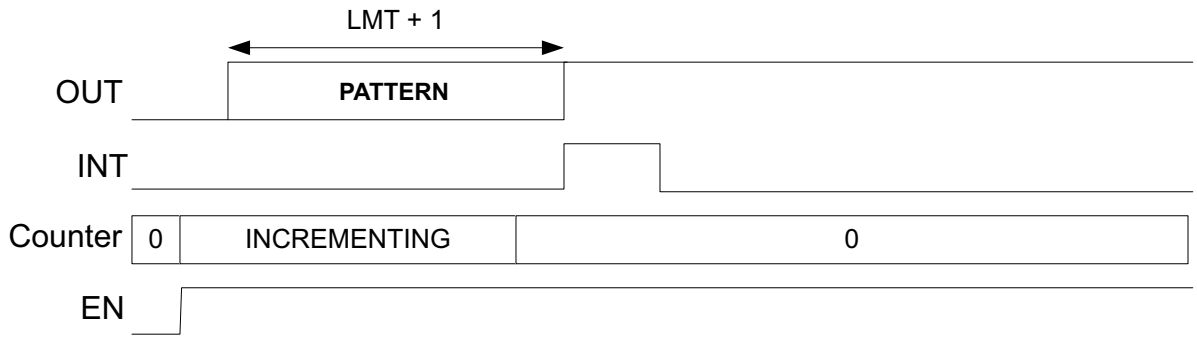


Figure 73. Counter/Timer Operation, FN = 4

13.2.8 Alternate Pulse (FN = 7)

Operation in this mode is shown in Figure 74, and is very similar to Repeated Pulse mode (FN = 3). The only difference is that at the end of each cycle, the comparison register switch between CMPR0/1 and CMPR2/3. This can be used to create a more complex stream of pulses, and may also be used to support an efficient software controlled audio output.

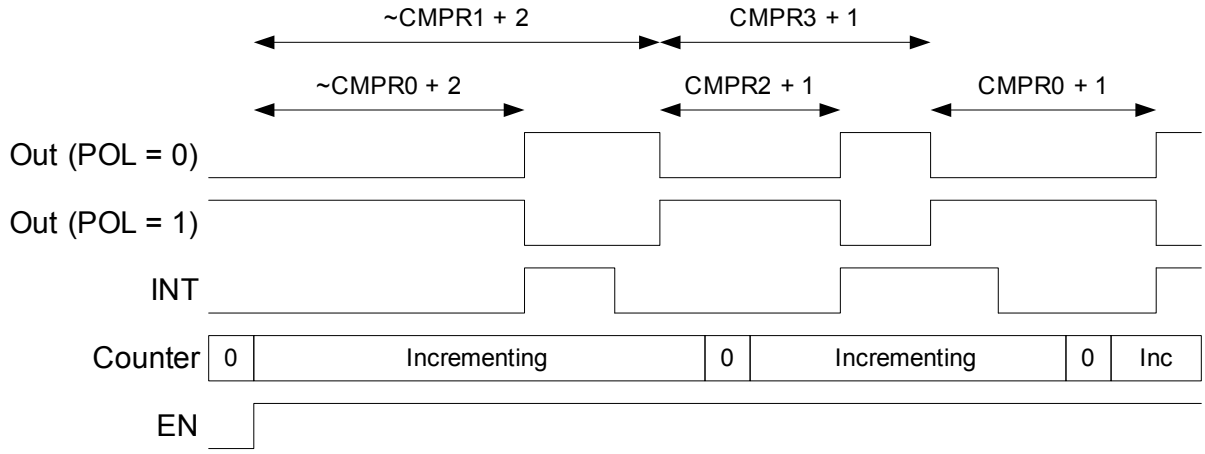


Figure 74. Counter/Timer Operation, FN = 7

The normal interrupt is generated on the rising edge of the output (before polarity is applied) if IE0 is set, as shown in Counter/Timer Operation, FN = 7Counter/Timer Operation, FN = 3. The secondary interrupt is generated on the falling edge of the output if the IE1 bit is set.

13.3 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 CMPR0 , CMPR1, CMPR2 and CMPR3 registers for each 16-bit counter are concatenated to provide the 32-bit comparison values, and all timer modes are supported.

13.4 Creating a Secondary Output with CMPR2/3

In any of the Count or Pulse modes (FN = 0, 1, 2 or 3), the REG_CTIMER_CMPR2 and REG_CTIMER_CMPR3 registers provide two additional comparison points. When the counter reaches a value in either CMPR2 or CMPR3, the secondary output OUT2 is toggled. This allows the creation of complex combinations of the two outputs, as shown in Complex Operations with CMPR2 and CMPR3. In these examples, the CTIMER is configured in repeated pulse mode (FN = 3) to produce the OUT output, and several variations of the output OUT2 are shown. The third example is particularly interesting. If CMPR2 and CMPR3 are set to the same value, or one of them is set to a value larger than CMPR1, OUT2 will toggle only once per OUT cycle, creating a divide-by-two signal.

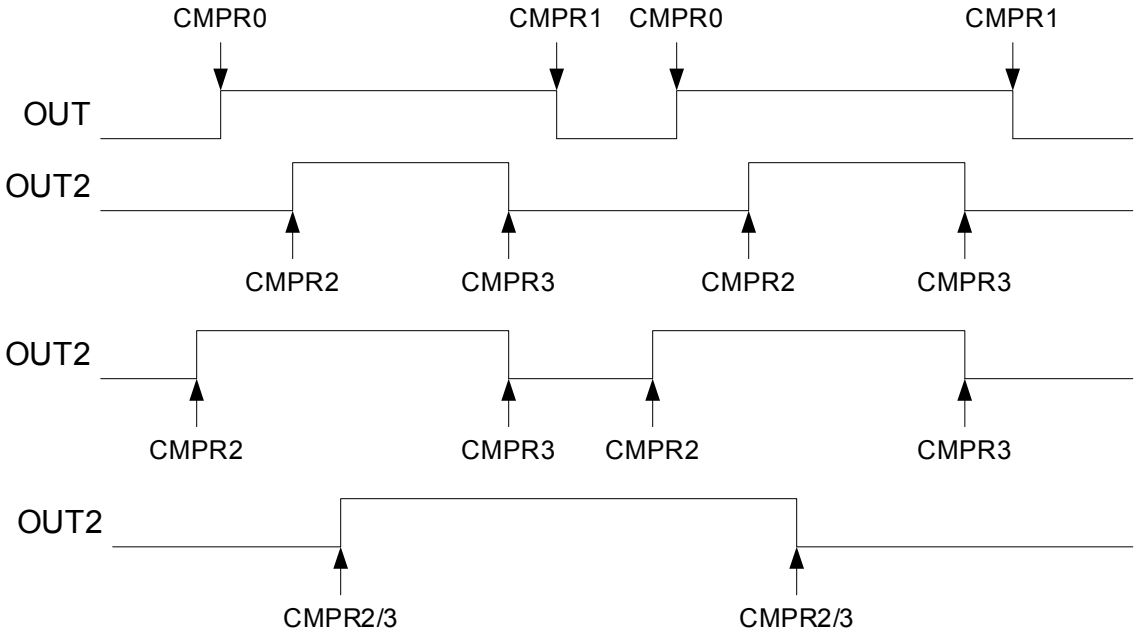


Figure 75. Complex Operations with CMPR2 and CMPR3

CMPR2 and CMPR3 operate in the same way for pulse and count modes. Thus in Single Count mode (FN = 0) for example, OUT2 can produce a single pulse or transition at any time prior to the termination of the count when OUT goes high. The polarity of OUT2 is controlled by the REG_CTIMER_CTAUXxy_POL23 bit.

13.5 Generating Dual Patterns

If the REG_CTIMER_CTAUXxy_EN23 bit is set in a Pattern Mode, a dual pattern will be created on OUT and OUT2. The pattern on OUT will use only the CMPR0/1 register bits, and the pattern on OUT2 will use the CMPR2/3 register bits, so that the longest dual pattern that can be created from a single CTIMER is 32 bits. The output patterns are shown in Figure 76. Dual patterns are particularly valuable in the case of stepper motor control signals, which require positive and negative signals to be generated synchronously.

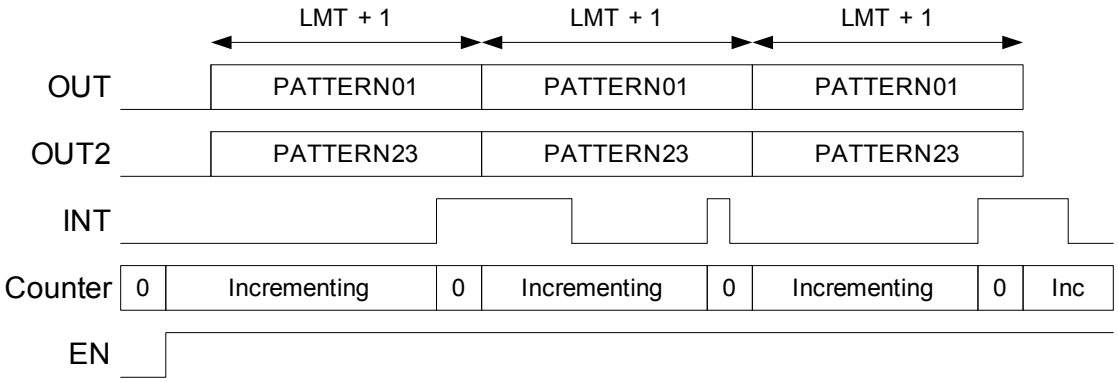


Figure 76. Dual Pattern Generation

13.6 Synchronized A/B Patterns

If the CTLINK bit is set for the B timer of a pair when a Pattern mode is selected (FN = 4 or 5), the pattern comparison value is taken from the A Counter rather than the B Counter. This allows the generation of dual up to 64-bit patterns using the OUT outputs of both the A and B Timers with EN23 clear in both cases, or quad up to 32-bit patterns using the OUT and OUT2 outputs of both timers with EN23 set in both cases.

13.7 Triggering Functions

The REG_CTIMER_CTAUXxy_TRIG field allows the specification of the output of another CTIMER to be used as a trigger. There are several areas where the trigger function may be used to create extremely sophisticated pattern outputs. If the TRIG field is 0, the triggering function is disabled. Otherwise, TRIG selects the internal timer output to be used as the trigger.

13.7.1 Initiating a One-shot Operation

If the mode of a CTIMER is a one-shot mode (FN = 2 or 4), when EN is asserted one cycle of the operation will be executed. At that point, a rising edge on the trigger signal selected by TRIG will cause the operation to be executed again. This allows the creation of complex operations with a single configuration. Figure 77 shows an example of this. The TRIG signal is the OUT signal from a CTIMER configured for Repeat Count.

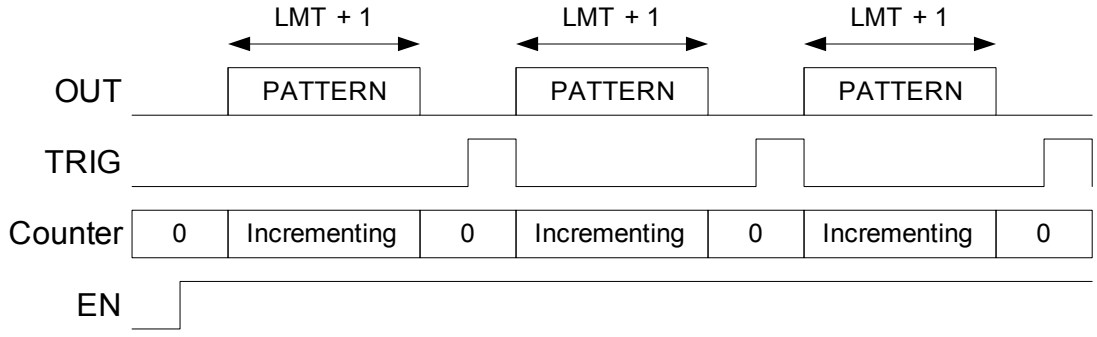


Figure 77. Triggered One-Shot Patterns

13.7.2 Terminating a Repeat Operation

If the mode of a CTIMER is a repeat mode (FN = 1, 3 or 5), the rising edge of the TRIG signal will cause the repeated operation to terminate after the current cycle is complete. This allows the easy creation of a

burst of pulses or patterns of a specified length. This is shown in Figure 78 for the case of Repeat Pulse mode. In this case the TRIG signal is the output of a TIMER configured in Single Pulse mode (FN = 2) with the time configured to be somewhat more than 3 times the pulse repeat. When the TRIG signal occurs, the pulse output is terminated at the end of the current cycle.

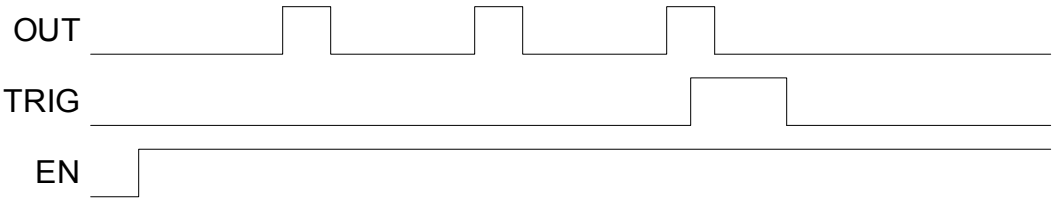


Figure 78. Terminated Repeat Patterns

13.7.3 Complex Patterns with Triggers

The two trigger modes can be combined to produce even more complex patterns. As a particular example, the OUT signal of Figure 78 could be selected as the TRIG signal in Figure 77. This would then produce exactly four bursts of the pattern.

13.7.4 Dual Edge Triggers

Some of the trigger input selections specify dual edge triggers. In that case, the trigger occurs on both the rising and falling edge of the trigger signal. This is very valuable in some stepper motor applications described below.

13.7.5 Trigger Controlled Inversion

If the REG_CTIMER_CTAUXxy_TMRxyTINV bit is set, both the OUT and OUT2 outputs will be XORed with the trigger signal selected by TRIG. This enables some complex stepper motor configurations described below.

13.8 Clocking Timer/Counters with Other Counter/Timer Outputs

There are cases where it is very valuable to use the output of a CTIMER as the clock of another CTIMER. The TMRxyCLK field includes choices which implement this function, in addition to the normal clocks taken from the internal oscillators. An example of such a function is the terminated count shown in Figure 78. If the clock of the timer which produces the TRIG signal were taken from the OUT output of the first timer, the CMPR0 value used for the trigger generator would be trivially calculated as 2, and would be independent of the actual clock used to generate the OUT signal.

13.9 Global Timer/Counter Enable

There are times when it is very important to be able to start multiple Timer/Counters precisely together, particularly in cases where one output is used as the trigger of another. The REG_CTIMER_GLOBEN register contains one enable bit for each Timer/Counter, which is ANDed with the local EN bit of the timer. The GLOBEN register normally has all bits set to 1, so that the local EN bits control the timers. For synchronized enabling, the GLOBEN register bits to be synchronized are set to 0, and then the local EN bits of those timers are set to 1. At that point a single write to the GLOBEN register will enable all of the selected timers at once.

13.10Power Optimization by Measuring HCLK

Each timer has the capability to select the processor clock HCLK 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 = 6 so that they count continuously. One is supplied HCLK 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 HCLKs could have occurred in that time, and the HCLK count value defines how many actual HCLKs were received in that time. The ratio is an accurate measurement of the percentage of time the CPU is asleep, and is an effective tool for power optimization.

13.11 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 REG_CTIMER_ CTCTRL3 _ADCEN bit is set, the output of the timer is sent to the ADC which uses it as a trigger. Mode 1 is typically selected. TMRA3IE may be set to generate an interrupt whenever the trigger occurs, but typically the ADC interrupt will be used for this purpose.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.

13.12 Software Generated Serial Data Stream

It is possible to use the Repeat Pattern mode to produce a serial data stream, such as PDM. A Timer/Counter would be configured to use an external pad as the clock. Software would load the CMPR0/1/2/3 registers with the first 64 bits of the pattern. When the secondary interrupt is received, the first 32 bits will have been transferred, and software can load the next 32 bits into CMPR0/1. When the next secondary interrupt is generated, software can load the next 32 bits into the CMPR2/3 registers, and continue to toggle between the two pairs of registers. This can continue indefinitely.

Note that the requirement for software to update the registers continuously will require the processor to remain awake, unless the transfer is quite slow. For example, if the clock frequency were 1 MHz, software would have to load a 32-bit register every 32 us, which would probably not support a wakeup interrupt. However, at lower frequencies this could be a useful function.

13.13 Software Generated PWM Audio Output

The Alternate Pulse mode can be used to efficiently create an audio output stream in software. In many cases, audio can be effectively produced by generating a stream of pulses with a fixed period, but a variable duty cycle (i.e. a variable pulse width). When this is applied through a low pass filter, reasonable audio output will result. In Alternate Pulse mode, CMPR1 and CMPR3 are set to the desired sample period, and CMPR0 and CMPR2 are configured with the widths of the first two pulses. When the interrupt occurs at the end of the period, software loads CMPR0 with the next pulse width. When the next interrupt occurs, software loads CMPR2 with the next pulse width, and toggles between the two registers for each subsequent pulse. Because the sample rate of audio is often quite slow, software can generally handle this process in an interrupt driven fashion. When the interrupt occurs, there is a full sample period before new register data is needed, so that the interrupt service requirement is easy to achieve.

13.14 Stepper Motors Driven by Pattern Generation

Stepper motors can be driven by the CTIMERs by utilizing the pattern generation feature. Some of the key pattern generation features are arbitrary patterns up to 128 bits long, synchronization of multiple CTIMER pattern generation outputs, and the ability to use another CTIMER to generate the pattern clock base for CTIMER pattern generation output.

13.15Pattern-based Sine Wave Examples

Some applications, such as driving the Linear Resonance Actuator (LRA) in a Haptic Driver or vibrator, require the generation of a pattern which is integrated into an analog signal, most commonly as a sine wave. Figure 79 shows the typical function. The square pulses have variable duty cycles, and they are integrated by the external device to produce the sine wave. The external device typically has positive and negative inputs, so the positive-going pulses occur on a pin which is connected to the positive input and the negative-going pulses are actually positive pulses connected to the negative input. The CTIMER can generate these pulse trains in two different ways.

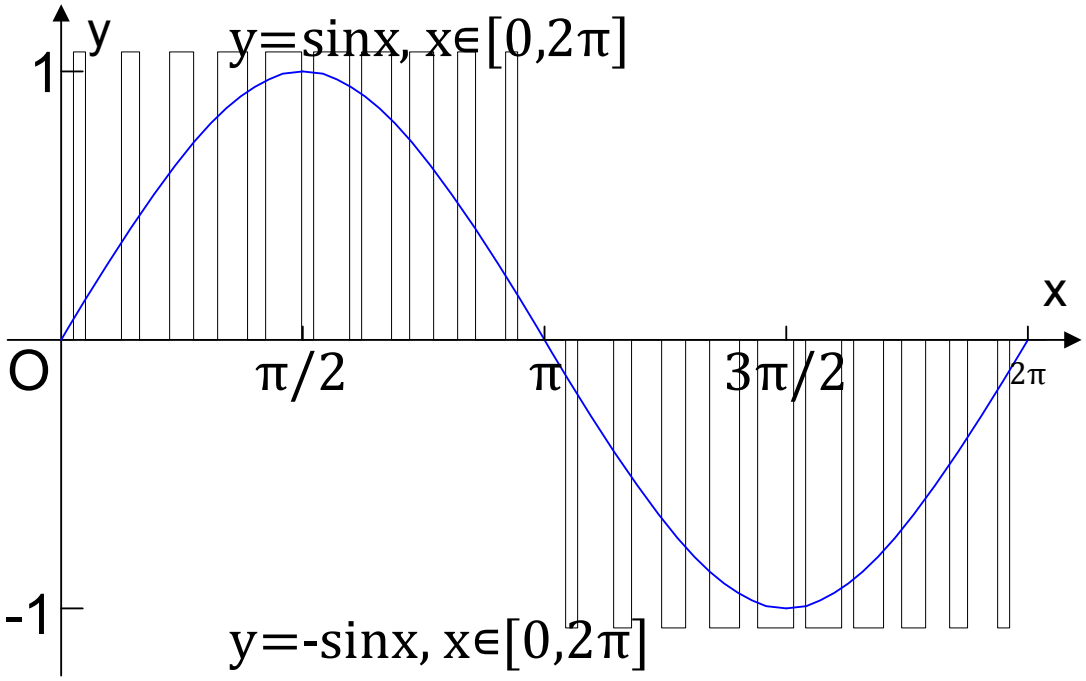


Figure 79. Creating a Sine Wave

13.15.1PWM-based Pulse Trains

The pulse patterns shown in Figure 79 can be generated using the Alternate Pulse function described in Section 13.2.8. The PWM pulses are shown in Figure 80. Initially CMPR0/1/2/3 are configured with the desired parameters for the first two pulses. In many cases, the PWM Period is fixed so that CMPR1/3 contain the same value and never change, but these times can also be varied. When the first interrupt is received at the end of first CMPR0 period, software will update the CMPR0 register with the value required in the next period (and would also update CMPR3 with the next value if it should change). When the next interrupt is received at the comparison to CMPR2, a new CMPR2 value (and a new CMPR1 value if desired) are loaded. This process proceeds throughout the cycle which generates ½ of the sine wave. A similar process is repeated for the negative output to produce the second ½ of the sine wave. Note that software must be able to respond to the interrupt within a period which is slightly longer than the PWM Period, in order to insure that the correct comparison values are loaded.

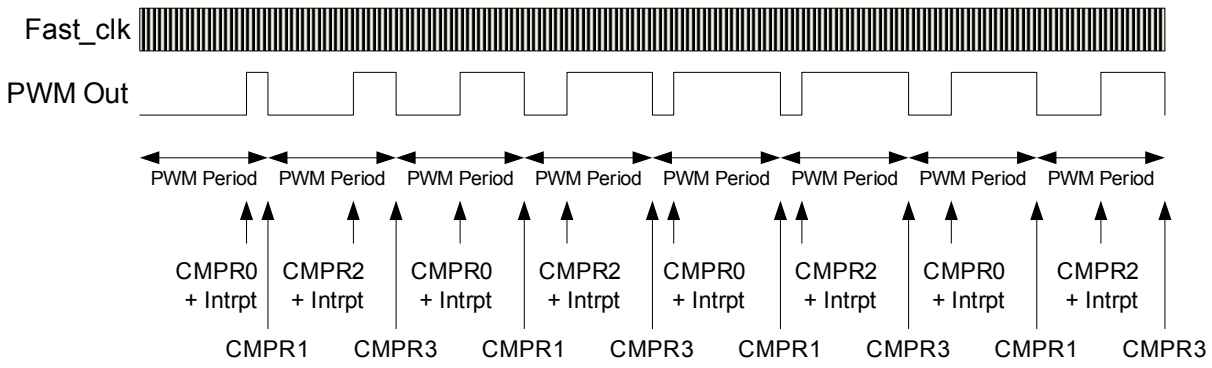


Figure 80. PWM-based Pulse Train

13.15.2Pattern-based Pulse Trains

The pulse patterns may also be generated using the Repeated Pattern function described in Section 13.2.6. This is shown in Figure 81, and assumes the same pattern as the one in Figure 80. However, in this case the first N bits of the desired pattern (where N can be 64 or 128) are loaded into the CMPR0/1/2/3 registers of either a single CTIMER (for 64-bit patterns) or the A and B CTIMERs of a CTIMER pair (for 128-bit patterns). A full pattern consists of both Pattern 1 and Pattern 2 in Figure 81, for example. The process is started, and an interrupt will occur after ½ of the first full pattern (32 or 64 bits) has been generated. At that point software loads a new ½ pattern into the appropriate CMPR registers, and the process continues until the complete pattern has been generated.

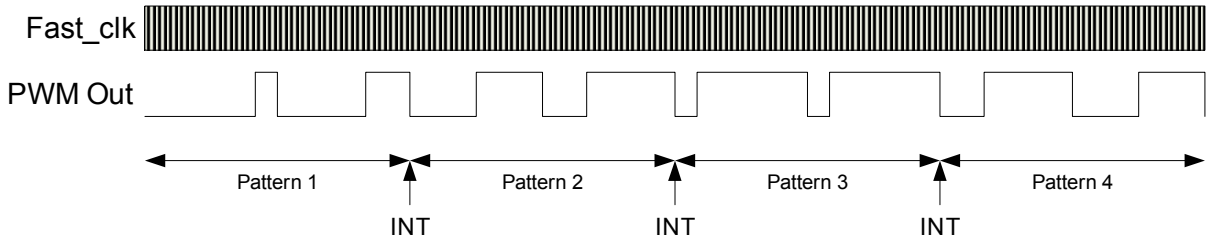


Figure 81. Pattern-based Pulse Train

The selection of 64-bit vs. 128-bit patterns is a tradeoff between power (the longer pattern results in half as many interrupts) and resource usage (the longer pattern requires two CTIMERs instead of one). If there are sufficient CTIMERs available, the 128-bit pattern is always more efficient.

13.15.3Selecting the Optimal Method

Both of the above approaches produce the same result in terms of the generated pattern, so the optimal selection is a function of minimizing the number of interrupts required to produce the overall pattern. Fewer interrupts result in longer CPU sleep times and less interrupt servicing overhead, which reduces the overall power. If the half pattern can cover more time than the average PWM Period, the Pattern-based approach will typically be more energy efficient. Whether this is the case is a function of the desired fast clock frequency, which determines the precision of the pulses, and the desired pulse lengths.

13.16CLR and EN Details

The overall operation of each CTIMER is controlled by two configuration bits, CLR and EN. When CLR is set to 1, the CTIMER is immediately set to all zeroes and will remain there independent of any other configuration. CLR is typically used to initialize a CTIMER before use.

EN is used to enable (when 1) and disable (when 0) the counting function of the CTIMER. However, EN and the deassertion of CLR are synchronized to the selected clock, which must be accounted for when they are used. CLR and EN Operation shows how this synchronization occurs. When CLR is set to 0, the Counter will begin counting on the second edge of the selected clock if EN is set to 1. When EN is set to 0, the Counter will increment on the next clock (from 1 to 2 in CLR and EN Operation) and then hold its current value. When EN is set to 1, the Counter will resume counting on the second following edge.

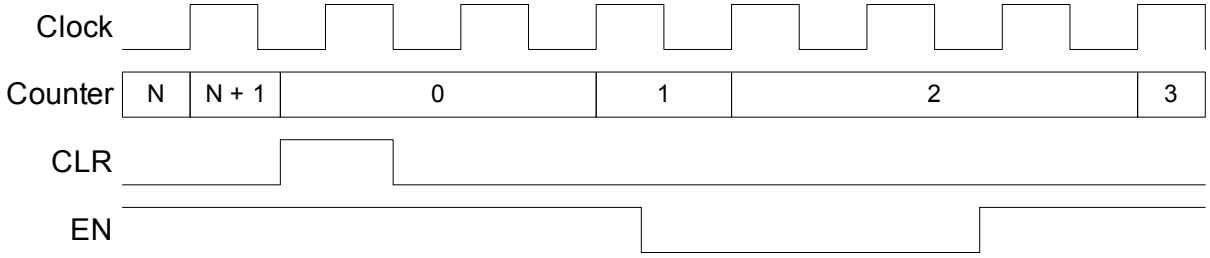


Figure 82. CLR and EN Operation

Since the operation of the processor is essentially asynchronous to the selected clock, the synchronization introduces an uncertainty as to when the Counter will begin counting. If the frequency of the selected clock is high relative to the processor clock, the impact of the synchronization will be negligible. However, for low frequency clocks, external pin clocks and the buck clocks the effective delay caused by the synchronization may be significant.

13.17NOSYNC Function

Under normal conditions, the CTIMER clocks are supplied directly by the clock selected in the REG_CTIMER_CTCTRLx_TMRxyCLK register field. However, if software reads the TMR value the clock will be temporarily synchronized to the processor clock, which may cause the time of edges on an output to move slightly. The time of any edge will be moved by at most 20 ns from the normal time. If this variation is unacceptable for a specific application, it can be eliminated in one of two ways:

1. Do not read the TMR value during the output generation. This is often an acceptable restriction.
2. Set the REG_CTIMER_CTAUXxy_NOSYNC register bit. This will disable the synchronization function.

If the NOSYNC bit is set, the TMR update will no longer be synchronized to the processor clock. As a result, the TMR value read might be incorrect. In this case, software should read the TMR three times in quick succession (with interrupts disabled) and determine the correct value from those. If the first two reads are the same, that is the correct value. If they are different, the third value is correct.

13.18Counter Functions

A CTIMER operates in Counter mode when the TMRxyCLK field selects either the external pad input (if 0x00) or a buck pulse input (if 0x10). Although any of the modes may be selected, the typical configuration is Continuous. In this mode the CTIMER will count edges on the selected clock, and may be configured to generate an interrupt on a particular count value. The different clock selections provide different functions.

13.18.1Counting External Edges

If the CLK field is 0x00, the CTIMER clock input comes from an external pad as selected by the INCFG register. This allows the CTIMER to monitor pulses or edges on an external signal.

13.18.2Counting Buck Converter Edges

Apollo3 includes three separate buck converters which provide power for the Processor power domain (BUCKA), the Memory power domain (BUCKB) and the BLE interface module (BUCKBLE). Each CTIMER may be connected to a pulse stream from any of the three analog Buck Converters. One pulse is generated each time the Buck Converter inserts charge into 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.

A possible option to determine energy consumption is as follows. Two counters could be configured with FN = 6 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.

13.19Interconnecting CTIMERs

The OUT or OUT2 output of one CTIMER may be used as either the Trigger or the clock of another CTIMER. Figure 83 shows the interconnection structure for two example CTIMERs, where p and q are A or B and x and y are 0 through 7. The selection of the actual clock or trigger interconnection is made within each CTIMER. The interconnection Matrix is not complete, as each CTIMER can select from only 15 triggers and 12 external clocks.

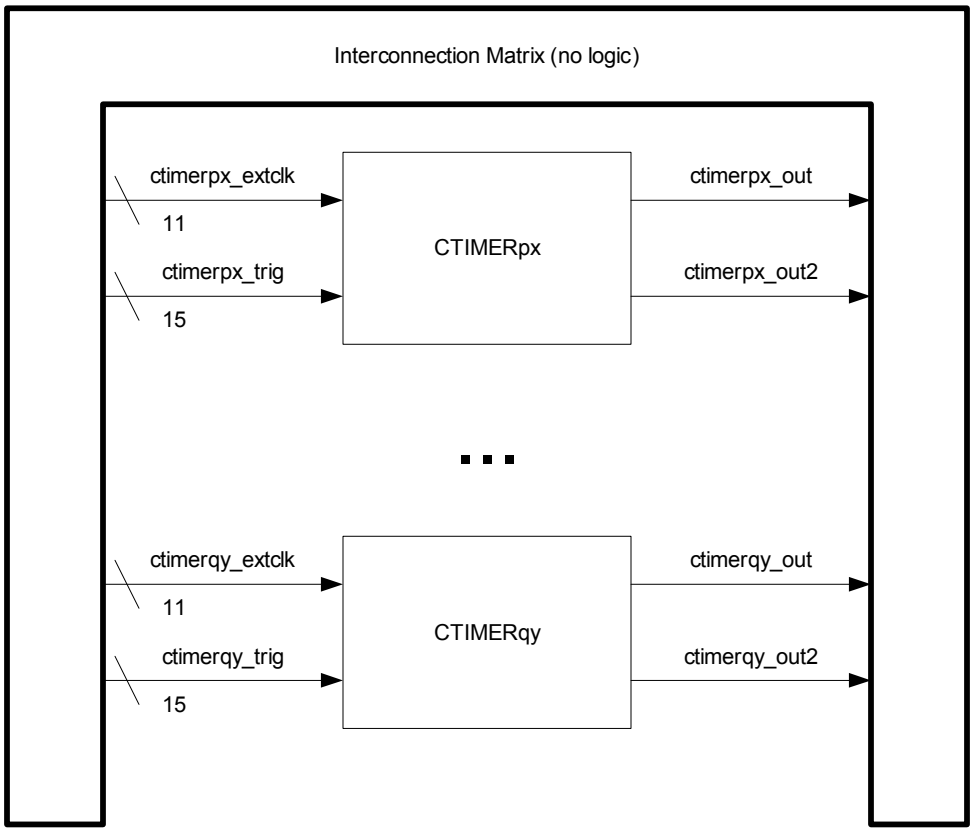


Figure 83. CTIMER Interconnection

13.20Pad Connections from the Timer/Counter

In order to provide flexibility in connecting timers to external devices, a secondary multiplexing mechanism is provided for the timer outputs. There are 32 pads which can be configured for either inputs to or outputs from the Timer/Counter module. Each of these pads can be driven by one of four outputs selected by the REG_CTIMER_OUTCFG0/1/2/3 registers, as shown in Table 818. The ctimer_out column specifies the bit in REG_GPIO_CTENCFG (EN0-EN31) which enables the corresponding pad CT0-31 output. The Output Selection columns define the output choices for each output based on the CFG fields in OUTCFGx, where CFG=0 forces the output to 0, CFG=1 forces the output to 1 and CFG=2 through 7 produce the choices in the table. The shaded cells indicate selections which correspond to the Timer/Counter connections in previous Apollo3 devices..

Table 816: 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
PAD27FNCSEL	2	A1	27
PAD44FNCSEL	2	A1	44
PAD19FNCSEL	2	B1	19
PAD28FNCSEL	2	B1	28
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

Table 817: 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
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

Table 818: Counter/Timer Pad Configuration

Pad (FNCSEL)	ctimer output signal	Output Selection (REG_CTIMER_INCFG)							
		0	1	2	3	4	5	6	7
PAD4 (6)	CT17	Force to 0	Force to 1	A4OUT2	B7OUT	A4OUT	A1OUT2	A6OUT2	A7OUT2
PAD5 (7)	CT8	Force to 0	Force to 1	A2OUT	A3OUT2	A4OUT2	B6OUT	A6OUT2	A7OUT2
PAD6 (5)	CT10	Force to 0	Force to 1	B2OUT	B3OUT2	B4OUT2	A6OUT	A6OUT2	A7OUT2
PAD7 (7)	CT19	Force to 0	Force to 1	B4OUT2	A2OUT	B4OUT	B1OUT2	A6OUT2	A7OUT2
PAD11 (2)	CT31	Force to 0	Force to 1	B7OUT2	A6OUT	B7OUT	B3OUT2	A6OUT2	A7OUT2
PAD12 (2)	CT0	Force to 0	Force to 1	A0OUT	B2OUT2	A5OUT2	A6OUT	A6OUT2	A7OUT2
PAD13 (2)	CT2	Force to 0	Force to 1	B0OUT	B1OUT2	B6OUT2	A7OUT	A6OUT2	A7OUT2
PAD18 (2)	CT4	Force to 0	Force to 1	A1OUT	A2OUT2	A5OUT2	B5OUT	A6OUT2	A7OUT2
PAD19 (2)	CT6	Force to 0	Force to 1	B1OUT	A1OUT	B5OUT2	B7OUT	A6OUT2	A7OUT2
PAD22 (2)	CT12	Force to 0	Force to 1	A3OUT	B1OUT	B0OUT2	B6OUT2	A6OUT2	A7OUT2
PAD23 (2)	CT14	Force to 0	Force to 1	B3OUT	B1OUT	B7OUT2	A7OUT	A6OUT2	A7OUT2
PAD24 (5)	CT21	Force to 0	Force to 1	A5OUT2	A1OUT	B5OUT	A0OUT2	A6OUT2	A7OUT2
PAD25 (2)	CT1	Force to 0	Force to 1	A0OUT2	A0OUT	A5OUT	B7OUT2	A6OUT2	A7OUT2
PAD26 (2)	CT3	Force to 0	Force to 1	B0OUT2	B0OUT	A1OUT	A6OUT	A6OUT2	A7OUT2
PAD27 (2)	CT5	Force to 0	Force to 1	A1OUT2	A1OUT	B6OUT	A7OUT	A6OUT2	A7OUT2
PAD28 (2)	CT7	Force to 0	Force to 1	B1OUT2	B1OUT	B5OUT	A7OUT	A6OUT2	A7OUT2
PAD29 (2)	CT9	Force to 0	Force to 1	A2OUT2	A2OUT	A4OUT	B0OUT	A6OUT2	A7OUT2
PAD30 (2)	CT11	Force to 0	Force to 1	B2OUT2	B2OUT	B4OUT	B5OUT2	A6OUT2	A7OUT2
PAD31 (2)	CT13	Force to 0	Force to 1	A3OUT2	A3OUT	A6OUT	B4OUT2	A6OUT2	A7OUT2
PAD32 (2)	CT15	Force to 0	Force to 1	B3OUT2	B3OUT	A7OUT	A4OUT2	A6OUT2	A7OUT2
PAD33 (6)	CT23	Force to 0	Force to 1	B5OUT2	A7OUT	A5OUT	B0OUT2	A6OUT2	A7OUT2
PAD35 (5)	CT27	Force to 0	Force to 1	B6OUT2	A1OUT	B6OUT	B2OUT2	A6OUT2	A7OUT2

Table 818: Counter/Timer Pad Configuration

Pad (FNCSEL)	ctimer output signal	Output Selection (REG_CTIMER_INCFG)							
		0	1	2	3	4	5	6	7
PAD37 (7)	CT29	Force to 0	Force to 1	B5OUT2	A1OUT	A7OUT	A3OUT2	A6OUT2	A7OUT2
PAD39 (2)	CT25	Force to 0	Force to 1	B4OUT2	B2OUT	A6OUT	A2OUT2	A6OUT2	A7OUT2
PAD42 (2)	CT16	Force to 0	Force to 1	A4OUT	A0OUT	A0OUT2	B3OUT2	A6OUT2	A7OUT2
PAD43 (2)	CT18	Force to 0	Force to 1	B4OUT	B0OUT	A0OUT	A3OUT2	A6OUT2	A7OUT2
PAD44 (2)	CT20	Force to 0	Force to 1	A5OUT	A1OUT	A1OUT2	B2OUT2	A6OUT2	A7OUT2
PAD45 (2)	CT22	Force to 0	Force to 1	B5OUT	B1OUT	A6OUT	A2OUT2	A6OUT2	A7OUT2
PAD46 (2)	CT24	Force to 0	Force to 1	A6OUT	A2OUT	A1OUT	B1OUT2	A6OUT2	A7OUT2
PAD47 (2)	CT26	Force to 0	Force to 1	B6OUT	B2OUT	A5OUT	A1OUT2	A6OUT2	A7OUT2
PAD48 (2)	CT28	Force to 0	Force to 1	A7OUTB	A3OUT	A5OUT2	B0OUT2	A6OUT2	A7OUT2
PAD49 (2)	CT30	Force to 0	Force to 1	B7OUT	B3OUT	A4OUT2	A0OUT2	A6OUT2	A7OUT2

Each timer may be clocked by one of two of the pads, selected by the REG_CTIMER_INCFG register as shown in Table 819. The polarity of the input clock is selection by the POL23 bit.

Table 819: CTIMER Pad Input Connections

CTIMER	INCFG		CTIMER	INCFG	
	0	1		0	1
CTIMERA0	CT0	CT1	CTIMERB0	CT2	CT3
CTIMERA1	CT4	CT5	CTIMERB1	CT6	CT7
CTIMERA2	CT8	CT9	CTIMERB2	CT10	CT11
CTIMERA3	CT12	CT13	CTIMERB3	CT14	CT15
CTIMERA4	CT16	CT17	CTIMERB4	CT18	CT19
CTIMERA5	CT20	CT21	CTIMERB5	CT22	CT23
CTIMERA6	CT24	CT25	CTIMERB6	CT26	CT27
CTIMERA7	CT28	CT29	CTIMERB7	CT30	CT31

The REG_GPIO_CTENCFG register holds one bit for each pad, which selects whether the pad is an output (if 0) or an input (if 1).

The assignments in Table 818 and Table 819 assume that COMMON outputs will be created from either A7OUT2 or A6OUT2. These outputs can also be used in the case where it is desired to drive multiple outputs from the same timer.

Note that for the Pulse and Count modes, the CMPR2/3 registers can always be configured so that OUT2 matches OUT. This provides more flexibility in the pin assignments, as any OUT2 connection can be used as the corresponding OUT function if a separate OUT2 function is not required. For a single 32-bit pattern from a timer, OUT2 can be configured in the CMPR2/3 registers to produce the same pattern as OUT.

The OUT and OUT2 outputs of each CTIMER will be toggling whenever the CTIMER is enabled, independent of any pin connections configured for it. This allows these signals to be used as clocks and triggers for other CTIMERs even when they are not being used as pin outputs.

Example flow is illustrated below:

1) Pick the pad you want to use, from column Pad (FNCSEL).
2) Set that pad's FNCSEL to the value in parentheses.
3) Determine which of the outputs in columns 2-7 you want to use to drive this pin.
4) Set the OUTCFG0/1/2/3_CFGx bitfield to the value of 2 through 7 to select the desired CTIMER output.
5) Clear the bit in CTENCFG corresponding to the CTxx value that matches the pad, to make it an output.

13.21CTIMER 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.

13.21.1Register Memory Map

Table 820: CTIMER Register Map

Address(s)	Register 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
0x40008014	CMPRAUXA0	Counter/Timer A0 Compare Registers
0x40008018	CMPRAUXB0	Counter/Timer B0 Compare Registers
0x4000801C	AUX0	Counter/Timer Auxiliary
0x40008020	TMR1	Counter/Timer Register
0x40008024	CMPRA1	Counter/Timer A1 Compare Registers
0x40008028	CMPRB1	Counter/Timer B1 Compare Registers
0x4000802C	CTRL1	Counter/Timer Control
0x40008034	CMPRAUXA1	Counter/Timer A1 Compare Registers
0x40008038	CMPRAUXB1	Counter/Timer B1 Compare Registers
0x4000803C	AUX1	Counter/Timer Auxiliary
0x40008040	TMR2	Counter/Timer Register
0x40008044	CMPRA2	Counter/Timer A2 Compare Registers
0x40008048	CMPRB2	Counter/Timer B2 Compare Registers
0x4000804C	CTRL2	Counter/Timer Control
0x40008054	CMPRAUXA2	Counter/Timer A2 Compare Registers
0x40008058	CMPRAUXB2	Counter/Timer B2 Compare Registers
0x4000805C	AUX2	Counter/Timer Auxiliary
0x40008060	TMR3	Counter/Timer Register
0x40008064	CMPRA3	Counter/Timer A3 Compare Registers
0x40008068	CMPRB3	Counter/Timer B3 Compare Registers
0x4000806C	CTRL3	Counter/Timer Control
0x40008074	CMPRAUXA3	Counter/Timer A3 Compare Registers
0x40008078	CMPRAUXB3	Counter/Timer B3 Compare Registers
0x4000807C	AUX3	Counter/Timer Auxiliary
0x40008080	TMR4	Counter/Timer Register
0x40008084	CMPRA4	Counter/Timer A4 Compare Registers
0x40008088	CMPRB4	Counter/Timer B4 Compare Registers
0x4000808C	CTRL4	Counter/Timer Control
0x40008094	CMPRAUXA4	Counter/Timer A4 Compare Registers
0x40008098	CMPRAUXB4	Counter/Timer B4 Compare Registers
0x4000809C	AUX4	Counter/Timer Auxiliary
0x400080A0	TMR5	Counter/Timer Register

Table 820: CTIMER Register Map

Address(s)	Register Name	Description
0x400080A4	CMPRA5	Counter/Timer A5 Compare Registers
0x400080A8	CMPRB5	Counter/Timer B5 Compare Registers
0x400080AC	CTRL5	Counter/Timer Control
0x400080B4	CMPRAUXA5	Counter/Timer A5 Compare Registers
0x400080B8	CMPRAUXB5	Counter/Timer B5 Compare Registers
0x400080BC	AUX5	Counter/Timer Auxiliary
0x400080C0	TMR6	Counter/Timer Register
0x400080C4	CMPRA6	Counter/Timer A6 Compare Registers
0x400080C8	CMPRB6	Counter/Timer B6 Compare Registers
0x400080CC	CTRL6	Counter/Timer Control
0x400080D4	CMPRAUXA6	Counter/Timer A6 Compare Registers
0x400080D8	CMPRAUXB6	Counter/Timer B6 Compare Registers
0x400080DC	AUX6	Counter/Timer Auxiliary
0x400080E0	TMR7	Counter/Timer Register
0x400080E4	CMPRA7	Counter/Timer A7 Compare Registers
0x400080E8	CMPRB7	Counter/Timer B7 Compare Registers
0x400080EC	CTRL7	Counter/Timer Control
0x400080F4	CMPRAUXA7	Counter/Timer A7 Compare Registers
0x400080F8	CMPRAUXB7	Counter/Timer B7 Compare Registers
0x400080FC	AUX7	Counter/Timer Auxiliary
0x40008100	GLOBEN	Counter/Timer Global Enable
0x40008104	OUTCFG0	Counter/Timer Output Config 0
0x40008108	OUTCFG1	Counter/Timer Output Config 1
0x4000810C	OUTCFG2	Counter/Timer Output Config 2
0x40008114	OUTCFG3	Counter/Timer Output Config 3
0x40008118	INCFG	Counter/Timer Input Config
0x40008200	INTEN	Counter/Timer Interrupts: Enable
0x40008204	INTSTAT	Counter/Timer Interrupts: Status
0x40008208	INTCLR	Counter/Timer Interrupts: Clear
0x4000820C	INTSET	Counter/Timer Interrupts: Set

13.21.2CTIMER Registers

13.21.2.1TMR0 Register

Counter/Timer Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40008000

This register holds the running time or event count for ctimer 0. This is either for each 16 bit half or for the whole 32 bit count when the pair is linked. If the pair is not linked, they can be running on seperate clocks and are completely independent.

Table 821: 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 822: TMR0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB0	0x0	RO	Counter/Timer B0.
15:0	CTTMRA0	0x0	RO	Counter/Timer A0.

13.21.2.2CMPRA0 Register

Counter/Timer A0 Compare Registers

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40008004

This contains the Compare limits for timer 0 A half.

Table 823: 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 824: 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.

13.21.2.3CMPRB0 Register

Counter/Timer B0 Compare Registers

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40008008

This contains the Compare limits for timer 0 B half.

Table 825: 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 826: 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.

13.21.2.4CTRL0 Register

Counter/Timer Control

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4000800C

This includes the Control bit fields for both halves of timer 0.

Table 827: 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		TMRB0POL	TMRB0CLR	TMRB0IE1	TMRB0IE0	TMRB0FN			TMRB0CLK					TMRB0EN	RSVD			TMRA0POL	TMRA0CLR	TMRA0IE1	TMRA0IE0	TMRA0FN			TMRA0CLK					TMRA0EN

Table 828: 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.
30:29	RSVD	0x0	RO	RESERVED
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

Table 828: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B0, assert, count to CMPR1B0, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM
21:17	TMRB0CLK	0x0	RW	Counter/Timer B0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA0 = 0x14 - Clock source is CTIMERA0 OUT. CTMRB1 = 0x15 - Clock source is CTIMERB1 OUT. CTMRA1 = 0x16 - Clock source is CTIMERA1 OUT. CTMRA2 = 0x17 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x18 - Clock source is CTIMERB2 OUT. CTMRB3 = 0x19 - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1A - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB0EN	0x0	RW	Counter/Timer B0 Enable bit. DIS = 0x0 - Counter/Timer B0 Disable. EN = 0x1 - Counter/Timer B0 Enable.
15:13	RSVD	0x0	RO	RESERVED

Table 828: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
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.
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A0, assert, count to CMPR1A0, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 828: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
5:1	TMRA0CLK	0x0	RW	Counter/Timer A0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB0 = 0x14 - Clock source is CTIMERB0 OUT. CTMRA1 = 0x15 - Clock source is CTIMERA1 OUT. CTMRB1 = 0x16 - Clock source is CTIMERB1 OUT. CTMRA2 = 0x17 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x18 - Clock source is CTIMERB2 OUT. CTMRB3 = 0x19 - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1A - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
0	TMRA0EN	0x0	RW	Counter/Timer A0 Enable bit. DIS = 0x0 - Counter/Timer A0 Disable. EN = 0x1 - Counter/Timer A0 Enable.

13.21.2.5CMPRAUXA0 Register

Counter/Timer A0 Compare Registers

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40008014

Enhanced compare limits for timer half A. This is valid if timer 0 is set to function 4 and function 5.

Table 829: CMPRAUXA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A0																CMPR2A0															

Table 830: CMPRAUXA0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A0	0x0	RW	Counter/Timer A0 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A0	0x0	RW	Counter/Timer A0 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.6CMPRAUXB0 Register

Counter/Timer B0 Compare Registers

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40008018

Enhanced compare limits for timer half B. This is valid if timer 0 is set to function 4 and function 5.

Table 831: CMPRAUXB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B0																CMPR2B0															

Table 832: CMPRAUXB0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B0	0x0	RW	Counter/Timer B0 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B0	0x0	RW	Counter/Timer B0 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.7AUX0 Register

Counter/Timer Auxiliary

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4000801C

Control bit fields for both halves of timer 0.

Table 833: AUX0 Register

31	30	29	28	27	26	25	24	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	TMRB0EN23	TMRB0POL23	TMRB0TINV	TMRB0NOSYNC	TMRB0TRIG				RSVD	TMRB0LMT						RSVD	TMRA0EN23	TMRA0POL23	TMRA0TINV	TMRA0NOSYNC	TMRA0TRIG				TMRA0LMT						

Table 834: AUX0 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB0EN23	0x0	RW	Counter/Timer B0 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB0POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB0TINV	0x0	RW	Counter/Timer B0 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
27	TMRB0NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB0TRIG	0x0	RW	Counter/Timer B0 Trigger Select. DIS = 0x0 - Trigger source is disabled. A0OUT = 0x1 - Trigger source is CTIMERA0 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. B2OUT = 0x4 - Trigger source is CTIMERB2 OUT. B5OUT = 0x5 - Trigger source is CTIMERB5 OUT. A4OUT = 0x6 - Trigger source is CTIMERA4 OUT. B4OUT = 0x7 - Trigger source is CTIMERB4 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. B7OUT2 = 0xA - Trigger source is CTIMERB7 OUT2. A2OUT2 = 0xB - Trigger source is CTIMERA2 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B5OUT2DUAL = 0xE - Trigger source is CTIMERB5 OUT2, dual edge. A5OUT2DUAL = 0xF - Trigger source is CTIMERA5 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED

Table 834: AUX0 Register Bits

Bit	Name	Reset	RW	Description
21:16	TMRB0LMT	0x0	RW	Counter/Timer B0 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA0EN23	0x0	RW	Counter/Timer A0 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA0POL23	0x0	RW	Counter/Timer A0 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA0TINV	0x0	RW	Counter/Timer A0 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA0NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA0TRIG	0x0	RW	Counter/Timer A0 Trigger Select. DIS = 0x0 - Trigger source is disabled. B0OUT = 0x1 - Trigger source is CTIMERB0 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A1OUT = 0x4 - Trigger source is CTIMERA1 OUT. B1OUT = 0x5 - Trigger source is CTIMERB1 OUT. A5OUT = 0x6 - Trigger source is CTIMERA5 OUT. B5OUT = 0x7 - Trigger source is CTIMERB5 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. B6OUT2 = 0xA - Trigger source is CTIMERB6 OUT2. A2OUT2 = 0xB - Trigger source is CTIMERA2 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge.
6:0	TMRA0LMT	0x0	RW	Counter/Timer A0 Pattern Limit Count.

13.21.2.8TMR1 Register

Counter/Timer Register

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40008020

This register holds the running time or event count for ctimer 1. This is either for each 16 bit half or for the whole 32 bit count when the pair is linked. If the pair is not linked, they can be running on seperate clocks and are completely independent.

Table 835: 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 836: TMR1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB1	0x0	RO	Counter/Timer B1.
15:0	CTTMRA1	0x0	RO	Counter/Timer A1.

13.21.2.9CMPRA1 Register

Counter/Timer A1 Compare Registers

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x40008024

This contains the Compare limits for timer 1 A half.

Table 837: 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 838: 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.

13.21.2.10CMPRB1 Register

Counter/Timer B1 Compare Registers

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x40008028

This contains the Compare limits for timer 1 B half.

Table 839: 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 840: 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.

13.21.2.11CTRL1 Register

Counter/Timer Control

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4000802C

This includes the Control bit fields for both halves of timer 1.

Table 841: 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		TMRB1POL	TMRB1CLR	TMRB1IE1	TMRB1IE0	TMRB1FN			TMRB1CLK					TMRB1EN	RSVD			TMRA1POL	TMRA1CLR	TMRA1IE1	TMRA1IE0	TMRA1FN			TMRA1CLK					TMRA1EN

Table 842: 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.
30:29	RSVD	0x0	RO	RESERVED

Table 842: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
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
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B1, assert, count to CMPR1B1, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 842: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB1CLK	0x0	RW	Counter/Timer B1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA1 = 0x14 - Clock source is CTIMERA1 OUT. CTMRA0 = 0x15 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x16 - Clock source is CTIMERB0 OUT. CTMRA2 = 0x17 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x18 - Clock source is CTIMERB2 OUT. CTMRB3 = 0x19 - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1A - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB1EN	0x0	RW	Counter/Timer B1 Enable bit. DIS = 0x0 - Counter/Timer B1 Disable. EN = 0x1 - Counter/Timer B1 Enable.
15:13	RSVD	0x0	RO	RESERVED
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 842: CTRL1 Register Bits

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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A1, assert, count to CMPR1A1, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM TRIGCOPY = 0x7 - Replicate the trigger input DUALTRIGPATTERN = 0x4 - Single pattern, trigger on either edge.
5:1	TMRA1CLK	0x0	RW	Counter/Timer A1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB1 = 0x14 - Clock source is CTIMERB1 OUT. CTMRA0 = 0x15 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x16 - Clock source is CTIMERB0 OUT. CTMRA2 = 0x17 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x18 - Clock source is CTIMERB2 OUT. CTMRB3 = 0x19 - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1A - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.

Table 842: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
0	TMRA1EN	0x0	RW	Counter/Timer A1 Enable bit. DIS = 0x0 - Counter/Timer A1 Disable. EN = 0x1 - Counter/Timer A1 Enable.

13.21.2.12CMPRAUXA1 Register

Counter/Timer A1 Compare Registers

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x40008034

Enhanced compare limits for timer half A. This is valid if timer 1 is set to function 4 and function 5.

Table 843: CMPRAUXA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A1																CMPR2A1															

Table 844: CMPRAUXA1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A1	0x0	RW	Counter/Timer A1 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A1	0x0	RW	Counter/Timer A1 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.13CMPRAUXB1 Register

Counter/Timer B1 Compare Registers

OFFSET: 0x00000038

INSTANCE 0 ADDRESS: 0x40008038

Enhanced compare limits for timer half B. This is valid if timer 1 is set to function 4 and function 5.

Table 845: CMPRAUXB1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B1																CMPR2B1															

Table 846: CMPRAUXB1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B1	0x0	RW	Counter/Timer B1 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B1	0x0	RW	Counter/Timer B1 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.14AUX1 Register

Counter/Timer Auxiliary

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0x4000803C

Control bit fields for both halves of timer 0.

Table 847: AUX1 Register

31	30	29	28	27	26	25	24	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	TMRB1EN23	TMRB1POL23	TMRB1TINV	TMRB1NOSYNC	TMRB1TRIG				RSVD	TMRB1LMT						RSVD	TMRA1EN23	TMRA1POL23	TMRA1TINV	TMRA1NOSYNC	TMRA1TRIG				TMRA1LMT							

Table 848: AUX1 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB1EN23	0x0	RW	Counter/Timer B1 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB1POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB1TINV	0x0	RW	Counter/Timer B1 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger

Table 848: AUX1 Register Bits

Bit	Name	Reset	RW	Description
27	TMRB1NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB1TRIG	0x0	RW	Counter/Timer B1 Trigger Select. DIS = 0x0 - Trigger source is disabled. A1OUT = 0x1 - Trigger source is CTIMERA1 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A6OUT = 0x4 - Trigger source is CTIMERA6 OUT. B6OUT = 0x5 - Trigger source is CTIMERB6 OUT. A0OUT = 0x6 - Trigger source is CTIMERA0 OUT. B0OUT = 0x7 - Trigger source is CTIMERB0 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A4OUT2 = 0xA - Trigger source is CTIMERA4 OUT2. B4OUT2 = 0xB - Trigger source is CTIMERB4 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B5OUT2DUAL = 0xE - Trigger source is CTIMERB5 OUT2, dual edge. A5OUT2DUAL = 0xF - Trigger source is CTIMERA5 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED
21:16	TMRB1LMT	0x0	RW	Counter/Timer B1 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA1EN23	0x0	RW	Counter/Timer A1 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA1POL23	0x0	RW	Counter/Timer A1 Upper output polarity NORMAL = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA1TINV	0x0	RW	Counter/Timer A1 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA1NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock

Table 848: AUX1 Register Bits

Bit	Name	Reset	RW	Description
10:7	TMRA1TRIG	0x0	RW	Counter/Timer A1 Trigger Select. DIS = 0x0 - Trigger source is disabled. B1OUT = 0x1 - Trigger source is CTIMERB1 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A0OUT = 0x4 - Trigger source is CTIMERA0 OUT. B0OUT = 0x5 - Trigger source is CTIMERB0 OUT. A5OUT = 0x6 - Trigger source is CTIMERA5 OUT. B5OUT = 0x7 - Trigger source is CTIMERB5 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A4OUT2 = 0xA - Trigger source is CTIMERA4 OUT2. B4OUT2 = 0xB - Trigger source is CTIMERB4 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B5OUT2DUAL = 0xE - Trigger source is CTIMERB5 OUT2, dual edge. A5OUT2DUAL = 0xF - Trigger source is CTIMERA5 OUT2, dual edge.
6:0	TMRA1LMT	0x0	RW	Counter/Timer A1 Pattern Limit Count.

13.21.2.15TMR2 Register

Counter/Timer Register

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x40008040

This register holds the running time or event count for ctimer 2. This is either for each 16 bit half or for the whole 32 bit count when the pair is linked. If the pair is not linked, they can be running on seperate clocks and are completely independent.

Table 849: 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 850: TMR2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB2	0x0	RO	Counter/Timer B2.
15:0	CTTMRA2	0x0	RO	Counter/Timer A2.

13.21.2.16CMPRA2 Register

Counter/Timer A2 Compare Registers

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0x40008044

This register holds the compare limits for timer 2 A half.

Table 851: 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 852: 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.

13.21.2.17CMPRB2 Register

Counter/Timer B2 Compare Registers

OFFSET: 0x00000048

INSTANCE 0 ADDRESS: 0x40008048

This register holds the compare limits for timer 2 B half.

Table 853: 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 854: 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.

13.21.2.18CTRL2 Register

Counter/Timer Control

OFFSET: 0x0000004C

INSTANCE 0 ADDRESS: 0x4000804C

This register holds the control bit fields for both halves of timer 2.

Table 855: 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		TMRB2POL	TMRB2CLR	TMRB2IE1	TMRB2IE0	TMRB2FN			TMRB2CLK					TMRB2EN	RSVD			TMRA2POL	TMRA2CLR	TMRA2IE1	TMRA2IE0	TMRA2FN			TMRA2CLK					TMRA2EN

Table 856: 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.
30:29	RSVD	0x0	RO	RESERVED
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.
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

Table 856: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B2, assert, count to CMPR1B2, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM
21:17	TMRB2CLK	0x0	RW	Counter/Timer B2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA2 = 0x14 - Clock source is CTIMERA2 OUT. CTMRB3 = 0x15 - Clock source is CTIMERA3 OUT. CTMRA3 = 0x16 - Clock source is CTIMERB3 OUT. CTMRA4 = 0x17 - Clock source is CTIMERA4 OUT. CTMRB4 = 0x18 - Clock source is CTIMERB4 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB2EN	0x0	RW	Counter/Timer B2 Enable bit. DIS = 0x0 - Counter/Timer B2 Disable. EN = 0x1 - Counter/Timer B2 Enable.
15:13	RSVD	0x0	RO	RESERVED

Table 856: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
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.
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A2, assert, count to CMPR1A2, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 856: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
5:1	TMRA2CLK	0x0	RW	Counter/Timer A2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB2 = 0x14 - Clock source is CTIMERB2 OUT. CTMRB3 = 0x15 - Clock source is CTIMERA3 OUT. CTMRA3 = 0x16 - Clock source is CTIMERB3 OUT. CTMRA4 = 0x17 - Clock source is CTIMERA4 OUT. CTMRB4 = 0x18 - Clock source is CTIMERB4 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
0	TMRA2EN	0x0	RW	Counter/Timer A2 Enable bit. DIS = 0x0 - Counter/Timer A2 Disable. EN = 0x1 - Counter/Timer A2 Enable.

13.21.2.19CMPRAUXA2 Register

Counter/Timer A2 Compare Registers

OFFSET: 0x00000054

INSTANCE 0 ADDRESS: 0x40008054

Enhanced compare limits for timer half A.

Table 857: CMPRAUXA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A2																CMPR2A2															

Table 858: CMPRAUXA2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A2	0x0	RW	Counter/Timer A2 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A2	0x0	RW	Counter/Timer A2 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.20CMPRAUXB2 Register

Counter/Timer B2 Compare Registers

OFFSET: 0x00000058

INSTANCE 0 ADDRESS: 0x40008058

Enhanced compare limits for timer half B.

Table 859: CMPRAUXB2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B2																CMPR2B2															

Table 860: CMPRAUXB2 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B2	0x0	RW	Counter/Timer B2 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B2	0x0	RW	Counter/Timer B2 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.21AUX2 Register

Counter/Timer Auxiliary

OFFSET: 0x0000005C

INSTANCE 0 ADDRESS: 0x4000805C

Control bit fields for both halves of timer 0.

Table 861: AUX2 Register

31	30	29	28	27	26	25	24	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	TMRB2EN23	TMRB2POL23	TMRB2TINV	TMRB2NOSYNC	TMRB2TRIG				RSVD	TMRB2LMT						RSVD	TMRA2EN23	TMRA2POL23	TMRA2TINV	TMRA2NOSYNC	TMRA2TRIG				TMRA2LMT							

Table 862: AUX2 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB2EN23	0x0	RW	Counter/Timer B2 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB2POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB2TINV	0x0	RW	Counter/Timer B2 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
27	TMRB2NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB2TRIG	0x0	RW	Counter/Timer B2 Trigger Select. DIS = 0x0 - Trigger source is disabled. A2OUT = 0x1 - Trigger source is CTIMERA2 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A1OUT = 0x4 - Trigger source is CTIMERA1 OUT. B1OUT = 0x5 - Trigger source is CTIMERB1 OUT. A4OUT = 0x6 - Trigger source is CTIMERA4 OUT. B4OUT = 0x7 - Trigger source is CTIMERB4 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A5OUT2 = 0xA - Trigger source is CTIMERA5 OUT2. B5OUT2 = 0xB - Trigger source is CTIMERB5 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED

Table 862: AUX2 Register Bits

Bit	Name	Reset	RW	Description
21:16	TMRB2LMT	0x0	RW	Counter/Timer B2 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA2EN23	0x0	RW	Counter/Timer A2 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA2POL23	0x0	RW	Counter/Timer A2 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA2TINV	0x0	RW	Counter/Timer A2 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA2NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA2TRIG	0x0	RW	Counter/Timer A2 Trigger Select. DIS = 0x0 - Trigger source is disabled. B2OUT = 0x1 - Trigger source is CTIMERB2 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A0OUT = 0x4 - Trigger source is CTIMERA0 OUT. B0OUT = 0x5 - Trigger source is CTIMERB0 OUT. A4OUT = 0x6 - Trigger source is CTIMERA4 OUT. B4OUT = 0x7 - Trigger source is CTIMERB4 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A5OUT2 = 0xA - Trigger source is CTIMERA5 OUT2. B5OUT2 = 0xB - Trigger source is CTIMERB5 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge.
6:0	TMRA2LMT	0x0	RW	Counter/Timer A2 Pattern Limit Count.

13.21.2.22TMR3 Register

Counter/Timer Register

OFFSET: 0x00000060

INSTANCE 0 ADDRESS: 0x40008060

Counter/Timer Register

Table 863: 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 864: TMR3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB3	0x0	RO	Counter/Timer B3.
15:0	CTTMRA3	0x0	RO	Counter/Timer A3.

13.21.2.23CMPRA3 Register

Counter/Timer A3 Compare Registers

OFFSET: 0x00000064

INSTANCE 0 ADDRESS: 0x40008064

This register holds the compare limits for timer half A.

Table 865: 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 866: 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.

13.21.2.24CMPRB3 Register

Counter/Timer B3 Compare Registers

OFFSET: 0x00000068

INSTANCE 0 ADDRESS: 0x40008068

This register holds the compare limits for timer half B.

Table 867: 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 868: 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.

13.21.2.25CTRL3 Register

Counter/Timer Control

OFFSET: 0x0000006C

INSTANCE 0 ADDRESS: 0x4000806C

This register holds the control bit fields for both halves of timer 3.

Table 869: 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		TMRB3POL	TMRB3CLR	TMRB3IE1	TMRB3IE0	TMRB3FN		TMRB3CLK						TMRB3EN	ADCEN	RSVD		TMRA3POL	TMRA3CLR	TMRA3IE1	TMRA3IE0	TMRA3FN		TMRA3CLK						TMRA3EN

Table 870: 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.
30:29	RSVD	0x0	RO	RESERVED

Table 870: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
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.
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B3, assert, count to CMPR1B3, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 870: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB3CLK	0x0	RW	Counter/Timer B3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA3 = 0x14 - Clock source is CTIMERA3 OUT. CTMRA2 = 0x15 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x16 - Clock source is CTIMERB2 OUT. CTMRA4 = 0x17 - Clock source is CTIMERA4 OUT. CTMRB4 = 0x18 - Clock source is CTIMERB4 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
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.
14:13	RSVD	0x0	RO	RESERVED
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.

Table 870: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
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, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A3, assert, count to CMPR1A3, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 870: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
5:1	TMRA3CLK	0x0	RW	Counter/Timer A3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB3 = 0x14 - Clock source is CTIMERB3 OUT. CTMRA2 = 0x15 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x16 - Clock source is CTIMERB2 OUT. CTMRA4 = 0x17 - Clock source is CTIMERA4 OUT. CTMRB4 = 0x18 - Clock source is CTIMERB4 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB5 = 0x1B - Clock source is CTIMERB5 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
0	TMRA3EN	0x0	RW	Counter/Timer A3 Enable bit. DIS = 0x0 - Counter/Timer A3 Disable. EN = 0x1 - Counter/Timer A3 Enable.

13.21.2.26CMPRAUXA3 Register

Counter/Timer A3 Compare Registers

OFFSET: 0x00000074

INSTANCE 0 ADDRESS: 0x40008074

Enhanced compare limits for timer half A.

Table 871: CMPRAUXA3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A3																CMPR2A3															

Table 872: CMPRAUXA3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A3	0x0	RW	Counter/Timer A3 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A3	0x0	RW	Counter/Timer A3 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.27CMPRAUXB3 Register

Counter/Timer B3 Compare Registers

OFFSET: 0x00000078

INSTANCE 0 ADDRESS: 0x40008078

Enhanced compare limits for timer half B.

Table 873: CMPRAUXB3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B3																CMPR2B3															

Table 874: CMPRAUXB3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B3	0x0	RW	Counter/Timer B3 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B3	0x0	RW	Counter/Timer B3 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.28AUX3 Register

Counter/Timer Auxiliary

OFFSET: 0x0000007C

INSTANCE 0 ADDRESS: 0x4000807C

Control bit fields for both halves of timer 0.

Table 875: AUX3 Register

31	30	29	28	27	26	25	24	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	TMRB3EN23	TMRB3POL23	TMRB3TINV	TMRB3NOSYNC	TMRB3TRIG				RSVD	TMRB3LMT						RSVD	TMRA3EN23	TMRA3POL23	TMRA3TINV	TMRA3NOSYNC	TMRA3TRIG				TMRA3LMT						

Table 876: AUX3 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB3EN23	0x0	RW	Counter/Timer B3 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB3POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB3TINV	0x0	RW	Counter/Timer B3 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
27	TMRB3NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB3TRIG	0x0	RW	Counter/Timer B3 Trigger Select. DIS = 0x0 - Trigger source is disabled. A3OUT = 0x1 - Trigger source is CTIMERA3 OUT. B2OUT = 0x2 - Trigger source is CTIMERB2 OUT. A2OUT = 0x3 - Trigger source is CTIMERA2 OUT. A4OUT = 0x4 - Trigger source is CTIMERA4 OUT. B4OUT = 0x5 - Trigger source is CTIMERB4 OUT. A6OUT = 0x6 - Trigger source is CTIMERA6 OUT. B6OUT = 0x7 - Trigger source is CTIMERB6 OUT. B5OUT2 = 0x8 - Trigger source is CTIMERB5 OUT2. A5OUT2 = 0x9 - Trigger source is CTIMERA5 OUT2. A1OUT2 = 0xA - Trigger source is CTIMERA1 OUT2. B1OUT2 = 0xB - Trigger source is CTIMERB1 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B2OUT2DUAL = 0xE - Trigger source is CTIMERB2 OUT2, dual edge. A2OUT2DUAL = 0xF - Trigger source is CTIMERA2 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED

Table 876: AUX3 Register Bits

Bit	Name	Reset	RW	Description
21:16	TMRB3LMT	0x0	RW	Counter/Timer B3 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA3EN23	0x0	RW	Counter/Timer A3 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA3POL23	0x0	RW	Counter/Timer A3 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA3TINV	0x0	RW	Counter/Timer A3 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA3NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA3TRIG	0x0	RW	Counter/Timer A3 Trigger Select. DIS = 0x0 - Trigger source is disabled. B3OUT = 0x1 - Trigger source is CTIMERB3 OUT. B2OUT = 0x2 - Trigger source is CTIMERB2 OUT. A2OUT = 0x3 - Trigger source is CTIMERA2 OUT. A4OUT = 0x4 - Trigger source is CTIMERA4 OUT. B4OUT = 0x5 - Trigger source is CTIMERB4 OUT. A7OUT = 0x6 - Trigger source is CTIMERA7 OUT. B7OUT = 0x7 - Trigger source is CTIMERB7 OUT. B5OUT2 = 0x8 - Trigger source is CTIMERB5 OUT2. A5OUT2 = 0x9 - Trigger source is CTIMERA5 OUT2. A1OUT2 = 0xA - Trigger source is CTIMERA1 OUT2. B1OUT2 = 0xB - Trigger source is CTIMERB1 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B2OUT2DUAL = 0xE - Trigger source is CTIMERB2 OUT2, dual edge. A2OUT2DUAL = 0xF - Trigger source is CTIMERA2 OUT2, dual edge.
6:0	TMRA3LMT	0x0	RW	Counter/Timer A3 Pattern Limit Count.

13.21.2.29TMR4 Register

Counter/Timer Register

OFFSET: 0x00000080

INSTANCE 0 ADDRESS: 0x40008080

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 877: TMR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB4																CTTMRA4															

Table 878: TMR4 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB4	0x0	RO	Counter/Timer B4.
15:0	CTTMRA4	0x0	RO	Counter/Timer A4.

13.21.2.30CMPRA4 Register

Counter/Timer A4 Compare Registers

OFFSET: 0x00000084

INSTANCE 0 ADDRESS: 0x40008084

Compare limits for timer half A.

Table 879: CMPRA4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A4																CMPR0A4															

Table 880: CMPRA4 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A4	0x0	RW	Counter/Timer A4 Compare Register 1. Holds the upper limit for timer half A.
15:0	CMPR0A4	0x0	RW	Counter/Timer A4 Compare Register 0. Holds the lower limit for timer half A.

13.21.2.31CMPRB4 Register

Counter/Timer B4 Compare Registers

OFFSET: 0x00000088

INSTANCE 0 ADDRESS: 0x40008088

Compare limits for timer half B.

Table 881: CMPRB4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B4																CMPR0B4															

Table 882: CMPRB4 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B4	0x0	RW	Counter/Timer B4 Compare Register 1. Holds the upper limit for timer half B.
15:0	CMPR0B4	0x0	RW	Counter/Timer B4 Compare Register 0. Holds the lower limit for timer half B.

13.21.2.32CTRL4 Register

Counter/Timer Control

OFFSET: 0x0000008C

INSTANCE 0 ADDRESS: 0x4000808C

Control bit fields for both halves of timer 4.

Table 883: CTRL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK4	RSVD		TMRB4POL	TMRB4CLR	TMRB4IE1	TMRB4IE0	TMRB4FN			TMRB4CLK					TMRB4EN	RSVD			TMRA4POL	TMRA4CLR	TMRA4IE1	TMRA4IE0	TMRA4FN			TMRA4CLK					TMRA4EN

Table 884: CTRL4 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK4	0x0	RW	Counter/Timer A4/B4 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A4/B4 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A4/B4 timers into a single 32-bit timer.
30:29	RSVD	0x0	RO	RESERVED

Table 884: CTRL4 Register Bits

Bit	Name	Reset	RW	Description
28	TMRB4POL	0x0	RW	Counter/Timer B4 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB4 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB4 pin is the inverse of the timer output.
27	TMRB4CLR	0x0	RW	Counter/Timer B4 Clear bit. RUN = 0x0 - Allow counter/timer B4 to run CLEAR = 0x1 - Holds counter/timer B4 at 0x0000.
26	TMRB4IE1	0x0	RW	Counter/Timer B4 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B4 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B4 to generate an interrupt based on COMPR1.
25	TMRB4IE0	0x0	RW	Counter/Timer B4 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B4 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B4 to generate an interrupt based on COMPR0
24:22	TMRB4FN	0x0	RW	Counter/Timer B4 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B4, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B4, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B4, assert, count to CMPR1B4, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B4, assert, count to CMPR1B4, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 884: CTRL4 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB4CLK	0x0	RW	Counter/Timer B4 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA4 = 0x14 - Clock source is CTIMERA4 OUT. CTMRA1 = 0x15 - Clock source is CTIMERA1 OUT. CTMRB1 = 0x16 - Clock source is CTIMERB1 OUT. CTMRA5 = 0x17 - Clock source is CTIMERA5 OUT. CTMRB5 = 0x18 - Clock source is CTIMERB5 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB2 = 0x1A - Clock source is CTIMERB2 OUT. CTMRB3 = 0x1B - Clock source is CTIMERB3 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB4EN	0x0	RW	Counter/Timer B4 Enable bit. DIS = 0x0 - Counter/Timer B4 Disable. EN = 0x1 - Counter/Timer B4 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA4POL	0x0	RW	Counter/Timer A4 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA4 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA4 pin is the inverse of the timer output.
11	TMRA4CLR	0x0	RW	Counter/Timer A4 Clear bit. RUN = 0x0 - Allow counter/timer A4 to run CLEAR = 0x1 - Holds counter/timer A4 at 0x0000.
10	TMRA4IE1	0x0	RW	Counter/Timer A4 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A4 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A4 to generate an interrupt based on COMPR1.

Table 884: CTRL4 Register Bits

Bit	Name	Reset	RW	Description
9	TMRA4IE0	0x0	RW	Counter/Timer A4 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A4 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A4 to generate an interrupt based on COMPR0.
8:6	TMRA4FN	0x0	RW	Counter/Timer A4 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A4, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A4, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A4, assert, count to CMPR1A4, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A4, assert, count to CMPR1A4, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM
5:1	TMRA4CLK	0x0	RW	Counter/Timer A4 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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. (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB4 = 0x14 - Clock source is CTIMERB4 OUT. CTMRA1 = 0x15 - Clock source is CTIMERA1 OUT. CTMRB1 = 0x16 - Clock source is CTIMERB1 OUT. CTMRA5 = 0x17 - Clock source is CTIMERA5 OUT. CTMRB5 = 0x18 - Clock source is CTIMERB5 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB2 = 0x1A - Clock source is CTIMERB2 OUT. CTMRB3 = 0x1B - Clock source is CTIMERB3 OUT. CTMRB6 = 0x1C - Clock source is CTIMERB6 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.

Table 884: CTRL4 Register Bits

Bit	Name	Reset	RW	Description
0	TMRA4EN	0x0	RW	Counter/Timer A4 Enable bit. DIS = 0x0 - Counter/Timer A4 Disable. EN = 0x1 - Counter/Timer A4 Enable.

13.21.2.33CMPRAUXA4 Register

Counter/Timer A4 Compare Registers

OFFSET: 0x00000094

INSTANCE 0 ADDRESS: 0x40008094

Enhanced compare limits for timer half A.

Table 885: CMPRAUXA4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A4																CMPR2A4															

Table 886: CMPRAUXA4 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A4	0x0	RW	Counter/Timer A4 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A4	0x0	RW	Counter/Timer A4 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.34CMPRAUXB4 Register

Counter/Timer B4 Compare Registers

OFFSET: 0x00000098

INSTANCE 0 ADDRESS: 0x40008098

Enhanced compare limits for timer half B.

Table 887: CMPRAUXB4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B4																CMPR2B4															

Table 888: CMPRAUXB4 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B4	0x0	RW	Counter/Timer B4 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B4	0x0	RW	Counter/Timer B4 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.35AUX4 Register

Counter/Timer Auxiliary

OFFSET: 0x0000009C

INSTANCE 0 ADDRESS: 0x4000809C

Control bit fields for both halves of timer 4.

Table 889: AUX4 Register

31	30	29	28	27	26	25	24	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	TMRB4EN23	TMRB4POL23	TMRB4TINV	TMRB4NOSYNC	TMRB4TRIG				RSVD	TMRB4LMT						RSVD	TMRA4EN23	TMRA4POL23	TMRA4TINV	TMRA4NOSYNC	TMRA4TRIG				TMRA4LMT						

Table 890: AUX4 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB4EN23	0x0	RW	Counter/Timer B4 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB4POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB4TINV	0x0	RW	Counter/Timer B4 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger

Table 890: AUX4 Register Bits

Bit	Name	Reset	RW	Description
27	TMRB4NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB4TRIG	0x0	RW	Counter/Timer B4 Trigger Select. DIS = 0x0 - Trigger source is disabled. A4OUT = 0x1 - Trigger source is CTIMERA4 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A7OUT = 0x4 - Trigger source is CTIMERA7 OUT. B7OUT = 0x5 - Trigger source is CTIMERB7 OUT. A1OUT = 0x6 - Trigger source is CTIMERA1 OUT. B1OUT = 0x7 - Trigger source is CTIMERB1 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A1OUT2 = 0xA - Trigger source is CTIMERA1 OUT2. B1OUT2 = 0xB - Trigger source is CTIMERB1 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B5OUT2DUAL = 0xE - Trigger source is CTIMERB5 OUT2, dual edge. A5OUT2DUAL = 0xF - Trigger source is CTIMERA5 OUT2, dual edge. STIMERCAP0 = 0x4 - Trigger source is STimer Capture0 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCAP1 = 0x5 - Trigger source is STimer Capture1 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCAP2 = 0x6 - Trigger source is STimer Capture2 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCAP3 = 0x7 - Trigger source is STimer Capture3 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP0 = 0x8 - Trigger source is STimer Compare0 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP1 = 0x9 - Trigger source is STimer Compare1 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP2 = 0xA - Trigger source is STimer Compare2 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP3 = 0xB - Trigger source is STimer Compare3 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP4 = 0xC - Trigger source is STimer Compare4 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP5 = 0xD - Trigger source is STimer Compare5 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP6 = 0xE - Trigger source is STimer Compare6 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0) STIMERCMP7 = 0xF - Trigger source is STimer Compare7 Interrupt. When CTLINK==1 and TMRA4TRIG==1. (Apollo3 - B0)
22	RSVD	0x0	RO	RESERVED
21:16	TMRB4LMT	0x0	RW	Counter/Timer B4 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA4EN23	0x0	RW	Counter/Timer A4 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.

Table 890: AUX4 Register Bits

Bit	Name	Reset	RW	Description
13	TMRA4POL23	0x0	RW	Counter/Timer A4 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA4TINV	0x0	RW	Counter/Timer A4 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA4NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA4TRIG	0x0	RW	Counter/Timer A4 Trigger Select. DIS = 0x0 - Trigger source is disabled. STIMER = 0x1 - Trigger source is STimer Interrupt. Only Active When CTLINK==1 and TMRB4TRIG!=0. TMRB4TRIG selects an STIMER interrupt B4OUT = 0x1 - Trigger source is CTIMERB4 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A6OUT = 0x4 - Trigger source is CTIMERA6 OUT. B6OUT = 0x5 - Trigger source is CTIMERB6 OUT. A2OUT = 0x6 - Trigger source is CTIMERA2 OUT. B2OUT = 0x7 - Trigger source is CTIMERB2 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A1OUT2 = 0xA - Trigger source is CTIMERA1 OUT2. B1OUT2 = 0xB - Trigger source is CTIMERB1 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B5OUT2DUAL = 0xE - Trigger source is CTIMERB5 OUT2, dual edge. A5OUT2DUAL = 0xF - Trigger source is CTIMERA5 OUT2, dual edge.
6:0	TMRA4LMT	0x0	RW	Counter/Timer A4 Pattern Limit Count.

13.21.2.36TMR5 Register

Counter/Timer Register

OFFSET: 0x000000A0

INSTANCE 0 ADDRESS: 0x400080A0

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 891: TMR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB5																CTTMRA5															

Table 892: TMR5 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB5	0x0	RO	Counter/Timer B5.
15:0	CTTMRA5	0x0	RO	Counter/Timer A5.

13.21.2.37CMPRA5 Register

Counter/Timer A5 Compare Registers

OFFSET: 0x000000A4

INSTANCE 0 ADDRESS: 0x400080A4

This register holds the compare limits for timer half A.

Table 893: CMPRA5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A5																CMPR0A5															

Table 894: CMPRA5 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A5	0x0	RW	Counter/Timer A5 Compare Register 1.
15:0	CMPR0A5	0x0	RW	Counter/Timer A5 Compare Register 0.

13.21.2.38CMPRB5 Register

Counter/Timer B5 Compare Registers

OFFSET: 0x000000A8

INSTANCE 0 ADDRESS: 0x400080A8

This register holds the compare limits for timer half B.

Table 895: CMPRB5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B5																CMPR0B5															

Table 896: CMPRB5 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B5	0x0	RW	Counter/Timer B5 Compare Register 1.
15:0	CMPR0B5	0x0	RW	Counter/Timer B5 Compare Register 0.

13.21.2.39CTRL5 Register

Counter/Timer Control

OFFSET: 0x000000AC

INSTANCE 0 ADDRESS: 0x400080AC

Control bit fields for both halves of timer 0.

Table 897: CTRL5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK5	RSVD		TMRB5POL	TMRB5CLR	TMRB5IE1	TMRB5IE0	TMRB5FN			TMRB5CLK					TMRB5EN	RSVD			TMRA5POL	TMRA5CLR	TMRA5IE1	TMRA5IE0	TMRA5FN			TMRA5CLK					TMRA5EN

Table 898: CTRL5 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK5	0x0	RW	Counter/Timer A5/B5 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A5/B5 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A5/B5 timers into a single 32-bit timer.
30:29	RSVD	0x0	RO	RESERVED
28	TMRB5POL	0x0	RW	Counter/Timer B5 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB5 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB5 pin is the inverse of the timer output.
27	TMRB5CLR	0x0	RW	Counter/Timer B5 Clear bit. RUN = 0x0 - Allow counter/timer B5 to run CLEAR = 0x1 - Holds counter/timer B5 at 0x0000.

Table 898: CTRL5 Register Bits

Bit	Name	Reset	RW	Description
26	TMRB5IE1	0x0	RW	Counter/Timer B5 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B5 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B5 to generate an interrupt based on COMPR1.
25	TMRB5IE0	0x0	RW	Counter/Timer B5 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B5 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B5 to generate an interrupt based on COMPR0
24:22	TMRB5FN	0x0	RW	Counter/Timer B5 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B5, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B5, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B5, assert, count to CMPR1B5, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B5, assert, count to CMPR1B5, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 898: CTRL5 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB5CLK	0x0	RW	Counter/Timer B5 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA5 = 0x14 - Clock source is CTIMERA5 OUT. CTMRA0 = 0x15 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x16 - Clock source is CTIMERB0 OUT. CTMRA6 = 0x17 - Clock source is CTIMERA6 OUT. CTMRB6 = 0x18 - Clock source is CTIMERB6 OUT. CTMRB1 = 0x19 - Clock source is CTIMERB1 OUT. CTMRB2 = 0x1A - Clock source is CTIMERB2 OUT. CTMRB3 = 0x1B - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1C - Clock source is CTIMERB4 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB5EN	0x0	RW	Counter/Timer B5 Enable bit. DIS = 0x0 - Counter/Timer B5 Disable. EN = 0x1 - Counter/Timer B5 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA5POL	0x0	RW	Counter/Timer A5 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA5 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA5 pin is the inverse of the timer output.
11	TMRA5CLR	0x0	RW	Counter/Timer A5 Clear bit. RUN = 0x0 - Allow counter/timer A5 to run CLEAR = 0x1 - Holds counter/timer A5 at 0x0000.
10	TMRA5IE1	0x0	RW	Counter/Timer A5 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A5 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A5 to generate an interrupt based on COMPR1.

Table 898: CTRL5 Register Bits

Bit	Name	Reset	RW	Description
9	TMRA5IE0	0x0	RW	Counter/Timer A5 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A5 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A5 to generate an interrupt based on COMPR0.
8:6	TMRA5FN	0x0	RW	Counter/Timer A5 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A5, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A5, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A5, assert, count to CMPR1A5, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A5, assert, count to CMPR1A5, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM TRIGCOPY = 0x7 - Replicate the trigger input DUALTRIGPATTERN = 0x4 - Single pattern, trigger on either edge.
5:1	TMRA5CLK	0x0	RW	Counter/Timer A5 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB5 = 0x14 - Clock source is CTIMERB5 OUT. CTMRA0 = 0x15 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x16 - Clock source is CTIMERB0 OUT. CTMRA6 = 0x17 - Clock source is CTIMERA6 OUT. CTMRB6 = 0x18 - Clock source is CTIMERB6 OUT. CTMRB1 = 0x19 - Clock source is CTIMERB1 OUT. CTMRB2 = 0x1A - Clock source is CTIMERB2 OUT. CTMRB3 = 0x1B - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1C - Clock source is CTIMERB4 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.

Table 898: CTRL5 Register Bits

Bit	Name	Reset	RW	Description
0	TMRA5EN	0x0	RW	Counter/Timer A5 Enable bit. DIS = 0x0 - Counter/Timer A5 Disable. EN = 0x1 - Counter/Timer A5 Enable.

13.21.2.40CMPRAUXA5 Register

Counter/Timer A5 Compare Registers

OFFSET: 0x000000B4

INSTANCE 0 ADDRESS: 0x400080B4

Enhanced compare limits for timer half A.

Table 899: CMPRAUXA5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A5																CMPR2A5															

Table 900: CMPRAUXA5 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A5	0x0	RW	Counter/Timer A5 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A5	0x0	RW	Counter/Timer A5 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.41CMPRAUXB5 Register

Counter/Timer B5 Compare Registers

OFFSET: 0x000000B8

INSTANCE 0 ADDRESS: 0x400080B8

Enhanced compare limits for timer half B.

Table 901: CMPRAUXB5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B5																CMPR2B5															

Table 902: CMPRAUXB5 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B5	0x0	RW	Counter/Timer B5 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B5	0x0	RW	Counter/Timer B5 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.42AUX5 Register

Counter/Timer Auxiliary

OFFSET: 0x000000BC

INSTANCE 0 ADDRESS: 0x400080BC

Control bit fields for both halves of timer 0.

Table 903: AUX5 Register

31	30	29	28	27	26	25	24	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	TMRB5EN23	TMRB5POL23	TMRB5TINV	TMRB5NOSYNC	TMRB5TRIG				RSVD	TMRB5LMT						RSVD	TMRA5EN23	TMRA5POL23	TMRA5TINV	TMRA5NOSYNC	TMRA5TRIG				TMRA5LMT						

Table 904: AUX5 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB5EN23	0x0	RW	Counter/Timer B5 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB5POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB5TINV	0x0	RW	Counter/Timer B5 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger

Table 904: AUX5 Register Bits

Bit	Name	Reset	RW	Description
27	TMRB5NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB5TRIG	0x0	RW	Counter/Timer B5 Trigger Select. DIS = 0x0 - Trigger source is disabled. A5OUT = 0x1 - Trigger source is CTIMERA5 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A6OUT = 0x4 - Trigger source is CTIMERA6 OUT. B6OUT = 0x5 - Trigger source is CTIMERB6 OUT. A1OUT = 0x6 - Trigger source is CTIMERA1 OUT. B1OUT = 0x7 - Trigger source is CTIMERB1 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A0OUT2 = 0xA - Trigger source is CTIMERA0 OUT2. B0OUT2 = 0xB - Trigger source is CTIMERB0 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge. STIMERCAP0 = 0x4 - Trigger source is STimer Capture0 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCAP1 = 0x5 - Trigger source is STimer Capture1 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCAP2 = 0x6 - Trigger source is STimer Capture2 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCAP3 = 0x7 - Trigger source is STimer Capture3 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP0 = 0x8 - Trigger source is STimer Compare0 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP1 = 0x9 - Trigger source is STimer Compare1 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP2 = 0xA - Trigger source is STimer Compare2 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP3 = 0xB - Trigger source is STimer Compare3 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP4 = 0xC - Trigger source is STimer Compare4 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP5 = 0xD - Trigger source is STimer Compare5 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP6 = 0xE - Trigger source is STimer Compare6 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0) STIMERCMP7 = 0xF - Trigger source is STimer Compare7 Interrupt. When CTLINK==1 and TMRA5TRIG==1. (Apollo3 - B0)
22	RSVD	0x0	RO	RESERVED
21:16	TMRB5LMT	0x0	RW	Counter/Timer B5 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA5EN23	0x0	RW	Counter/Timer A5 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.

Table 904: AUX5 Register Bits

Bit	Name	Reset	RW	Description
13	TMRA5POL23	0x0	RW	Counter/Timer A5 Upper output polarity NORMAL = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA5TINV	0x0	RW	Counter/Timer A5 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA5NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA5TRIG	0x0	RW	Counter/Timer A5 Trigger Select. DIS = 0x0 - Trigger source is disabled. STIMER = 0x1 - Trigger source is STimer Interrupt. Only Active When CTLINK==1 and TMRB5TRIG!=0. TMRB5TRIG selects an STIMER interrupt B5OUT = 0x1 - Trigger source is CTIMERB5 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A4OUT = 0x4 - Trigger source is CTIMERA4 OUT. B4OUT = 0x5 - Trigger source is CTIMERB4 OUT. A2OUT = 0x6 - Trigger source is CTIMERA2 OUT. B2OUT = 0x7 - Trigger source is CTIMERB2 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A0OUT2 = 0xA - Trigger source is CTIMERA0 OUT2. B0OUT2 = 0xB - Trigger source is CTIMERB0 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge.
6:0	TMRA5LMT	0x0	RW	Counter/Timer A5 Pattern Limit Count.

13.21.2.43TMR6 Register

Counter/Timer Register

OFFSET: 0x000000C0

INSTANCE 0 ADDRESS: 0x400080C0

Counter/Timer Register

Table 905: TMR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB6																CTTMRA6															

Table 906: TMR6 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB6	0x0	RO	Counter/Timer B6.
15:0	CTTMRA6	0x0	RO	Counter/Timer A6.

13.21.2.44CMPRA6 Register

Counter/Timer A6 Compare Registers

OFFSET: 0x000000C4

INSTANCE 0 ADDRESS: 0x400080C4

This register holds the compare limits for timer half A.

Table 907: CMPRA6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A6																CMPR0A6															

Table 908: CMPRA6 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A6	0x0	RW	Counter/Timer A6 Compare Register 1.
15:0	CMPR0A6	0x0	RW	Counter/Timer A6 Compare Register 0.

13.21.2.45CMPRB6 Register

Counter/Timer B6 Compare Registers

OFFSET: 0x000000C8

INSTANCE 0 ADDRESS: 0x400080C8

This register holds the compare limits for timer half B.

Table 909: CMPRB6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B6																CMPR0B6															

Table 910: CMPRB6 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B6	0x0	RW	Counter/Timer B6 Compare Register 1.
15:0	CMPR0B6	0x0	RW	Counter/Timer B6 Compare Register 0.

13.21.2.46CTRL6 Register

Counter/Timer Control

OFFSET: 0x000000CC

INSTANCE 0 ADDRESS: 0x400080CC

This register holds the control bit fields for both halves of timer 6.

Table 911: CTRL6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK6	RSVD		TMRB6POL	TMRB6CLR	TMRB6IE1	TMRB6IE0	TMRB6FN			TMRB6CLK					TMRB6EN	RSVD			TMRA6POL	TMRA6CLR	TMRA6IE1	TMRA6IE0	TMRA6FN			TMRA6CLK					TMRA6EN

Table 912: CTRL6 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK6	0x0	RW	Counter/Timer A6/B6 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A6/B6 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A6/B6 timers into a single 32-bit timer.
30:29	RSVD	0x0	RO	RESERVED
28	TMRB6POL	0x0	RW	Counter/Timer B6 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB6 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB6 pin is the inverse of the timer output.
27	TMRB6CLR	0x0	RW	Counter/Timer B6 Clear bit. RUN = 0x0 - Allow counter/timer B6 to run CLEAR = 0x1 - Holds counter/timer B6 at 0x0000.

Table 912: CTRL6 Register Bits

Bit	Name	Reset	RW	Description
26	TMRB6IE1	0x0	RW	Counter/Timer B6 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B6 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B6 to generate an interrupt based on COMPR1.
25	TMRB6IE0	0x0	RW	Counter/Timer B6 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B6 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B6 to generate an interrupt based on COMPR0
24:22	TMRB6FN	0x0	RW	Counter/Timer B6 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B6, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B6, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B6, assert, count to CMPR1B6, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B6, assert, count to CMPR1B6, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 912: CTRL6 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB6CLK	0x0	RW	Counter/Timer B6 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA6 = 0x14 - Clock source is CTIMERA6 OUT. CTMRA3 = 0x15 - Clock source is CTIMERA3 OUT. CTMRB3 = 0x16 - Clock source is CTIMERB3 OUT. CTMRA7 = 0x17 - Clock source is CTIMERA7 OUT. CTMRB7 = 0x18 - Clock source is CTIMERB7 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB2 = 0x1B - Clock source is CTIMERB2 OUT. CTMRB4 = 0x1C - Clock source is CTIMERB4 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB6EN	0x0	RW	Counter/Timer B6 Enable bit. DIS = 0x0 - Counter/Timer B6 Disable. EN = 0x1 - Counter/Timer B6 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA6POL	0x0	RW	Counter/Timer A6 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA6 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA6 pin is the inverse of the timer output.
11	TMRA6CLR	0x0	RW	Counter/Timer A6 Clear bit. RUN = 0x0 - Allow counter/timer A6 to run CLEAR = 0x1 - Holds counter/timer A6 at 0x0000.
10	TMRA6IE1	0x0	RW	Counter/Timer A6 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A6 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A6 to generate an interrupt based on COMPR1.

Table 912: CTRL6 Register Bits

Bit	Name	Reset	RW	Description
9	TMRA6IE0	0x0	RW	Counter/Timer A6 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A6 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A6 to generate an interrupt based on COMPR0.
8:6	TMRA6FN	0x0	RW	Counter/Timer A6 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A6, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A6, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A6, assert, count to CMPR1A6, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A6, assert, count to CMPR1A6, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM
5:1	TMRA6CLK	0x0	RW	Counter/Timer A6 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB6 = 0x14 - Clock source is CTIMERB6 OUT. CTMRA3 = 0x15 - Clock source is CTIMERA3 OUT. CTMRB3 = 0x16 - Clock source is CTIMERB3 OUT. CTMRA7 = 0x17 - Clock source is CTIMERA7 OUT. CTMRB7 = 0x18 - Clock source is CTIMERB7 OUT. CTMRB0 = 0x19 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x1A - Clock source is CTIMERB1 OUT. CTMRB2 = 0x1B - Clock source is CTIMERB2 OUT. CTMRB4 = 0x1C - Clock source is CTIMERB4 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.

Table 912: CTRL6 Register Bits

Bit	Name	Reset	RW	Description
0	TMRA6EN	0x0	RW	Counter/Timer A6 Enable bit. DIS = 0x0 - Counter/Timer A6 Disable. EN = 0x1 - Counter/Timer A6 Enable.

13.21.2.47CMPRAUXA6 Register

Counter/Timer A6 Compare Registers

OFFSET: 0x000000D4

INSTANCE 0 ADDRESS: 0x400080D4

Enhanced compare limits for timer half A.

Table 913: CMPRAUXA6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A6																CMPR2A6															

Table 914: CMPRAUXA6 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A6	0x0	RW	Counter/Timer A6 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A6	0x0	RW	Counter/Timer A6 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.48CMPRAUXB6 Register

Counter/Timer B6 Compare Registers

OFFSET: 0x000000D8

INSTANCE 0 ADDRESS: 0x400080D8

Enhanced compare limits for timer half B.

Table 915: CMPRAUXB6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B6																CMPR2B6															

Table 916: CMPRAUXB6 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B6	0x0	RW	Counter/Timer B6 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B6	0x0	RW	Counter/Timer B6 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.49AUX6 Register

Counter/Timer Auxiliary

OFFSET: 0x000000DC

INSTANCE 0 ADDRESS: 0x400080DC

Control bit fields for both halves of timer 0.

Table 917: AUX6 Register

31	30	29	28	27	26	25	24	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	TMRB6EN23	TMRB6POL23	TMRB6TINV	TMRB6NOSYNC	TMRB6TRIG				RSVD	TMRB6LMT						RSVD	TMRA6EN23	TMRA6POL23	TMRA6TINV	TMRA6NOSYNC	TMRA6TRIG				TMRA6LMT							

Table 918: AUX6 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB6EN23	0x0	RW	Counter/Timer B6 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB6POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB6TINV	0x0	RW	Counter/Timer B6 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger

Table 918: AUX6 Register Bits

Bit	Name	Reset	RW	Description
27	TMRB6NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB6TRIG	0x0	RW	Counter/Timer B6 Trigger Select. DIS = 0x0 - Trigger source is disabled. A6OUT = 0x1 - Trigger source is CTIMERA6 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A4OUT = 0x4 - Trigger source is CTIMERA4 OUT. B4OUT = 0x5 - Trigger source is CTIMERB4 OUT. A1OUT = 0x6 - Trigger source is CTIMERA1 OUT. B1OUT = 0x7 - Trigger source is CTIMERB1 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A2OUT2 = 0xA - Trigger source is CTIMERA2 OUT2. B2OUT2 = 0xB - Trigger source is CTIMERB2 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B0OUT2DUAL = 0xE - Trigger source is CTIMERB0 OUT2, dual edge. A0OUT2DUAL = 0xF - Trigger source is CTIMERA0 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED
21:16	TMRB6LMT	0x0	RW	Counter/Timer B6 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA6EN23	0x0	RW	Counter/Timer A6 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA6POL23	0x0	RW	Counter/Timer A6 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA6TINV	0x0	RW	Counter/Timer A6 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA6NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock

Table 918: AUX6 Register Bits

Bit	Name	Reset	RW	Description
10:7	TMRA6TRIG	0x0	RW	Counter/Timer A6 Trigger Select. DIS = 0x0 - Trigger source is disabled. B6OUT = 0x1 - Trigger source is CTIMERB6 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A5OUT = 0x4 - Trigger source is CTIMERA5 OUT. B5OUT = 0x5 - Trigger source is CTIMERB5 OUT. A1OUT = 0x6 - Trigger source is CTIMERA1 OUT. B1OUT = 0x7 - Trigger source is CTIMERB1 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A2OUT2 = 0xA - Trigger source is CTIMERA2 OUT2. B2OUT2 = 0xB - Trigger source is CTIMERBb OUT2. A5OUT2DUAL = 0xC - Trigger source is CTIMERA5 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B0OUT2DUAL = 0xE - Trigger source is CTIMERB0 OUT2, dual edge. A0OUT2DUAL = 0xF - Trigger source is CTIMERA0 OUT2, dual edge.
6:0	TMRA6LMT	0x0	RW	Counter/Timer A6 Pattern Limit Count.

13.21.2.50TMR7 Register

Counter/Timer Register

OFFSET: 0x000000E0

INSTANCE 0 ADDRESS: 0x400080E0

Counter/Timer Register

Table 919: TMR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTTMRB7																CTTMRA7															

Table 920: TMR7 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB7	0x0	RO	Counter/Timer B7.
15:0	CTTMRA7	0x0	RO	Counter/Timer A7.

13.21.2.51CMPRA7 Register

Counter/Timer A7 Compare Registers

OFFSET: 0x000000E4

INSTANCE 0 ADDRESS: 0x400080E4

This register holds the compare limits for timer half A.

Table 921: CMPRA7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1A7																CMPR0A7															

Table 922: CMPRA7 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1A7	0x0	RW	Counter/Timer A7 Compare Register 1.
15:0	CMPR0A7	0x0	RW	Counter/Timer A7 Compare Register 0.

13.21.2.52CMPRB7 Register

Counter/Timer B7 Compare Registers

OFFSET: 0x000000E8

INSTANCE 0 ADDRESS: 0x400080E8

This register holds the compare limits for timer half B.

Table 923: CMPRB7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR1B7																CMPR0B7															

Table 924: CMPRB7 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR1B7	0x0	RW	Counter/Timer B3 Compare Register 1.
15:0	CMPR0B7	0x0	RW	Counter/Timer B3 Compare Register 0.

13.21.2.53CTRL7 Register

Counter/Timer Control

OFFSET: 0x000000EC

INSTANCE 0 ADDRESS: 0x400080EC

This register holds the control bit fields for both halves of timer 7.

Table 925: CTRL7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CTLINK7	RSVD		TMRB7POL	TMRB7CLR	TMRB7IE1	TMRB7IE0	TMRB7FN			TMRB7CLK					TMRB7EN	RSVD			TMRA7POL	TMRA7CLR	TMRA7IE1	TMRA7IE0	TMRA7FN			TMRA7CLK					TMRA7EN

Table 926: CTRL7 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK7	0x0	RW	Counter/Timer A7/B7 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A7/B7 timers as two independent 16-bit timers (default). 32BIT_TIMER = 0x1 - Link A7/B7 timers into a single 32-bit timer.
30:29	RSVD	0x0	RO	RESERVED
28	TMRB7POL	0x0	RW	Counter/Timer B7 output polarity. NORMAL = 0x0 - The polarity of the TMRPINB7 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB7 pin is the inverse of the timer output.
27	TMRB7CLR	0x0	RW	Counter/Timer B7 Clear bit. RUN = 0x0 - Allow counter/timer B7 to run CLEAR = 0x1 - Holds counter/timer B7 at 0x0000.
26	TMRB7IE1	0x0	RW	Counter/Timer B7 Interrupt Enable bit for COMPR1. DIS = 0x0 - Disable counter/timer B7 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer B7 to generate an interrupt based on COMPR1.
25	TMRB7IE0	0x0	RW	Counter/Timer B7 Interrupt Enable bit for COMPR0. DIS = 0x0 - Disable counter/timer B7 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer B7 to generate an interrupt based on COMPR0

Table 926: CTRL7 Register Bits

Bit	Name	Reset	RW	Description
24:22	TMRB7FN	0x0	RW	Counter/Timer B7 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B7, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B7, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B7, assert, count to CMPR1B7, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B7, assert, count to CMPR1B7, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM
21:17	TMRB7CLK	0x0	RW	Counter/Timer B7 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRA7 = 0x14 - Clock source is CTIMERA7 OUT. CTMRA2 = 0x15 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x16 - Clock source is CTIMERB2 OUT. CTMRA0 = 0x17 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x18 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x19 - Clock source is CTIMERB1 OUT. CTMRB3 = 0x1A - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1B - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1C - Clock source is CTIMERB5 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
16	TMRB7EN	0x0	RW	Counter/Timer B7 Enable bit. DIS = 0x0 - Counter/Timer B7 Disable. EN = 0x1 - Counter/Timer B7 Enable.
15:13	RSVD	0x0	RO	RESERVED

Table 926: CTRL7 Register Bits

Bit	Name	Reset	RW	Description
12	TMRA7POL	0x0	RW	Counter/Timer A7 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA7 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA7 pin is the inverse of the timer output.
11	TMRA7CLR	0x0	RW	Counter/Timer A7 Clear bit. RUN = 0x0 - Allow counter/timer A7 to run CLEAR = 0x1 - Holds counter/timer A7 at 0x0000.
10	TMRA7IE1	0x0	RW	Counter/Timer A7 Interrupt Enable bit based on COMPR1. DIS = 0x0 - Disable counter/timer A7 from generating an interrupt based on COMPR1. EN = 0x1 - Enable counter/timer A7 to generate an interrupt based on COMPR1.
9	TMRA7IE0	0x0	RW	Counter/Timer A7 Interrupt Enable bit based on COMPR0. DIS = 0x0 - Disable counter/timer A7 from generating an interrupt based on COMPR0. EN = 0x1 - Enable counter/timer A7 to generate an interrupt based on COMPR0.
8:6	TMRA7FN	0x0	RW	Counter/Timer A7 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A7, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A7, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A7, assert, count to CMPR1A7, deassert, stop. PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A7, assert, count to CMPR1A7, deassert, restart. SINGLEPATTERN = 0x4 - Single pattern. REPEATPATTERN = 0x5 - Repeated pattern. CONTINUOUS = 0x6 - Continuous run (aka Free Run). Count continuously. ALTPWN = 0x7 - Alternate PWM

Table 926: CTRL7 Register Bits

Bit	Name	Reset	RW	Description
5:1	TMRA7CLK	0x0	RW	Counter/Timer A7 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC_DIV4 = 0x1 - Clock source is the 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_DIV128 = 0x9 - Clock source is XT / 128 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 (note: this clock is only available when MCU is in active mode) XT_DIV4 = 0x10 - Clock source is XT / 4 XT_DIV8 = 0x11 - Clock source is XT / 8 XT_DIV32 = 0x12 - Clock source is XT / 32 RSVD = 0x13 - Clock source is Reserved. CTMRB7 = 0x14 - Clock source is CTIMERB7 OUT. CTMRA2 = 0x15 - Clock source is CTIMERA2 OUT. CTMRB2 = 0x16 - Clock source is CTIMERB2 OUT. CTMRA0 = 0x17 - Clock source is CTIMERA0 OUT. CTMRB0 = 0x18 - Clock source is CTIMERB0 OUT. CTMRB1 = 0x19 - Clock source is CTIMERB1 OUT. CTMRB3 = 0x1A - Clock source is CTIMERB3 OUT. CTMRB4 = 0x1B - Clock source is CTIMERB4 OUT. CTMRB5 = 0x1C - Clock source is CTIMERB5 OUT. BUCKBLE = 0x1D - Clock source is BLE buck converter TON pulses. BUCKB = 0x1E - Clock source is Memory buck converter TON pulses. BUCKA = 0x1F - Clock source is CPU buck converter TON pulses.
0	TMRA7EN	0x0	RW	Counter/Timer A7 Enable bit. DIS = 0x0 - Counter/Timer A7 Disable. EN = 0x1 - Counter/Timer A7 Enable.

13.21.2.54CMPRAUXA7 Register

Counter/Timer A7 Compare Registers

OFFSET: 0x000000F4

INSTANCE 0 ADDRESS: 0x400080F4

Enhanced compare limits for timer half A.

Table 927: CMPRAUXA7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3A7																CMPR2A7															

Table 928: CMPRAUXA7 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3A7	0x0	RW	Counter/Timer A7 Compare Register 3. Holds the upper limit for timer half A.
15:0	CMPR2A7	0x0	RW	Counter/Timer A7 Compare Register 2. Holds the lower limit for timer half A.

13.21.2.55CMPRAUXB7 Register

Counter/Timer B7 Compare Registers

OFFSET: 0x000000F8

INSTANCE 0 ADDRESS: 0x400080F8

Enhanced compare limits for timer half B.

Table 929: CMPRAUXB7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMPR3B7																CMPR2B7															

Table 930: CMPRAUXB7 Register Bits

Bit	Name	Reset	RW	Description
31:16	CMPR3B7	0x0	RW	Counter/Timer B7 Compare Register 3. Holds the upper limit for timer half B.
15:0	CMPR2B7	0x0	RW	Counter/Timer B7 Compare Register 2. Holds the lower limit for timer half B.

13.21.2.56AUX7 Register

Counter/Timer Auxiliary

OFFSET: 0x000000FC

INSTANCE 0 ADDRESS: 0x400080FC

Control bit fields for both halves of timer 0.

Table 931: AUX7 Register

31	30	29	28	27	26	25	24	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	TMRB7EN23	TMRB7POL23	TMRB7TINV	TMRB7NOSYNC	TMRB7TRIG				RSVD	TMRB7LMT						RSVD	TMRA7EN23	TMRA7POL23	TMRA7TINV	TMRA7NOSYNC	TMRA7TRIG				TMRA7LMT							

Table 932: AUX7 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	TMRB7EN23	0x0	RW	Counter/Timer B7 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
29	TMRB7POL23	0x0	RW	Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
28	TMRB7TINV	0x0	RW	Counter/Timer B7 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
27	TMRB7NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
26:23	TMRB7TRIG	0x0	RW	Counter/Timer B7 Trigger Select. DIS = 0x0 - Trigger source is disabled. A7OUT = 0x1 - Trigger source is CTIMERA7 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A5OUT = 0x4 - Trigger source is CTIMERA5 OUT. B5OUT = 0x5 - Trigger source is CTIMERB5 OUT. A2OUT = 0x6 - Trigger source is CTIMERA2 OUT. B2OUT = 0x7 - Trigger source is CTIMERB2 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A2OUT2 = 0xA - Trigger source is CTIMERA2 OUT2. B2OUT2 = 0xB - Trigger source is CTIMERB2 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A7OUT2DUAL = 0xD - Trigger source is CTIMERA7 OUT2, dual edge. B1OUT2DUAL = 0xE - Trigger source is CTIMERB1 OUT2, dual edge. A1OUT2DUAL = 0xF - Trigger source is CTIMERA1 OUT2, dual edge.
22	RSVD	0x0	RO	RESERVED

Table 932: AUX7 Register Bits

Bit	Name	Reset	RW	Description
21:16	TMRB7LMT	0x0	RW	Counter/Timer B7 Pattern Limit Count.
15	RSVD	0x0	RO	RESERVED
14	TMRA7EN23	0x0	RW	Counter/Timer A7 Upper compare enable. DIS = 0x1 - Disable enhanced functions. EN = 0x0 - Enable enhanced functions.
13	TMRA7POL23	0x0	RW	Counter/Timer A7 Upper output polarity NORM = 0x0 - Upper output normal polarity INV = 0x1 - Upper output inverted polarity.
12	TMRA7TINV	0x0	RW	Counter/Timer A7 Invert on trigger. DIS = 0x0 - Disable invert on trigger EN = 0x1 - Enable invert on trigger
11	TMRA7NO-SYNC	0x0	RW	Source clock synchronization control. DIS = 0x0 - Synchronization on source clock NOSYNC = 0x1 - No synchronization on source clock
10:7	TMRA7TRIG	0x0	RW	Counter/Timer A7 Trigger Select. DIS = 0x0 - Trigger source is disabled. B7OUT = 0x1 - Trigger source is CTIMERB7 OUT. B3OUT = 0x2 - Trigger source is CTIMERB3 OUT. A3OUT = 0x3 - Trigger source is CTIMERA3 OUT. A1OUT = 0x4 - Trigger source is CTIMERA1 OUT. B1OUT = 0x5 - Trigger source is CTIMERB1 OUT. A4OUT = 0x6 - Trigger source is CTIMERA4 OUT. B4OUT = 0x7 - Trigger source is CTIMERB4 OUT. B3OUT2 = 0x8 - Trigger source is CTIMERB3 OUT2. A3OUT2 = 0x9 - Trigger source is CTIMERA3 OUT2. A2OUT2 = 0xA - Trigger source is CTIMERA2 OUT2. B2OUT2 = 0xB - Trigger source is CTIMERB2 OUT2. A6OUT2DUAL = 0xC - Trigger source is CTIMERA6 OUT2, dual edge. A5OUT2DUAL = 0xD - Trigger source is CTIMERA5 OUT2, dual edge. B4OUT2DUAL = 0xE - Trigger source is CTIMERB4 OUT2, dual edge. A4OUT2DUAL = 0xF - Trigger source is CTIMERA4 OUT2, dual edge.
6:0	TMRA7LMT	0x0	RW	Counter/Timer A7 Pattern Limit Count.

13.21.2.57GLOBEN Register

Counter/Timer Global Enable

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x40008100

Alternate enables for all CTIMERs.

Table 933: GLOBEN Register

31	30	29	28	27	26	25	24	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																ENB7	ENA7	ENB6	ENA6	ENB5	ENA5	ENB4	ENA4	ENB3	ENA3	ENB2	ENA2	ENB1	ENA1	ENB0	ENA0

Table 934: GLOBEN Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	ENB7	0x1	RW	Alternate enable for B7. LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
14	ENA7	0x1	RW	Alternate enable for A7 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
13	ENB6	0x1	RW	Alternate enable for B6 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
12	ENA6	0x1	RW	Alternate enable for A6 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
11	ENB5	0x1	RW	Alternate enable for B5 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
10	ENA5	0x1	RW	Alternate enable for A5 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
9	ENB4	0x1	RW	Alternate enable for B4 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
8	ENA4	0x1	RW	Alternate enable for A4 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
7	ENB3	0x1	RW	Alternate enable for B3. LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.

Table 934: GLOBEN Register Bits

Bit	Name	Reset	RW	Description
6	ENA3	0x1	RW	Alternate enable for A3 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
5	ENB2	0x1	RW	Alternate enable for B2 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
4	ENA2	0x1	RW	Alternate enable for A2 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
3	ENB1	0x1	RW	Alternate enable for B1 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
2	ENA1	0x1	RW	Alternate enable for A1 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
1	ENB0	0x1	RW	Alternate enable for B0 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.
0	ENA0	0x1	RW	Alternate enable for A0 LCO = 0x1 - Use local enable. DIS = 0x0 - Disable CTIMER.

13.21.2.58OUTCFG0 Register

Counter/Timer Output Config 0

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x40008104

Pad output configuration 0.

Table 935: OUTCFG0 Register

31	30	29	28	27	26	25	24	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	CFG9			CFG8			CFG7			CFG6			CFG5			RSVD	CFG4			CFG3			CFG2			CFG1			CFG0		

Table 936: OUTCFG0 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30:28	CFG9	0x2	RW	Pad output 9 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B0OUT = 0x5 - Output is B0OUT. A4OUT = 0x4 - Output is A4OUT. A2OUT = 0x3 - Output is A2OUT. A2OUT2 = 0x2 - Output is A2OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
27:25	CFG8	0x2	RW	Pad output 8 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B6OUT = 0x5 - Output is B6OUT. A4OUT2 = 0x4 - Output is A4OUT2. A3OUT2 = 0x3 - Output is A3OUT. A2OUT = 0x2 - Output is A2OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
24:22	CFG7	0x2	RW	Pad output 7 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A7OUT = 0x5 - Output is A7OUT. B5OUT = 0x4 - Output is B5OUT. B1OUT = 0x3 - Output is B1OUT. B1OUT2 = 0x2 - Output is B1OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
21:19	CFG6	0x2	RW	Pad output 6 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B7OUT = 0x5 - Output is B7OUT. B5OUT2 = 0x4 - Output is B5OUT2. A1OUT = 0x3 - Output is A1OUT. B1OUT = 0x2 - Output is B1OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
18:16	CFG5	0x2	RW	Pad output 5 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A7OUT = 0x5 - Output is A7OUT. B6OUT = 0x4 - Output is A5OUT. A1OUT = 0x3 - Output is A1OUT. A1OUT2 = 0x2 - Output is A1OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
15	RSVD	0x0	RO	RESERVED

Table 936: OUTCFG0 Register Bits

Bit	Name	Reset	RW	Description
14:12	CFG4	0x2	RW	Pad output 4 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B5OUT = 0x5 - Output is B5OUT. A5OUT2 = 0x4 - Output is A5OUT2. A2OUT2 = 0x3 - Output is A2OUT2. A1OUT = 0x2 - Output is A1OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
11:9	CFG3	0x1	RW	Pad output 3 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A6OUT = 0x5 - Output is A6OUT. A1OUT = 0x4 - Output is A1OUT. B0OUT = 0x3 - Output is B0OUT. B0OUT2 = 0x2 - Output is B0OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
8:6	CFG2	0x2	RW	Pad output 2 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A7OUT = 0x5 - Output is A7OUT. B6OUT2 = 0x4 - Output is B6OUT2. B1OUT2 = 0x3 - Output is B1OUT2. B0OUT = 0x2 - Output is B0OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
5:3	CFG1	0x2	RW	Pad output 1 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B7OUT2 = 0x5 - Output is B7OUT2. A5OUT = 0x4 - Output is A5OUT. A0OUT = 0x3 - Output is A0OUT. A0OUT2 = 0x2 - Output is A0OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
2:0	CFG0	0x2	RW	Pad output 0 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A6OUT = 0x5 - Output is A6OUT. A5OUT2 = 0x4 - Output is A5OUT2. B2OUT2 = 0x3 - Output is B2OUT2. A0OUT = 0x2 - Output is A0OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

13.21.2.59OUTCFG1 Register

Counter/Timer Output Config 1

OFFSET: 0x00000108

INSTANCE 0 ADDRESS: 0x40008108

Pad output configuration 1.

Table 937: OUTCFG1 Register

31	30	29	28	27	26	25	24	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	CFG19			CFG18			CFG17			CFG16			CFG15			RSVD	CFG14			CFG13			CFG12			CFG11			CFG10		

Table 938: OUTCFG1 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30:28	CFG19	0x2	RW	Pad output 19 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B1OUT2 = 0x5 - Output is B1OUT2. B4OUT = 0x4 - Output is B4OUT. A2OUT = 0x3 - Output is A2OUT. B4OUT2 = 0x2 - Output is B4OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
27:25	CFG18	0x2	RW	Pad output 18 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A3OUT2 = 0x5 - Output is A3OUT2. A0OUT = 0x4 - Output is A0OUT. B0OUT = 0x3 - Output is B0OUT. B4OUT = 0x2 - Output is B4OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
24:22	CFG17	0x2	RW	Pad output 17 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A1OUT2 = 0x5 - Output is A1OUT2. A4OUT = 0x4 - Output is A4OUT. B7OUT = 0x3 - Output is B7OUT. A4OUT2 = 0x2 - Output is A4OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
21:19	CFG16	0x2	RW	Pad output 16 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B3OUT2 = 0x5 - Output is B3OUT2. A0OUT2 = 0x4 - Output is A0OUT2. A0OUT = 0x3 - Output is A0OUT. A4OUT = 0x2 - Output is A4OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

Table 938: OUTCFG1 Register Bits

Bit	Name	Reset	RW	Description
18:16	CFG15	0x2	RW	Pad output 15 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A4OUT2 = 0x5 - Output is A4OUT2. A7OUT = 0x4 - Output is A7OUT. B3OUT = 0x3 - Output is B3OUT. B3OUT2 = 0x2 - Output is B3OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
15	RSVD	0x0	RO	RESERVED
14:12	CFG14	0x2	RW	Pad output 14 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A7OUT = 0x5 - Output is A7OUT. B7OUT2 = 0x4 - Output is B7OUT2. B1OUT = 0x3 - Output is B1OUT. B3OUT = 0x2 - Output is B3OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
11:9	CFG13	0x1	RW	Pad output 13 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B4OUT2 = 0x5 - Output is B4OUT2. A6OUT = 0x4 - Output is A6OUT. A3OUT = 0x3 - Output is A3OUT. A3OUT2 = 0x2 - Output is A3OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
8:6	CFG12	0x2	RW	Pad output 12 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B6OUT2 = 0x5 - Output is B6OUT2. B0OUT2 = 0x4 - Output is B0OUT2. B1OUT = 0x3 - Output is B1OUT. A3OUT = 0x2 - Output is A3OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
5:3	CFG11	0x2	RW	Pad output 11 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B5OUT2 = 0x5 - Output is B5OUT2. B4OUT = 0x4 - Output is B4OUT. B2OUT = 0x3 - Output is B2OUT. B2OUT2 = 0x2 - Output is B2OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

Table 938: OUTCFG1 Register Bits

Bit	Name	Reset	RW	Description
2:0	CFG10	0x2	RW	Pad output 10 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A6OUT = 0x5 - Output is A6OUT. B4OUT2 = 0x4 - Output is B4OUT2. B3OUT2 = 0x3 - Output is B3OUT2. B2OUT = 0x2 - Output is B2OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

13.21.2.60OUTCFG2 Register

Counter/Timer Output Config 2

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x4000810C

Pad output configuration 2.

Table 939: OUTCFG2 Register

31	30	29	28	27	26	25	24	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	CFG29			CFG28			CFG27			CFG26			CFG25			RSVD	CFG24			CFG23			CFG22			CFG21			CFG20		

Table 940: OUTCFG2 Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30:28	CFG29	0x2	RW	Pad output 29 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A3OUT2 = 0x5 - Output is A3OUT2. A7OUT = 0x4 - Output is A7OUT. A1OUT = 0x3 - Output is A1OUT. B5OUT2 = 0x2 - Output is B5OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

Table 940: OUTCFG2 Register Bits

Bit	Name	Reset	RW	Description
27:25	CFG28	0x2	RW	Pad output 28 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B0OUT2 = 0x5 - Output is B0OUT2. A5OUT2 = 0x4 - Output is A5OUT2. A3OUT = 0x3 - Output is A3OUT. A7OUT = 0x2 - Output is A7OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
24:22	CFG27	0x2	RW	Pad output 27 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B2OUT2 = 0x5 - Output is B2OUT2. B6OUT = 0x4 - Output is B6OUT. A1OUT = 0x3 - Output is A1OUT. B6OUT2 = 0x2 - Output is B6OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
21:19	CFG26	0x2	RW	Pad output 26 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A1OUT2 = 0x5 - Output is A1OUT2. A5OUT = 0x4 - Output is A5OUT. B2OUT = 0x3 - Output is B2OUT. B6OUT = 0x2 - Output is B6OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
18:16	CFG25	0x2	RW	Pad output 25 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A2OUT2 = 0x5 - Output is A2OUT2. A6OUT = 0x4 - Output is A6OUT. B2OUT = 0x3 - Output is B2OUT. B4OUT2 = 0x2 - Output is B4OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
15	RSVD	0x0	RO	RESERVED
14:12	CFG24	0x2	RW	Pad output 24 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B1OUT2 = 0x5 - Output is B1OUT2. A1OUT = 0x4 - Output is A1OUT. A2OUT = 0x3 - Output is A2OUT. A6OUT = 0x2 - Output is A6OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

Table 940: OUTCFG2 Register Bits

Bit	Name	Reset	RW	Description
11:9	CFG23	0x1	RW	Pad output 23 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B0OUT2 = 0x5 - Output is B0OUT2. A5OUT = 0x4 - Output is A5OUT. A7OUT = 0x3 - Output is A7OUT. B5OUT2 = 0x2 - Output is B5OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
8:6	CFG22	0x2	RW	Pad output 22 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A2OUT2 = 0x5 - Output is A2OUT2. A1OUT = 0x4 - Output is A1OUT. A6OUT = 0x3 - Output is A6OUT. B5OUT = 0x2 - Output is B5OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
5:3	CFG21	0x2	RW	Pad output 21 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A0OUT2 = 0x5 - Output is A0OUT2. B5OUT = 0x4 - Output is B5OUT. A1OUT = 0x3 - Output is A1OUT. A5OUT2 = 0x2 - Output is A5OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
2:0	CFG20	0x2	RW	Pad output 20 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B2OUT2 = 0x5 - Output is B2OUT2. A1OUT2 = 0x4 - Output is A1OUT2. A1OUT = 0x3 - Output is A1OUT. A5OUT = 0x2 - Output is A5OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

13.21.2.61OUTCFG3 Register

Counter/Timer Output Config 3

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x40008114

Pad output configuration 3.

Table 941: OUTCFG3 Register

31	30	29	28	27	26	25	24	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																										CFG31			CFG30		

Table 942: OUTCFG3 Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5:3	CFG31	0x2	RW	Pad output 31 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. B3OUT2 = 0x5 - Output is B3OUT2. B7OUT = 0x4 - Output is B7OUT. A6OUT = 0x3 - Output is A6OUT. B7OUT2 = 0x2 - Output is B7OUT2 ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0
2:0	CFG30	0x2	RW	Pad output 30 configuration A7OUT2 = 0x7 - Output is A7OUT2. A6OUT2 = 0x6 - Output is A6OUT2. A0OUT2 = 0x5 - Output is A0OUT2. A4OUT2 = 0x4 - Output is A4OUT2. B3OUT = 0x3 - Output is B3OUT. B7OUT = 0x2 - Output is B7OUT ONE = 0x1 - Force output to 1. ZERO = 0x0 - Force output to 0

13.21.2.62INCFG Register

Counter/Timer Input Config

OFFSET: 0x00000118

INSTANCE 0 ADDRESS: 0x40008118

Pad input configuration.

Table 943: INCFG Register

31	30	29	28	27	26	25	24	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																CFGB7	CFGA7	CFGB6	CFGA6	CFGB5	CFGA5	CFGB4	CFGA4	CFGB3	CFGA3	CFGB2	CFGA2	CFGB1	CFGA1	CFGB0	CFGA0

Table 944: INCFG Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	RESERVED
15	CFGB7	0x0	RW	CTIMER B7 input configuration CT31 = 0x1 - Input is CT31 CT30 = 0x0 - Input is CT30
14	CFGA7	0x0	RW	CTIMER A7 input configuration CT29 = 0x1 - Input is CT29 CT28 = 0x0 - Input is CT28
13	CFGB6	0x0	RW	CTIMER B6 input configuration CT27 = 0x1 - Input is CT27 CT26 = 0x0 - Input is CT26
12	CFGA6	0x0	RW	CTIMER A6 input configuration CT25 = 0x1 - Input is CT25 CT24 = 0x0 - Input is CT24
11	CFGB5	0x0	RW	CTIMER B5 input configuration CT23 = 0x1 - Input is CT23 CT22 = 0x0 - Input is CT22
10	CFGA5	0x0	RW	CTIMER A5 input configuration CT21 = 0x1 - Input is CT21 CT20 = 0x0 - Input is CT20
9	CFGB4	0x0	RW	CTIMER B4 input configuration CT19 = 0x1 - Input is CT19 CT18 = 0x0 - Input is CT18
8	CFGA4	0x0	RW	CTIMER A4 input configuration CT17 = 0x1 - Input is CT17 CT16 = 0x0 - Input is CT16
7	CFGB3	0x0	RW	CTIMER B3 input configuration CT15 = 0x1 - Input is CT15 CT14 = 0x0 - Input is CT14
6	CFGA3	0x0	RW	CTIMER A3 input configuration CT13 = 0x1 - Input is CT13 CT12 = 0x0 - Input is CT12
5	CFGB2	0x0	RW	CTIMER B2 input configuration CT11 = 0x1 - Input is CT11 CT10 = 0x0 - Input is CT10
4	CFGA2	0x0	RW	CTIMER A2 input configuration CT9 = 0x1 - Input is CT9 CT8 = 0x0 - Input is CT8

Table 944: INCFG Register Bits

Bit	Name	Reset	RW	Description
3	CFGB1	0x0	RW	CTIMER B1 input configuration CT7 = 0x1 - Input is CT7 CT6 = 0x0 - Input is CT6
2	CFGA1	0x0	RW	CTIMER A1 input configuration CT5 = 0x1 - Input is CT5 CT4 = 0x0 - Input is CT4
1	CFGB0	0x0	RW	CTIMER B0 input configuration CT3 = 0x1 - Input is CT3 CT2 = 0x0 - Input is CT2
0	CFGA0	0x0	RW	CTIMER A0 input configuration CT1 = 0x1 - Input is CT1 CT0 = 0x0 - Input is CT0

13.21.2.63INTEN 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 945: 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
CTMRB7C1INT	CTMRA7C1INT	CTMRB6C1INT	CTMRA6C1INT	CTMRB5C1INT	CTMRA5C1INT	CTMRB4C1INT	CTMRA4C1INT	CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB7C0INT	CTMRA7C0INT	CTMRB6C0INT	CTMRA6C0INT	CTMRB5C0INT	CTMRA5C0INT	CTMRB4C0INT	CTMRA4C0INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 946: INTEN Register Bits

Bit	Name	Reset	RW	Description
31	CTMRB7C1INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR1.
30	CTMRA7C1INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR1.
29	CTMRB6C1INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR1.
28	CTMRA6C1INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR1.

Table 946: INTEN Register Bits

Bit	Name	Reset	RW	Description
27	CTMRB5C1INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR1.
26	CTMRA5C1INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR1.
25	CTMRB4C1INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR1.
24	CTMRA4C1INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR1.
23	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
22	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
21	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
20	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
19	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
18	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
17	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
16	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
15	CTMRB7C0INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR0.
14	CTMRA7C0INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR0.
13	CTMRB6C0INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR0.
12	CTMRA6C0INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR0.
11	CTMRB5C0INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR0.
10	CTMRA5C0INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR0.
9	CTMRB4C0INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR0.
8	CTMRA4C0INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR0.
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.

Table 946: INTEN Register Bits

Bit	Name	Reset	RW	Description
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.

13.21.2.64INTSTAT 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 947: 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
CTMRB7C1INT	CTMRA7C1INT	CTMRB6C1INT	CTMRA6C1INT	CTMRB5C1INT	CTMRA5C1INT	CTMRB4C1INT	CTMRA4C1INT	CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB7C0INT	CTMRA7C0INT	CTMRB6C0INT	CTMRA6C0INT	CTMRB5C0INT	CTMRA5C0INT	CTMRB4C0INT	CTMRA4C0INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 948: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31	CTMRB7C1INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR1.
30	CTMRA7C1INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR1.
29	CTMRB6C1INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR1.
28	CTMRA6C1INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR1.
27	CTMRB5C1INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR1.
26	CTMRA5C1INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR1.

Table 948: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
25	CTMRB4C1INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR1.
24	CTMRA4C1INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR1.
23	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
22	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
21	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
20	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
19	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
18	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
17	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
16	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
15	CTMRB7C0INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR0.
14	CTMRA7C0INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR0.
13	CTMRB6C0INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR0.
12	CTMRA6C0INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR0.
11	CTMRB5C0INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR0.
10	CTMRA5C0INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR0.
9	CTMRB4C0INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR0.
8	CTMRA4C0INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR0.
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.

Table 948: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
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.

13.21.2.65INTCLR 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 949: 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
CTMRB7C1INT	CTMRA7C1INT	CTMRB6C1INT	CTMRA6C1INT	CTMRB5C1INT	CTMRA5C1INT	CTMRB4C1INT	CTMRA4C1INT	CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB7C0INT	CTMRA7C0INT	CTMRB6C0INT	CTMRA6C0INT	CTMRB5C0INT	CTMRA5C0INT	CTMRB4C0INT	CTMRA4C0INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 950: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31	CTMRB7C1INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR1.
30	CTMRA7C1INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR1.
29	CTMRB6C1INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR1.
28	CTMRA6C1INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR1.
27	CTMRB5C1INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR1.
26	CTMRA5C1INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR1.
25	CTMRB4C1INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR1.
24	CTMRA4C1INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR1.

Table 950: INTCLR Register Bits

Bit	Name	Reset	RW	Description
23	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
22	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.
21	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
20	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
19	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
18	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
17	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
16	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
15	CTMRB7C0INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR0.
14	CTMRA7C0INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR0.
13	CTMRB6C0INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR0.
12	CTMRA6C0INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR0.
11	CTMRB5C0INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR0.
10	CTMRA5C0INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR0.
9	CTMRB4C0INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR0.
8	CTMRA4C0INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR0.
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.

Table 950: INTCLR Register Bits

Bit	Name	Reset	RW	Description
0	CTMRA0C0INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR0.

13.21.2.66INTSET 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 951: 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
CTMRB7C1INT	CTMRA7C1INT	CTMRB6C1INT	CTMRA6C1INT	CTMRB5C1INT	CTMRA5C1INT	CTMRB4C1INT	CTMRA4C1INT	CTMRB3C1INT	CTMRA3C1INT	CTMRB2C1INT	CTMRA2C1INT	CTMRB1C1INT	CTMRA1C1INT	CTMRB0C1INT	CTMRA0C1INT	CTMRB7C0INT	CTMRA7C0INT	CTMRB6C0INT	CTMRA6C0INT	CTMRB5C0INT	CTMRA5C0INT	CTMRB4C0INT	CTMRA4C0INT	CTMRB3C0INT	CTMRA3C0INT	CTMRB2C0INT	CTMRA2C0INT	CTMRB1C0INT	CTMRA1C0INT	CTMRB0C0INT	CTMRA0C0INT

Table 952: INTSET Register Bits

Bit	Name	Reset	RW	Description
31	CTMRB7C1INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR1.
30	CTMRA7C1INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR1.
29	CTMRB6C1INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR1.
28	CTMRA6C1INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR1.
27	CTMRB5C1INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR1.
26	CTMRA5C1INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR1.
25	CTMRB4C1INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR1.
24	CTMRA4C1INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR1.
23	CTMRB3C1INT	0x0	RW	Counter/Timer B3 interrupt based on COMPR1.
22	CTMRA3C1INT	0x0	RW	Counter/Timer A3 interrupt based on COMPR1.

Table 952: INTSET Register Bits

Bit	Name	Reset	RW	Description
21	CTMRB2C1INT	0x0	RW	Counter/Timer B2 interrupt based on COMPR1.
20	CTMRA2C1INT	0x0	RW	Counter/Timer A2 interrupt based on COMPR1.
19	CTMRB1C1INT	0x0	RW	Counter/Timer B1 interrupt based on COMPR1.
18	CTMRA1C1INT	0x0	RW	Counter/Timer A1 interrupt based on COMPR1.
17	CTMRB0C1INT	0x0	RW	Counter/Timer B0 interrupt based on COMPR1.
16	CTMRA0C1INT	0x0	RW	Counter/Timer A0 interrupt based on COMPR1.
15	CTMRB7C0INT	0x0	RW	Counter/Timer B7 interrupt based on COMPR0.
14	CTMRA7C0INT	0x0	RW	Counter/Timer A7 interrupt based on COMPR0.
13	CTMRB6C0INT	0x0	RW	Counter/Timer B6 interrupt based on COMPR0.
12	CTMRA6C0INT	0x0	RW	Counter/Timer A6 interrupt based on COMPR0.
11	CTMRB5C0INT	0x0	RW	Counter/Timer B5 interrupt based on COMPR0.
10	CTMRA5C0INT	0x0	RW	Counter/Timer A5 interrupt based on COMPR0.
9	CTMRB4C0INT	0x0	RW	Counter/Timer B4 interrupt based on COMPR0.
8	CTMRA4C0INT	0x0	RW	Counter/Timer A4 interrupt based on COMPR0.
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.

14. System Timer Module

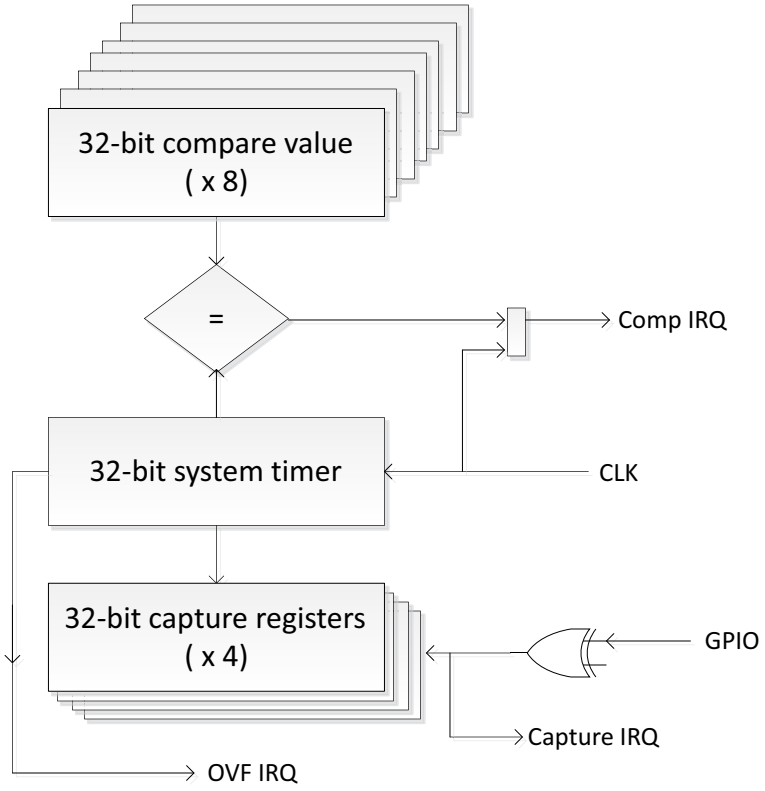


Figure 84. Block Diagram for the System Timer

14.1 Functional Overview

The Apollo3 Blue MCU System Timer (STIMER), shown above in Figure 84, 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 Apollo3 Blue MCU. This counter is reset at the actual power cycle reset of the MCU. 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.

14.2 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 timestamping events when performing capture or compare functions.

14.2.1 Register Memory Map

Table 953: STIMER Register Map

Address(s)	Register Name	Description
0x40008140	STCFG	Configuration Register
0x40008144	STTMR	System Timer Count Register (Real Time Counter)
0x40008148	CAPTURECONTROL	Capture Control Register
0x40008150	SCMPR0	Compare Register A
0x40008154	SCMPR1	Compare Register B
0x40008158	SCMPR2	Compare Register C
0x4000815C	SCMPR3	Compare Register D
0x40008160	SCMPR4	Compare Register E
0x40008164	SCMPR5	Compare Register F
0x40008168	SCMPR6	Compare Register G
0x4000816C	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
0x400081FC	SNVR3	System Timer NVRAM_D 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

14.2.2 STIMER Registers

14.2.2.1 STCFG Register

Configuration Register

OFFSET: 0x00000140

INSTANCE 0 ADDRESS: 0x40008140

The STIMER Configuration Register contains the software control for selecting the clock divider and source feeding the system timer.

Table 954: 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 955: 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 MCU. 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	COM-PARE_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 comparision is met. DISABLE = 0x0 - Compare H disabled. ENABLE = 0x1 - Compare H enabled.
14	COM-PARE_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 comparision is met. DISABLE = 0x0 - Compare G disabled. ENABLE = 0x1 - Compare G enabled.

Table 955: STCFG Register Bits

Bit	Name	Reset	RW	Description
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 comparision is met. DISABLE = 0x0 - Compare F disabled. ENABLE = 0x1 - Compare F enabled.
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 comparision 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 comparision 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 comparision 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 comparision 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 comparision 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.

14.2.2.2 STTMR Register

System Timer Count Register (Real Time Counter)

OFFSET: 0x00000144

INSTANCE 0 ADDRESS: 0x40008144

The COUNTER Register contains the running count of time as maintained by incrementing for every rising clock edge of the clock source selected in the configuration register. It is this counter value that 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. Use the FREEZE bit will stop this counter from incrementing.

Table 956: 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
STTMR																															

Table 957: STTMR Register Bits

Bit	Name	Reset	RW	Description
31:0	STTMR	0x0	RO	Value of the 32-bit counter as it ticks over.

14.2.2.3 CAPTURECONTROL Register

Capture Control Register

OFFSET: 0x00000148

INSTANCE 0 ADDRESS: 0x40008148

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: 8-bit writes can control individual capture registers atomically.

Table 958: CAPTURECONTROL Register

31	30	29	28	27	26	25	24	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																												CAPTURE3	CAPTURE2	CAPTURE1	CAPTURE0

Table 959: CAPTURECONTROL Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED.
3	CAPTURE3	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.
2	CAPTURE2	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.
1	CAPTURE1	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.
0	CAPTURE0	0x0	RW	Selects whether capture is enabled for the specified capture register. DISABLE = 0x0 - Capture function disabled. ENABLE = 0x1 - Capture function enabled.

14.2.2.4 SCMPR0 Register

Compare Register A

OFFSET: 0x00000150

INSTANCE 0 ADDRESS: 0x40008150

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 960: 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
SCMPR0																															

Table 961: SCMPR0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR0	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.

14.2.2.5 SCMPR1 Register

Compare Register B

OFFSET: 0x00000154

INSTANCE 0 ADDRESS: 0x40008154

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 962: 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
SCMPR1																															

Table 963: SCMPR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR1	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.

14.2.2.6 SCMPR2 Register

Compare Register C

OFFSET: 0x00000158

INSTANCE 0 ADDRESS: 0x40008158

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 964: 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
SCMPR2																															

Table 965: SCMPR2 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR2	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.

14.2.2.7 SCMPR3 Register

Compare Register D

OFFSET: 0x0000015C

INSTANCE 0 ADDRESS: 0x4000815C

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 966: 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
SCMPR3																															

Table 967: SCMPR3 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR3	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.

14.2.2.8 SCMPR4 Register

Compare Register E

OFFSET: 0x00000160

INSTANCE 0 ADDRESS: 0x40008160

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 968: 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
SCMPR4																															

Table 969: SCMPR4 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR4	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.

14.2.2.9 SCMPR5 Register

Compare Register F

OFFSET: 0x00000164

INSTANCE 0 ADDRESS: 0x40008164

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 970: 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
SCMPR5																															

Table 971: SCMPR5 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR5	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.

14.2.2.10SCMPR6 Register

Compare Register G

OFFSET: 0x00000168

INSTANCE 0 ADDRESS: 0x40008168

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 972: 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
SCMPR6																															

Table 973: SCMPR6 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR6	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.

14.2.2.11SCMPR7 Register

Compare Register H

OFFSET: 0x0000016C

INSTANCE 0 ADDRESS: 0x4000816C

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 974: 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
SCMPR7																															

Table 975: SCMPR7 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCMPR7	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.

14.2.2.12SCAPT0 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 976: 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
SCAPT0																															

Table 977: SCAPT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCAPT0	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.

14.2.2.13SCAPT1 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 978: 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
SCAPT1																															

Table 979: SCAPT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCAPT1	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.

14.2.2.14SCAPT2 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 980: 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
SCAPT2																															

Table 981: SCAPT2 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCAPT2	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.

14.2.2.15SCAPT3 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 982: 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
SCAPT3																															

Table 983: SCAPT3 Register Bits

Bit	Name	Reset	RW	Description
31:0	SCAPT3	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.

14.2.2.16SNVR0 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 984: 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
SNVR0																															

Table 985: SNVR0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SNVR0	0x0	RW	Value of the 32-bit counter as it ticks over.

14.2.2.17SNVR1 Register

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 986: 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
SNVR1																															

Table 987: SNVR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SNVR1	0x0	RW	Value of the 32-bit counter as it ticks over.

14.2.2.18SNVR2 Register

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 988: 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
SNVR2																															

Table 989: SNVR2 Register Bits

Bit	Name	Reset	RW	Description
31:0	SNVR2	0x0	RW	Value of the 32-bit counter as it ticks over.

14.2.2.19SNVR3 Register

System Timer NVRAM_D Register

OFFSET: 0x000001FC

INSTANCE 0 ADDRESS: 0x400081FC

The NVRAM_D 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 990: SNVR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SNVR3																															

Table 991: SNVR3 Register Bits

Bit	Name	Reset	RW	Description
31:0	SNVR3	0x0	RW	Value of the 32-bit counter as it ticks over.

14.2.2.20STMINTEN Register

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 992: 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 993: 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.
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.

Table 993: STMINTEN Register Bits

Bit	Name	Reset	RW	Description
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.

14.2.2.21STMINTSTAT Register

STIMER Interrupt registers: Status

OFFSET: 0x00000304

INSTANCE 0 ADDRESS: 0x40008304

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

Table 994: 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 995: 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.

14.2.2.22STMINTCLR Register

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 996: 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 997: 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.

Table 997: STMINTCLR Register Bits

Bit	Name	Reset	RW	Description
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.

14.2.2.23STMINTSET Register

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 998: 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 999: 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.

Table 999: STMINTSET Register Bits

Bit	Name	Reset	RW	Description
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.

15. Watchdog Timer Module

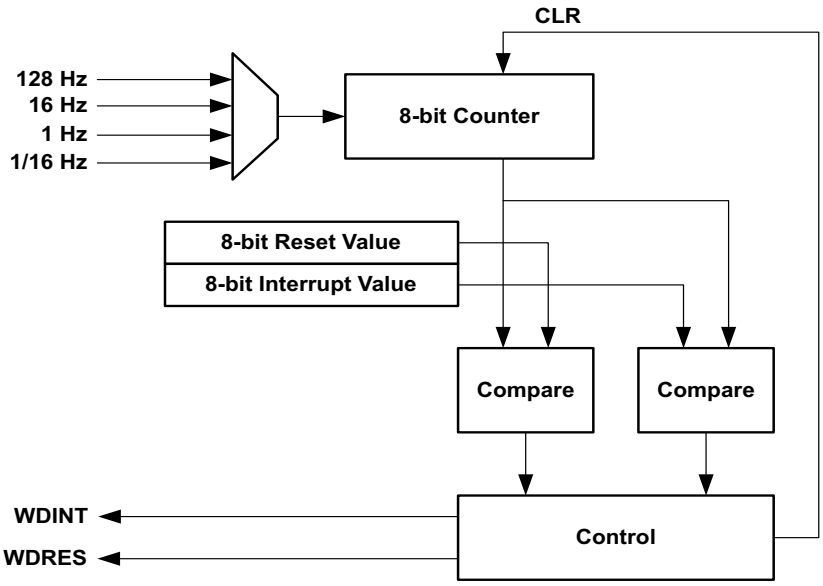


Figure 85. Block diagram for the Watchdog Timer Module

15.1 Functional Overview

The Watchdog Timer (WDT), shown in Figure 85, is used to insure that software is operational, by resetting the Apollo3 Blue MCU if the WDT reaches a configurable value before being cleared by software. The WDT can beclocked 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.

15.2 WDT Registers

Watchdog Timer

INSTANCE 0 BASE ADDRESS:0x40024000

15.2.1 Register Memory Map

Table 1000: WDT Register Map

Address(s)	Register 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

15.2.2 WDT Registers

15.2.2.1 CFG Register

Configuration Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40024000

This is the configuration register for the watch dog timer. It controls the enable, interrupt set, clocks for the timer, the compare values for the counters to trigger a reset or interrupt. This register can only be written to if the watch dog timer is unlocked (WDTLOCK is not set).

Table 1001: 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 1002: CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	This bitfield 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. This setting disables the watch dog timer. 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 bitfield is the compare value for counter bits 7:0 to generate a watch-dog interrupt.
15:8	RESVAL	0xff	RW	This bitfield is the compare value for counter bits 7:0 to generate a watch-dog reset. This will cause a software reset.
7:3	RSVD	0x0	RO	This bitfield is reserved for future use.
2	RESEN	0x0	RW	This bitfield enables the WDT reset. This needs to be set together with the WDREN bit in REG_RSTGEN_CFG register (in reset gen) to trigger the reset.
1	INTEN	0x0	RW	This bitfield 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.

Table 1002: CFG Register Bits

Bit	Name	Reset	RW	Description
0	WDTEN	0x0	RW	This bitfield enables the WDT.

15.2.2.2 RSTRT Register

Restart the watchdog timer.

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40024004

This register will Restart the watchdog timer. Writing a special key value into this register will result in the watch dog timer being reset, so that the count will start again. It is expected that the software will periodically write to this register to indicate that the system is functional. The watch dog timer can continue running when the system is in deep sleep, and the interrupt will trigger the wake. After the wake, the core can reset the watch dog timer.

Table 1003: 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 1004: RSTRT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bitfield is reserved for future use.
7:0	RSTRT	0x0	WO	Writing 0xB2 to WDTRSTRT restarts the watchdog timer. This is a write only register. Reading this register will only provide all 0. KEYVALUE = 0xB2 - This is the key value to write to WDTRSTRT to restart the WDT. This is a write only register.

15.2.2.3 LOCK Register

Locks the WDT

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40024008

This register locks the watch dog timer. Once it is locked, the configuration register (WDTCFG) for watch dog timer cannot be written to.

Table 1005: 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 1006: LOCK Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bitfield 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.

15.2.2.4 COUNT Register

Current Counter Value for WDT

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4002400C

This register holds the current count for the watch dog timer. This is a read only register. SW cannot set the value in the counter, but can reset it.

Table 1007: 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 1008: COUNT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bitfield is reserved for future use.
7:0	COUNT	0x0	RO	Read-Only current value of the WDT counter

15.2.2.5 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 1009: 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																															WDTIN

Table 1010: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.
0	WDTINT	0x0	RW	Watchdog Timer Interrupt.

15.2.2.6 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 1011: 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																															WDTIN

Table 1012: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.
0	WDTINT	0x0	RW	Watchdog Timer Interrupt.

15.2.2.7 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 1013: 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																															WDTIN

Table 1014: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.
0	WDTINT	0x0	RW	Watchdog Timer Interrupt.

15.2.2.8 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 1015: 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																															WDTIN

Table 1016: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.
0	WDTINT	0x0	RW	Watchdog Timer Interrupt.

16. Reset Generator Module

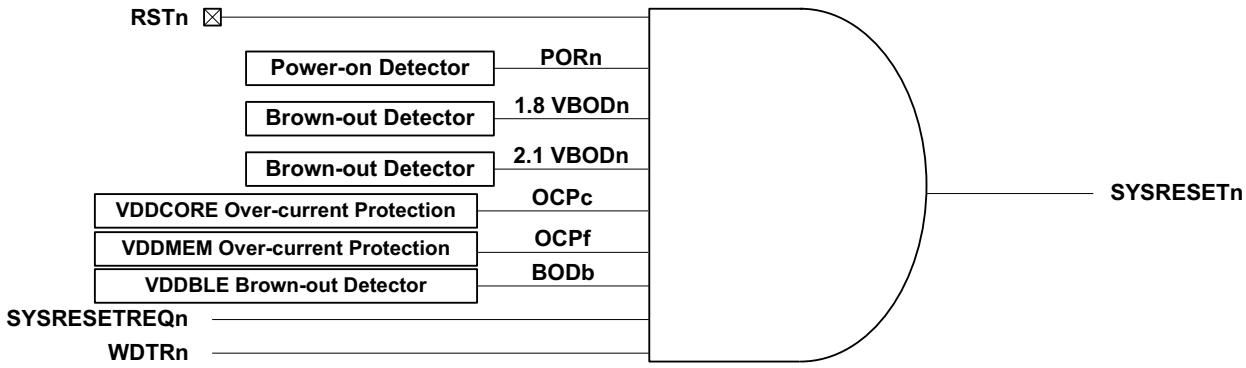


Figure 86. Block diagram for the Reset Generator Module

16.1 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 events
- Software request (SYSRESETREQn)
- Watchdog expiration

16.2 External Reset Pin

The active-low RSTn pin can be used to generate a reset using an off-chip component (e.g., a push-button). An internal pull-up resistor in the RSTn pad enables optional floating of the RSTn pin, and a de-bounce circuit ensures that bounceglitches on RSTn doesnot 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 87

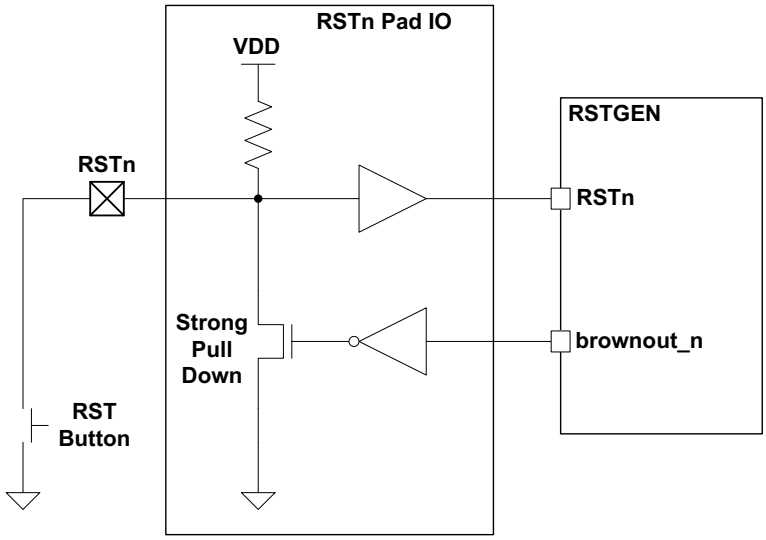


Figure 87. Block diagram of circuitry for Reset pin

16.3 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.

16.4 Brown-out Events

There are multiple brownout detectors in Apollo3. An integrated brown-out detector monitors the primary supply voltage and causes an automatic and non-configurable 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 addition, there are individual brownout detector monitors integrated within the core/memory and BLE supply regulators which cause separate/maskable reset assertions when the voltage falls below critical level for the respective voltage rails. In the event the primary supply voltage falls below the 1.755 V threshold, or 2.1 V threshold or any of the core/memory/BLE thresholds 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 32kHz 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

16.5 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.

16.6 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.

16.7 Watchdog Expiration

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.

16.8 RSTGEN Registers

MCU Reset Generator

INSTANCE 0 BASE ADDRESS:0x40000000

16.8.1 Register Memory Map

Table 1017: RSTGEN Register Map

Address(s)	Register Name	Description
0x40000000	CFG	Configuration Register
0x40000004	SWPOI	Software POI Reset
0x40000008	SWPOR	Software POR Reset
0x40000014	TPIURST	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
0x4FFFF000	STAT	Status Register (SBL)

16.8.2 RSTGEN Registers

16.8.2.1 CFG Register

Configuration Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40000000

Reset configuration register. This controls the reset enables for brownout condition, and for the expiration of the watch dog timer.

Table 1018: 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 1019: 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. This includes enabling the RESEN bit in WDTCFG register in Watch dog timer block.
0	BODHREN	0x0	RW	Brown out high (2.1v) reset enable.

16.8.2.2 SWPOI Register

Software POI Reset

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40000004

This is the software POI reset. writing the key value to this register will trigger a POI to the system. This will cause a reset to all blocks except for registers in clock gen, RTC and the stimer.

Table 1020: 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 1021: 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. This is a write-only register. Reading from this register will yield only all 0s. KEYVALUE = 0x1B - Writing 0x1B key value generates a software POI reset.

16.8.2.3 SWPOR Register

Software POR Reset

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40000008

This is the software POR reset. Writing the key value to this register will trigger a POR to the system. This will cause a reset to all blocks except for registers in clock gen, RTC, power management unit, the stimer, and the power management unit.

Table 1022: 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 1023: 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.

16.8.2.4 TPIURST Register

TPIU reset

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40000014

This will trigger a reset for the TPIU unit.

Table 1024: TPIURST Register

31	30	29	28	27	26	25	24	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 1025: TPIURST 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.

16.8.2.5 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 1026: 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 1027: 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.

16.8.2.6 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 1028: 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 1029: 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.

16.8.2.7 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 1030: 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 1031: 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.

16.8.2.8 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 1032: 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 1033: 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.

16.8.2.9 STAT Register

Status Register (SBL)

OFFSET: 0x0FFFF000

INSTANCE 0 ADDRESS: 0x4FFFF000

This register contains the status for brownout events and the causes for resets.in NOTE 1: All bits in this register, including reserved bits, are writable. Therefore care should be taken not to write this register.\n

NOTE 1: This register does not retain its value across a core deepsleep cycle. Therefore applications needing to use this value after deep sleep must copy and save this register to SRAM before initiating the first deep sleep cycle.

Table 1034: 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
SBOOT	FBOOT	RSVD																			BOBSTAT	BOFSTAT	BOCSTAT	BOUSTAT	WDRSTAT	DBGRSTAT	POIRSTAT	SWRSTAT	BORSTAT	PORSTAT	EXRSTAT

Table 1035: STAT Register Bits

Bit	Name	Reset	RW	Description
31	SBOOT	0x0	RW	Set when booting securely (SBL).
30	FBOOT	0x0	RW	Set if current boot was initiated by soft reset and resulted in Fast Boot (SBL).
29:11	RSVD	0x0	RW	RESERVED.
10	BOBSTAT	0x0	RW	A BLE/Burst Regulator Brownout Event occurred (SBL).
9	BOFSTAT	0x0	RW	A Memory Regulator Brownout Event occurred (SBL).
8	BOCSTAT	0x0	RW	A Core Regulator Brownout Event occurred (SBL).
7	BOUSTAT	0x0	RW	An Unregulated Supply Brownout Event occurred (SBL).
6	WDRSTAT	0x0	RW	Reset was initiated by a Watchdog Timer Reset (SBL).
5	DBGRSTAT	0x0	RW	Reset was a initiated by Debugger Reset (SBL).
4	POIRSTAT	0x0	RW	Reset was a initiated by Software POI Reset (SBL).
3	SWRSTAT	0x0	RW	Reset was a initiated by SW POR or AIRCR Reset (SBL).
2	BORSTAT	0x0	RW	Reset was initiated by a Brown-Out Reset (SBL).
1	PORSTAT	0x0	RW	Reset was initiated by a Power-On Reset (SBL).
0	EXRSTAT	0x0	RW	Reset was initiated by an External Reset (SBL).

17. UART Module

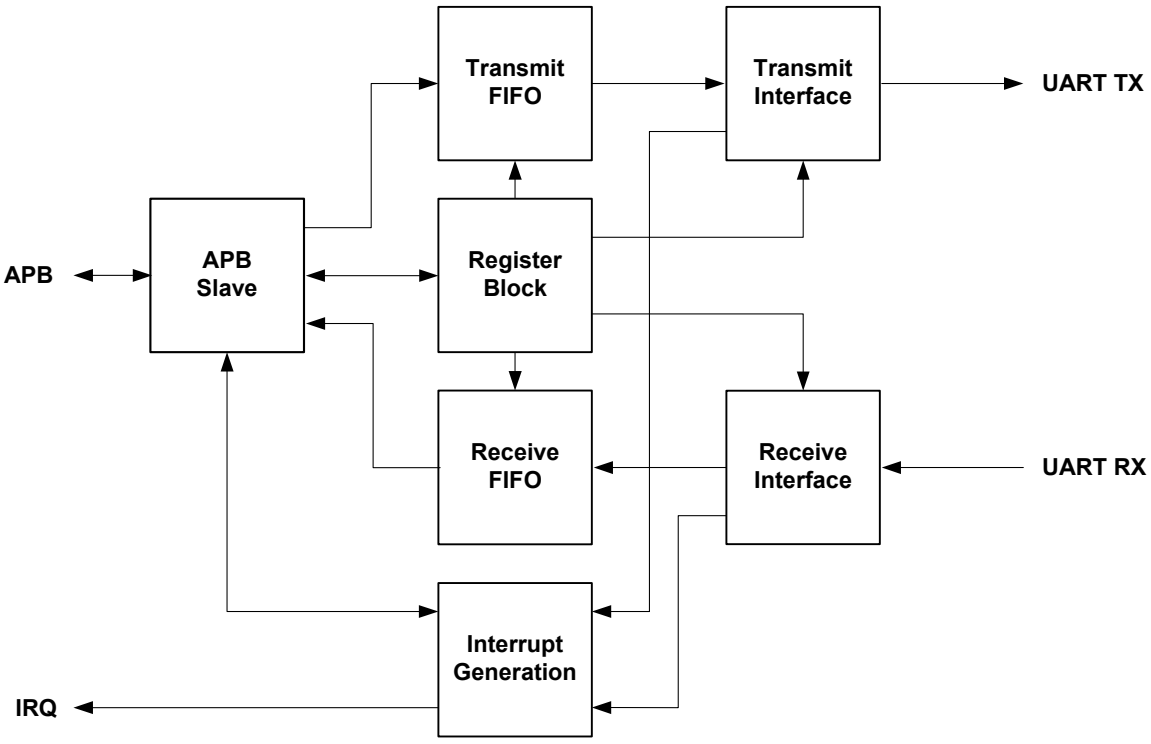


Figure 88. Block Diagram for the UART Module

17.1 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 MCU 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

17.2 Functional Overview

Shown in Figure 88, 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 MCU 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 the HFRC.

The major functional blocks of the UART are discussed briefly in the subsequent sections.

17.3 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU 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 Apollo3 Blue MCU 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.

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 is awake and set to match UART_clk when 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.

17.4 Configuration

The UART Register Block in Figure 88 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.

The UART Module supports independent CTS and RTS hardware flow control. All flow control configuration may be set using the UART_CR register.

17.5 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.

17.6 UART Registers

Serial UART

INSTANCE 0 BASE ADDRESS:0x4001C000

INSTANCE 1 BASE ADDRESS:0x4001D000

17.6.1 Register Memory Map

Table 1036: UART Register Map

Address(s)	Register 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
0x4001C040 0x4001D040	MIS	Masked Interrupt Status
0x4001C044 0x4001D044	IEC	Interrupt Clear

17.6.2 UART Registers

17.6.2.1 DR Register

UART Data Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x4001C000

INSTANCE 1 ADDRESS: 0x4001D000

UART Data Register

Table 1037: 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 1038: DR Register Bits

Bit	Name	Reset	RW	Description
31:12	RSVD	0x0	RO	This bitfield 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.

17.6.2.2 RSR Register

UART Status Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x4001C004

INSTANCE 1 ADDRESS: 0x4001D004

UART Status Register

Table 1039: 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 1040: RSR Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	This bitfield 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.

17.6.2.3 FR Register

Flag Register

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x4001C018

INSTANCE 1 ADDRESS: 0x4001D018

Flag Register

Table 1041: 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 1042: FR Register Bits

Bit	Name	Reset	RW	Description
31:9	RSVD	0x0	RO	This bitfield 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.
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.

17.6.2.4 ILPR Register

IrDA Counter

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x4001C020

INSTANCE 1 ADDRESS: 0x4001D020

IrDA Counter

Table 1043: 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 1044: ILPR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bitfield is reserved for future use.
7:0	ILPDVSR	0x0	RW	These bits hold the IrDA counter divisor.

17.6.2.5 IBRD Register

Integer Baud Rate Divisor

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x4001C024

INSTANCE 1 ADDRESS: 0x4001D024

Integer Baud Rate Divisor

Table 1045: 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 1046: IBRD Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	This bitfield is reserved for future use.
15:0	DIVINT	0x0	RW	These bits hold the baud integer divisor.

17.6.2.6 FBRD Register

Fractional Baud Rate Divisor

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x4001C028

INSTANCE 1 ADDRESS: 0x4001D028

Fractional Baud Rate Divisor

Table 1047: 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 1048: FBRD Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	This bitfield is reserved for future use.
5:0	DIVFRAC	0x0	RW	These bits hold the baud fractional divisor.

17.6.2.7 LCRH Register

Line Control High

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4001C02C

INSTANCE 1 ADDRESS: 0x4001D02C

Line Control High

Table 1049: 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 1050: LCRH Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	This bitfield 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

Table 1050: LCRH Register Bits

Bit	Name	Reset	RW	Description
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.

17.6.2.8 CR Register

Control Register

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x4001C030

INSTANCE 1 ADDRESS: 0x4001D030

Control Register

Table 1051: 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 1052: CR Register Bits

Bit	Name	Reset	RW	Description
31:16	RSVD	0x0	RO	This bitfield 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.

Table 1052: CR Register Bits

Bit	Name	Reset	RW	Description
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.
6:4	CLKSEL	0x0	RW	This bitfield 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.

17.6.2.9 IFLS Register

FIFO Interrupt Level Select

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x4001C034

INSTANCE 1 ADDRESS: 0x4001D034

FIFO Interrupt Level Select

Table 1053: 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 1054: IFLS Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	This bitfield 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.

17.6.2.10IER Register

Interrupt Enable

OFFSET: 0x00000038

INSTANCE 0 ADDRESS: 0x4001C038

INSTANCE 1 ADDRESS: 0x4001D038

Interrupt Enable

Table 1055: 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 1056: IER Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield 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.

Table 1056: IER Register Bits

Bit	Name	Reset	RW	Description
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.

17.6.2.11IES Register

Interrupt Status

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0x4001C03C

INSTANCE 1 ADDRESS: 0x4001D03C

Interrupt Status

Table 1057: 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 1058: IES Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield 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.
7	FERIS	0x0	RO	This bit holds the framing error interrupt status.
6	RTRIS	0x0	RO	This bit holds the receive timeout interrupt status.

Table 1058: IES Register Bits

Bit	Name	Reset	RW	Description
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.

17.6.2.12MIS Register

Masked Interrupt Status

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x4001C040

INSTANCE 1 ADDRESS: 0x4001D040

Masked Interrupt Status

Table 1059: 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 1060: MIS Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield 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.

Table 1060: MIS Register Bits

Bit	Name	Reset	RW	Description
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.

17.6.2.13IEC Register

Interrupt Clear

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0x4001C044

INSTANCE 1 ADDRESS: 0x4001D044

Interrupt Clear

Table 1061: 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 1062: IEC Register Bits

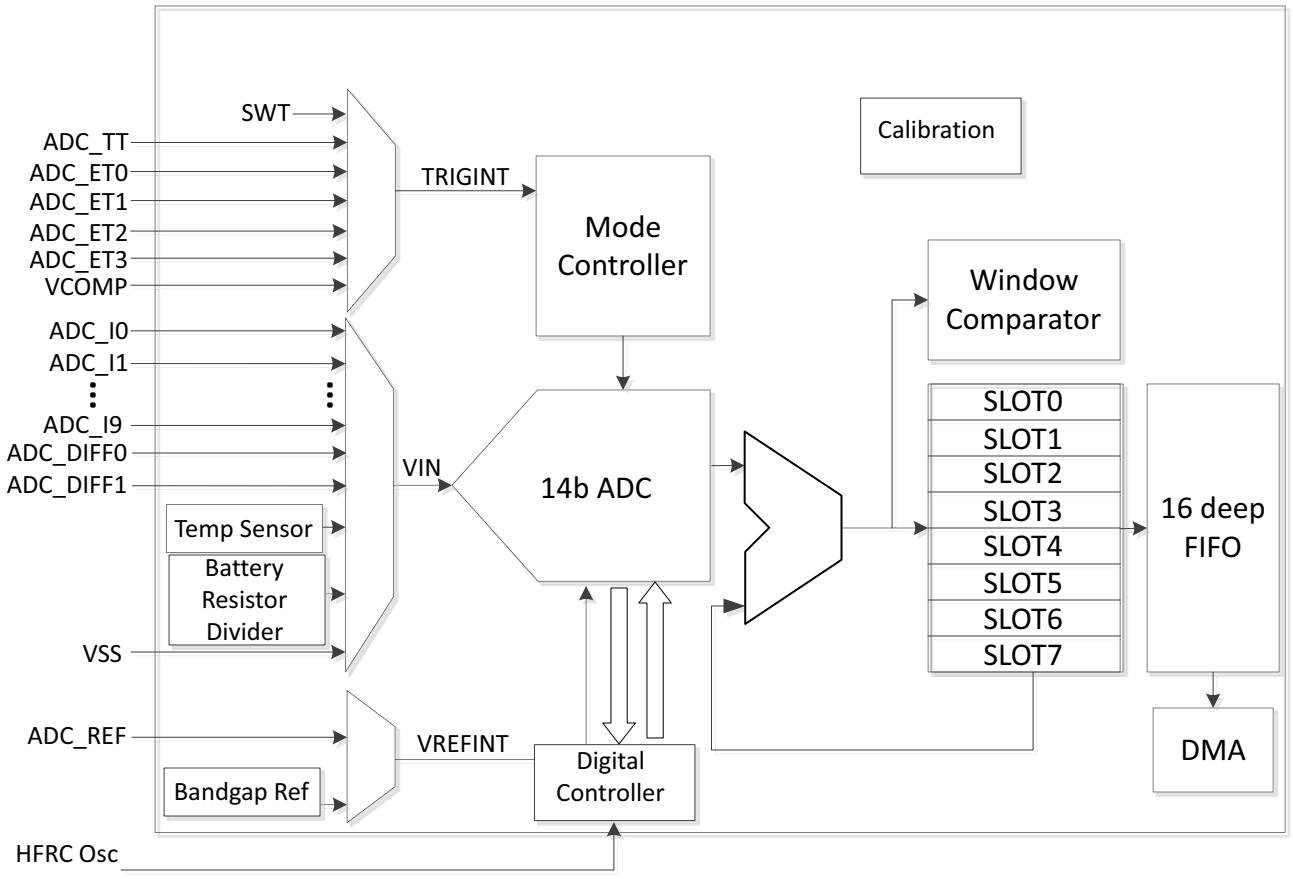
Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	This bitfield is reserved for future use.
10	OEIC	0x0	WO	This bit holds the overflow interrupt clear.
9	BEIC	0x0	WO	This bit holds the break error interrupt clear.
8	PEIC	0x0	WO	This bit holds the parity error interrupt clear.

Table 1062: IEC Register Bits

Bit	Name	Reset	RW	Description
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.

18. ADC and Temperature Sensor Module

Figure 89. Block Diagram for ADC and Temperature Sensor



18.1 Features

The Analog to Digital Converter (ADC) and Temperature Sensor Module includes a 14 bitmulti-channel Successive Approximation Register (SAR) ADC as shown in Figure 89.

Key features include:

- 15 user-selectable channels with sources including:
  - External pins
    - 10 single ended
    - 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 and DMA capability for storing measurement results and maximizing MCU 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

18.2 Functional Overview

The Apollo3 Blue MCU 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.

18.2.1 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 an HFRC 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 clock frequency modes: 24MHz and 48MHz HFRC. 48MHz mode is the default mode of operation.

18.2.2 Channel Analog Mux

As shown in Figure 89, the ADC block contains a channel analog multiplexer on the input port to the analog to digital converter. of the GPIO pins on the 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_EXT7 external GPIO pin connection.
5. ADC_EXT8 external GPIO pin connection.
6. ADC_EXT9 external GPIO pin connection.
7. ADC_EXT_DIFF0P external GPIO connection (muxed with EXT8)

   ADC_EXT_DIFF0N external GPIO connection (muxed with EXT9)
8. ADC_EXTDIFF1P external GPIO pin connection.

   ADC_EXT DIFF1N external GPIO pin connection.
9. ADC_TEMP internal temperature sensor.
10. ADC_DIV3 internal voltage divide by 3 connection to the input power rail.
11. 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.

18.2.3 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 Apollo3 Blue MCU 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) ADC_EXT3 (TRIG3) VCOMP Voltage Comparator trigger.
3. <Reserved>
4. <Reserved>
5. 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.

Finally it is important to note that a trigger event applies to all enabled slots as a whole. Individual slots can not be separately triggered.

18.2.4 Voltage Reference Sources

The Apollo3 Blue MCU ADC allows 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

18.2.5 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 "One SLOT Configuration Register" on page 726.

Table 1063: 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 “DMA” on page 730. The number of samples to accumulate will also be explained in a subsequent section. See “Automatic Sample Accumulation and Scaling” on page 726.

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 “Channel Analog Mux” on page 724.

Each of the eight conversion slots can independently specify:

- Analog Multiplexer Channel Selection
- Participation in Window Comparisons
- Automatic Sample Accumulation

18.2.6 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. 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.

IMPORTANT: 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 1064: 10.6 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 1065, “Per Slot Sample Accumulator,” on page 727. 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 1065: 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 1066 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 1066: 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													

18.2.7 Sixteen Entry Result FIFO

All results written to the FIFO have exactly the same format as shown in Table 1067. 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 1067: 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 1068: 14-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 1069: 12-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 1070: 10-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
128	0	0	0	0	10										6					

Table 1070: 10-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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
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 1071: 8-bit FIFO Data Format

# Samples	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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.

18.2.8 DMA

When enabled, the ADC can use DMA to keep its FIFO serviced and transfers samples to SRAM. Generally, DMA should be used when the desired use case is autonomous recording of samples to a pre-allocated buffer in SRAM. The buffer may be byte-aligned but must be a word-multiple in size.

The general steps to enabling ADC DMA are as follows:

1. Ensure SRAM target(s) are powered up.
2. Power up the ADC if it's not already on.
3. Configure ADC slots and ADCCFG register.
4. Set DMATOTCOUNT to the total amount of data to transfer. While the DMA is in progress, this register contains a live count of the remaining data to transfer.
5. Configure DMATARGADDR, the SRAM target byte address, for the location in memory of the first sample to be written by DMA.
6. Select a DMA trigger level by configuring DMATRIGEN to either FIFO 100% full or FIFO 75% full. This defines what conditions will initiate a DMA transfer.
7. Configure DMACFG, including setting DMAEN.
8. Trigger the ADC multiple times, using either the timer trigger (when using repeat mode), multiple SW triggers, or multiple external triggers.

Each time the FIFO fills to the appropriate level, the DMA will start and the FIFO will be drained. During this time, depending on the particular use case, it may be appropriate to put the MCU to sleep or deepsleep.

To monitor progress of the DMA, there is a DMASTAT status register. When the DMA is actively transferring data from the ADC FIFO to SRAM, DMATIP will be asserted. At the end of an entire transfer (DMATOTCOUNT reaches 0), then DMACPL will be set. Last, but not least, if an error occurs due to the DMA being asked to perform an illegal operation, DMAERR will be asserted. Causes of a DMA error include:

- DMA transfers to address outside SRAM memory region
- Popping from the FIFO while the DMA is underway

Care must be taken to avoid powering down SRAM that the DMA wants to write to.

If the DMA complete interrupt is enabled, this can be used to wake the MCU from sleep or deepsleep and communicate that the SRAM buffer has been filled and is ready for processing. The DMA error interrupt may also be used to signal the MCU that there is a problem with the DMA configuration.

To recover from a DMA error, disable any repeating trigger, disable the DMA via DMACFG's DMAEN field, and manually drain the ADC FIFO.— Then follow the procedure described above for enabling ADC DMA while correcting the configuration issue.

Some additional capabilities of the DMA include:

- ADC auto-power-off upon DMA completion: This feature, enabled via the DMACFG register's DPWROFF field, allows the ADC to power off once DMATOTCOUNT reaches zero. Note that this feature is incompatible with waking the MCU from sleep or deepsleep using the DMA complete interrupt.
- Masking FIFOCNT and SLOTNUM data from FIFO data: The DMA engine can be configured to write only samples to SRAM without the FIFOCNT and SLOTNUM data. This allows the MCU to skip the manual process of masking the potentially undesirable upper bits of each data value written to SRAM.

18.2.9 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 can't 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 1072: 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 1073: 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.

18.3 Operating Modes and the Mode Controller

The mode controller of Figure 89 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 89 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 90

An ADC conversion is the process of averaging measurements following one or more scans for each slot that is enabled.

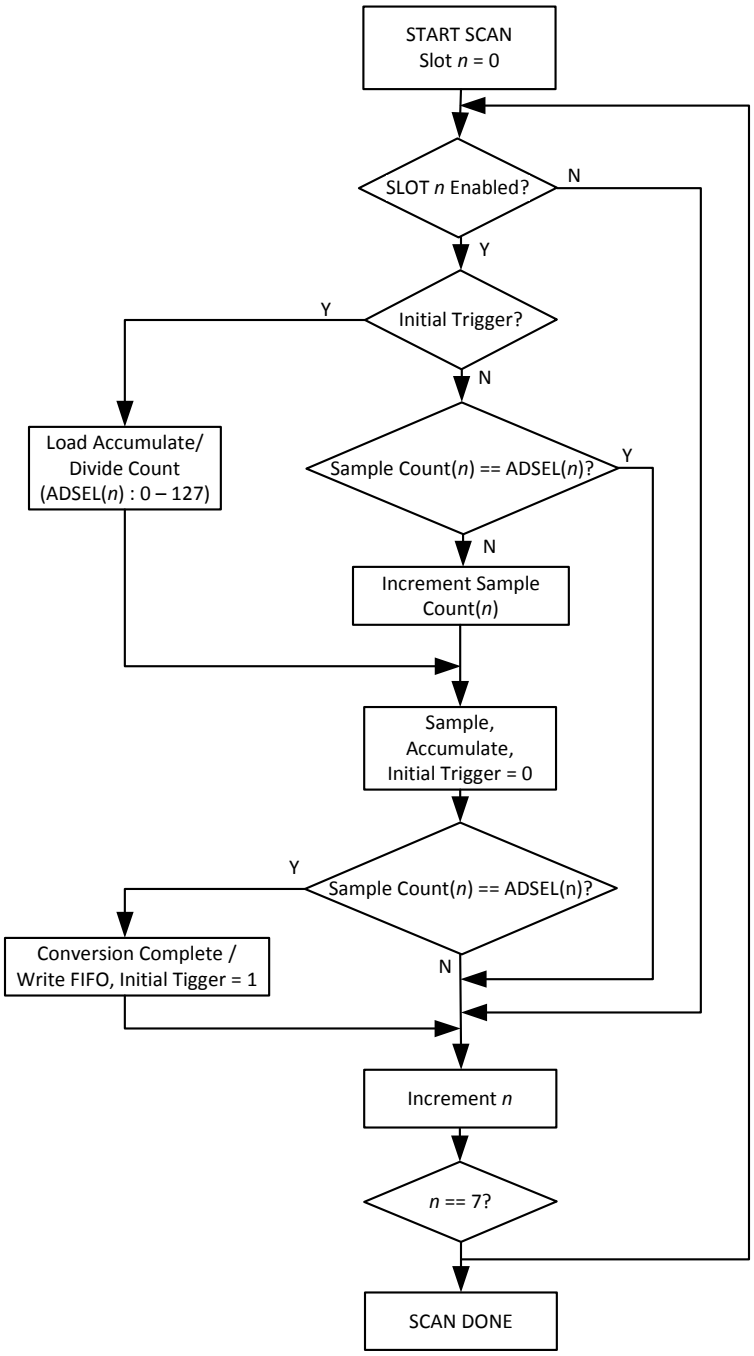


Figure 90. Scan Flowchart

18.3.1 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.

18.3.2 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.

18.3.3 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. The table below 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 1074: ADC Power Modes

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

18.3.3.1 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].

18.3.3.2 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 deepsleep 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.

18.3.3.3 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

1.

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.

18.4 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)
7. DMA transfer complete
8. DMA error condition

The window comparator interrupts are discussed above, see Section 18.2.8

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.

The DMA transfer complete interrupt is triggered upon completion of the currently configured DMA.

The DMA error interrupt is triggered if the DMA has been instructed to perform an illegal operation such as:

- writing outside SRAM
- writing to powered-down SRAM (doesn't always work - CORVETTE-628, CORVETTE-800)
- popping from the FIFO while the DMA is underway

The DMA supports none of these features.

18.5 Voltage Divider and Switchable Battery Load

The Apollo3 Blue MCU'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 MCU 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.

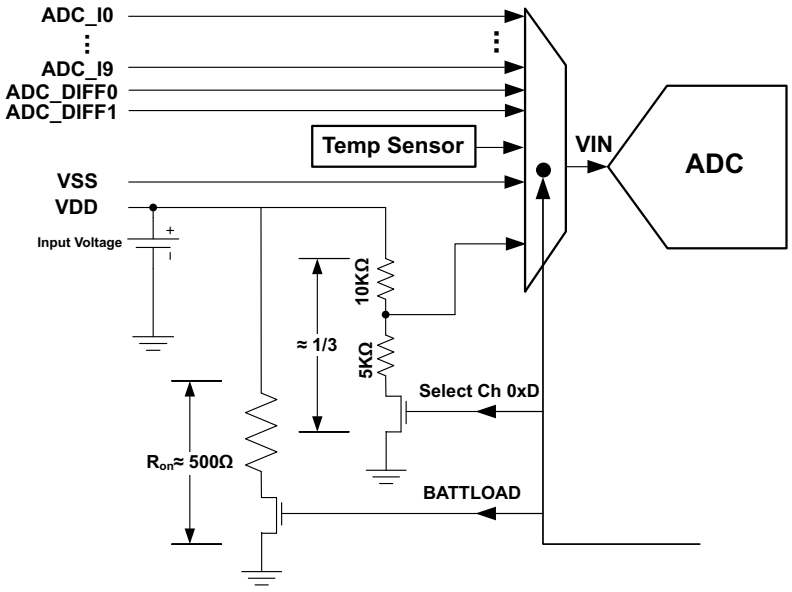


Figure 91. Switchable Battery Load

The switchable load resistor is enabled by the BATTLOAD bit as shown in the ADCBATTLOAD 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.

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.

Register Memory Map

Table 1075: ADC Register Map

Address(s)	Register 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

Table 1075: ADC Register Map

Address(s)	Register Name	Description
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
0x5001003C	FIFOPR	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
0x50010240	DMATRIGEN	DMA Trigger Enable Register
0x50010244	DMATRIGSTAT	DMA Trigger Status Register
0x50010280	DMACFG	DMA Configuration Register
0x50010284	DMABCOUNT	DMA Burst Transfer Count
0x50010288	DMATOTCOUNT	DMA Total Transfer Count
0x5001028C	DMATARGADDR	DMA Target Address Register
0x50010290	DMASTAT	DMA Status Register

ADC Registers

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 1076: 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		RSVD				TRIGPOL	TRIGSEL			RSVD			DFIFORDEN	RSVD		REFSEL		RSVD			CKMODE	LPMODE	RPTEN	RSVD	ADCEN

Table 1077: 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 divided by (CORESEL+1) HFRC_DIV2 = 0x2 - HFRC Core Clock / 2 further divided by (CORESEL+1)
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
15:13	RSVD	0x0	RO	RESERVED.
12	DFIFORDEN	0x0	RW	Destructive FIFO Read Enable. Setting this will enable FIFO pop upon reading the FIFOPR register. DIS = 0x0 - Destructive Reads are prevented. Reads to the FIFOPR register will not POP an entry off the FIFO. EN = 0x1 - Reads to the FIFOPR register will automatically pop an entry off the FIFO.
11:10	RSVD	0x0	RO	RESERVED.

Table 1077: CFG Register Bits

Bit	Name	Reset	RW	Description
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
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.

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 1078: 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																															PWDSTAT

Table 1079: 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.

SWT Register

Software trigger

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x50010008

This register enables initiating an ADC scan through software.

Table 1080: SWT Register

31	30	29	28	27	26	25	24	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 1081: 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.

SL0CFG Register

Slot 0 Configuration Register

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x5001000C

Slot 0 Configuration Register

Table 1082: SL0CFG Register

31	30	29	28	27	26	25	24	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 1083: 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 1083: SL0CFG Register Bits

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.

SL1CFG Register

Slot 1 Configuration Register

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x50010010

Slot 1 Configuration Register

Table 1084: SL1CFG Register

31	30	29	28	27	26	25	24	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 1085: SL1CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.

Table 1085: SL1CFG Register Bits

Bit	Name	Reset	RW	Description
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.

SL2CFG Register

Slot 2 Configuration Register

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x50010014

Slot 2 Configuration Register

Table 1086: SL2CFG Register

31	30	29	28	27	26	25	24	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 1087: 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 1087: SL2CFG Register Bits

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.

SL3CFG Register

Slot 3 Configuration Register

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x50010018

Slot 3 Configuration Register

Table 1088: SL3CFG Register

31	30	29	28	27	26	25	24	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 1089: SL3CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.

Table 1089: SL3CFG Register Bits

Bit	Name	Reset	RW	Description
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.

SL4CFG Register

Slot 4 Configuration Register

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x5001001C

Slot 4 Configuration Register

Table 1090: SL4CFG Register

31	30	29	28	27	26	25	24	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 1091: 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 1091: SL4CFG Register Bits

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.

SL5CFG Register

Slot 5 Configuration Register

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x50010020

Slot 5 Configuration Register

Table 1092: SL5CFG Register

31	30	29	28	27	26	25	24	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 1093: SL5CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.

Table 1093: SL5CFG Register Bits

Bit	Name	Reset	RW	Description
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.

SL6CFG Register

Slot 6 Configuration Register

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x50010024

Slot 6 Configuration Register

Table 1094: SL6CFG Register

31	30	29	28	27	26	25	24	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 1095: 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 1095: SL6CFG Register Bits

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.

SL7CFG Register

Slot 7 Configuration Register

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x50010028

Slot 7 Configuration Register

Table 1096: SL7CFG Register

31	30	29	28	27	26	25	24	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 1097: SL7CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.

Table 1097: SL7CFG Register Bits

Bit	Name	Reset	RW	Description
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.

WULIM Register

Window Comparator Upper Limits Register

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x5001002C

Window Comparator Upper Limits Register

Table 1098: WULIM Register

31	30	29	28	27	26	25	24	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 1099: 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 window comparator.

WLLIM Register

Window Comparator Lower Limits Register

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x50010030

Window Comparator Lower Limits Register

Table 1100: WLLIM Register

31	30	29	28	27	26	25	24	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 1101: 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 window comparator.

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 1102: 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
RSVD	SLOTNUM			COUNT								DATA																			

Table 1103: 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.

FIFOPR Register

FIFO Data and Valid Count Register

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0x5001003C

This is a Pop Read mirrored copy of the ADCFIFO register with the only difference being that reading this register will result in a simultaneous FIFO POP which is also achieved by writing to the ADCFIFO Register. Note: The DFIFORDEN bit must be set in the CFG register for the destructive read to be enabled.

Table 1104: FIFOPR Register

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

Table 1105: FIFOPR Register Bits

Bit	Name	Reset	RW	Description
31	RSVDPR	0x0	RO	RESERVED.
30:28	SLOTNUMPR	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.

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 1106: 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																								DERR	DCMP	WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 1107: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7	DERR	0x0	RW	DMA Error Condition DMAERROR = 0x1 - DMA Error Condition Occurred
6	DCMP	0x0	RW	DMA Transfer Complete DMACOMPLETE = 0x1 - DMA Completed a transfer
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.

Table 1107: INTEN Register Bits

Bit	Name	Reset	RW	Description
3	FIFOOVR2	0x0	RW	FIFO 100 percent full interrupt. FIFOFULLINT = 0x1 - FIFO 100 percent full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75 percent full interrupt. FIFO75INT = 0x1 - FIFO 75 percent 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.

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 1108: 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																								DERR	DCMP	WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 1109: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7	DERR	0x0	RW	DMA Error Condition DMAERROR = 0x1 - DMA Error Condition Occurred
6	DCMP	0x0	RW	DMA Transfer Complete DMACOMPLETE = 0x1 - DMA Completed a transfer
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.

Table 1109: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
3	FIFOOVR2	0x0	RW	FIFO 100 percent full interrupt. FIFOFULLINT = 0x1 - FIFO 100 percent full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75 percent full interrupt. FIFO75INT = 0x1 - FIFO 75 percent 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.

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 1110: 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																								DERR	DCMP	WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 1111: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7	DERR	0x0	RW	DMA Error Condition DMAERROR = 0x1 - DMA Error Condition Occurred
6	DCMP	0x0	RW	DMA Transfer Complete DMACOMPLETE = 0x1 - DMA Completed a transfer
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.

Table 1111: INTCLR Register Bits

Bit	Name	Reset	RW	Description
3	FIFOOVR2	0x0	RW	FIFO 100 percent full interrupt. FIFOFULLINT = 0x1 - FIFO 100 percent full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75 percent full interrupt. FIFO75INT = 0x1 - FIFO 75 percent 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.

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 1112: 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																								DERR	DCMP	WCINC	WCEXC	FIFOOVR2	FIFOOVR1	SCNCMP	CNVCMP

Table 1113: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED.
7	DERR	0x0	RW	DMA Error Condition DMAERROR = 0x1 - DMA Error Condition Occurred
6	DCMP	0x0	RW	DMA Transfer Complete DMACOMPLETE = 0x1 - DMA Completed a transfer
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.

Table 1113: INTSET Register Bits

Bit	Name	Reset	RW	Description
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOOVR2	0x0	RW	FIFO 100 percent full interrupt. FIFOFULLINT = 0x1 - FIFO 100 percent full interrupt.
2	FIFOOVR1	0x0	RW	FIFO 75 percent full interrupt. FIFO75INT = 0x1 - FIFO 75 percent 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.

DMATRIGEN Register

DMA Trigger Enable Register

OFFSET: 0x00000240

INSTANCE 0 ADDRESS: 0x50010240

DMA Trigger Enable Register

Table 1114: DMATRIGEN Register

31	30	29	28	27	26	25	24	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																														DFIFOFULL	DFIFO75

Table 1115: DMATRIGEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DFIFOFULL	0x0	RW	Trigger DMA upon FIFO 100& Full
0	DFIFO75	0x0	RW	Trigger DMA upon FIFO 75 percent Full

DMATRIGSTAT Register

DMA Trigger Status Register

OFFSET: 0x00000244

INSTANCE 0 ADDRESS: 0x50010244

DMA Trigger Status Register

Table 1116: DMATRIGSTAT Register

31	30	29	28	27	26	25	24	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																														DFULLSTAT	D75STAT

Table 1117: DMATRIGSTAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	RESERVED.
1	DFULLSTAT	0x0	RO	Triggered DMA from FIFO 100& Full
0	D75STAT	0x0	RO	Triggered DMA from FIFO 75 percent Full

DMACFG Register

DMA Configuration Register

OFFSET: 0x00000280

INSTANCE 0 ADDRESS: 0x50010280

DMA Configuration Register

Table 1118: DMACFG Register

31	30	29	28	27	26	25	24	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													DPWROFF	DMAMSK	RSVD	RSVD						DMADYNPRI	DMAPRI	RSVD						DMADIR	RSVD	DMAEN

Table 1119: DMACFG Register Bits

Bit	Name	Reset	RW	Description
31:19	RSVD	0x0	RO	RESERVED.

Table 1119: DMACFG Register Bits

Bit	Name	Reset	RW	Description
18	DPWROFF	0x0	RW	Power Off the ADC System upon DMACPL.
17	DMAMSK	0x0	RW	Mask the FIFOCNT and SLOTNUM when transferring FIFO contents to memory DIS = 0x0 - FIFO Contents are copied directly to memory without modification. EN = 0x1 - Only the FIFODATA contents are copied to memory on DMA transfers. The SLOTNUM and FIFOCNT contents are cleared to zero.
16	RSVD	0x0	RO	RESERVED
15:10	RSVD	0x0	RO	RESERVED.
9	DMADYNPRI	0x0	RW	Enables dynamic priority based on FIFO fullness. When FIFO is full, priority is automatically set to HIGH. Otherwise, DMAPRI is used. DIS = 0x0 - Disable dynamic priority (use DMAPRI setting only) EN = 0x1 - Enable dynamic priority
8	DMAPRI	0x0	RW	Sets the Priority of the DMA request LOW = 0x0 - Low Priority (service as best effort) HIGH = 0x1 - High Priority (service immediately)
7:3	RSVD	0x0	RO	RESERVED.
2	DMADIR	0x0	RO	Direction P2M = 0x0 - Peripheral to Memory (SRAM) transaction M2P = 0x1 - Memory to Peripheral transaction
1	RSVD	0x0	RO	RESERVED.
0	DMAEN	0x0	RW	DMA Enable DIS = 0x0 - Disable DMA Function EN = 0x1 - Enable DMA Function

DMABCOUNT Register

DMA Burst Transfer Count

OFFSET: 0x00000284

INSTANCE 0 ADDRESS: 0x50010284

DMA Burst Transfer Count

Table 1120: DMABCOUNT Register

31	30	29	28	27	26	25	24	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																															

Table 1121: DMABCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:0	RSVD	0x0	RO	RESERVED.

DMATOTCOUNT Register

DMA Total Transfer Count

OFFSET: 0x00000288

INSTANCE 0 ADDRESS: 0x50010288

DMA Total Transfer Count

Table 1122: DMATOTCOUNT Register

31	30	29	28	27	26	25	24	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														TOTCOUNT																BTOTCOUNT	

Table 1123: DMATOTCOUNT Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED.
17:2	TOTCOUNT	0x0	RW	Total Transfer Count
1:0	BTOTCOUNT	0x0	RO	RESERVED.

DMATARGADDR Register

DMA Target Address Register

OFFSET: 0x0000028C

INSTANCE 0 ADDRESS: 0x5001028C

DMA Target Address Register

Table 1124: DMATARGADDR Register

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

Table 1125: DMATARGADDR Register Bits

Bit	Name	Reset	RW	Description
31:19	UTARGADDR	0x400	RO	SRAM Target
18:0	LTARGADDR	0x0	RW	DMA Target Address

DMASTAT Register

DMA Status Register

OFFSET: 0x00000290

INSTANCE 0 ADDRESS: 0x50010290

DMA Status Register

Table 1126: DMASTAT Register

31	30	29	28	27	26	25	24	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																													DMAERR	DMACPL	DMATIP

Table 1127: DMASTAT Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x0	RO	RESERVED.
2	DMAERR	0x0	RW	DMA Error
1	DMACPL	0x0	RW	DMA Transfer Complete
0	DMATIP	0x0	RW	DMA Transfer In Progress

19. Voltage Comparator Module

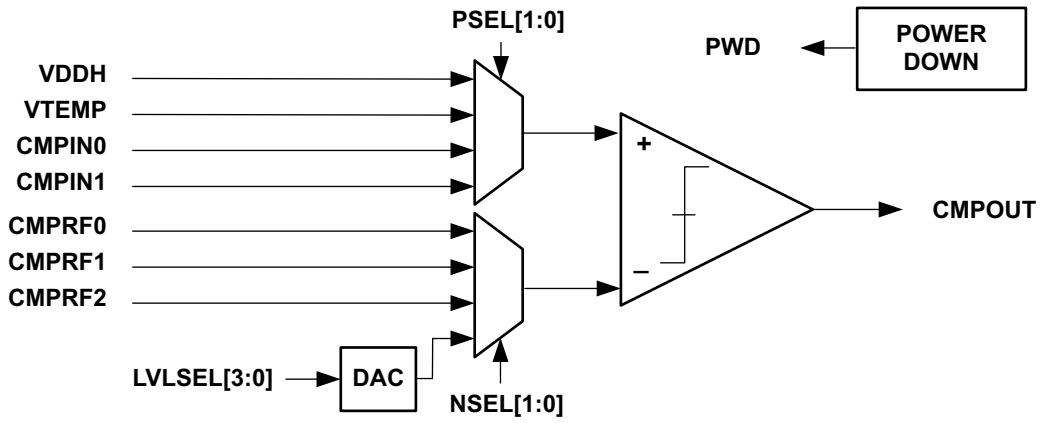


Figure 92. Block diagram for the Voltage Comparator Module

19.1 Functional Overview

The Voltage Comparator Module, shown in Figure 92, 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), 2) the PTAT voltage from the temperature sensor (VTEMP), or 3) two external voltage channels (CMPIN0 and CMPIN1).

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, 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 21.11 for hysteresis range). 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

19.2 VCOMP Registers

Voltage Comparator

INSTANCE 0 BASE ADDRESS:0x4000C000

This is the detailed description of the Voltage Comparator Register Block. The Voltage Cmparator Register Block contains the software control for selecting the comparator inputs, powerdown control, observing comparator output status and enabling interrupts.

19.2.1 Register Memory Map

Table 1128: VCOMP Register Map

Address(s)	Register Name	Description
0x4000C000	CFG	Configuration Register
0x4000C004	STAT	Status Register
0x4000C008	PWDKEY	Key Register for Powering Down the Voltage Comparator
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

19.2.2 VCOMP Registers

19.2.2.1 CFG Register

Configuration Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x4000C000

The Voltage Comparator Configuration Register contains the software control for selecting beween the 4 options for the positive input as well as the multiple options for the reference input.

Table 1129: 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												LVLSEL				RSVD						NSEL		RSVD						PSEL	

Table 1130: CFG Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	This bitfield is reserved for future use.
19:16	LVLSEL	0x0	RW	When the reference input NSEL is set to NSEL_DAC, this bitfield selects the voltage level for the negative input to the comparator. 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 bitfield is reserved for future use.
9:8	NSEL	0x0	RW	This bitfield selects the negative input to the comparator. VREFEXT1 = 0x0 - Use external reference 1 for reference input. VREFEXT2 = 0x1 - Use external reference 2 for reference input. VREFEXT3 = 0x2 - Use external reference 3 for reference input. DAC = 0x3 - Use DAC output selected by LVLSEL for reference input.
7:2	RSVD	0x0	RO	This bitfield is reserved for future use.

Table 1130: CFG Register Bits

Bit	Name	Reset	RW	Description
1:0	PSEL	0x0	RW	This bitfield selects the positive input to the comparator. VDDADJ = 0x0 - Use VDDADJ for the positive input. VTEMP = 0x1 - Use the temperature sensor output for the positive input. Note: If this channel is selected for PSEL, the bandap circuit required for temperature comparisons will automatically turn on. The bandgap circuit requires 11us to stabalize. VEXT1 = 0x2 - Use external voltage 0 for positive input. VEXT2 = 0x3 - Use external voltage 1 for positive input.

19.2.2.2 STAT Register

Status Register

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x4000C004

Status Register

Table 1131: 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																														PWDSTAT	CMPOUT

Table 1132: STAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield 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.

19.2.2.3 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 1133: PWDKEY Register

31	30	29	28	27	26	25	24	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 1134: PWDKEY Register Bits

Bit	Name	Reset	RW	Description
31:0	PWDKEY	0x0	RW	Key register value. Key = 0x37 - Key

19.2.2.4 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 1135: 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																														OUTHI	OUTLOW

Table 1136: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield 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.

19.2.2.5 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 1137: 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																														OUTHI	OUTLOW

Table 1138: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield 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.

19.2.2.6 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 1139: 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																														OUTHI	OUTLOW

Table 1140: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1	OUTHI	0x0	RW	This bit is the vcompout high interrupt.

Table 1140: INTCLR Register Bits

Bit	Name	Reset	RW	Description
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

19.2.2.7 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 1141: 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																														OUTHI	OUTLOW

Table 1142: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield 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.

20. Voltage Regulator Module

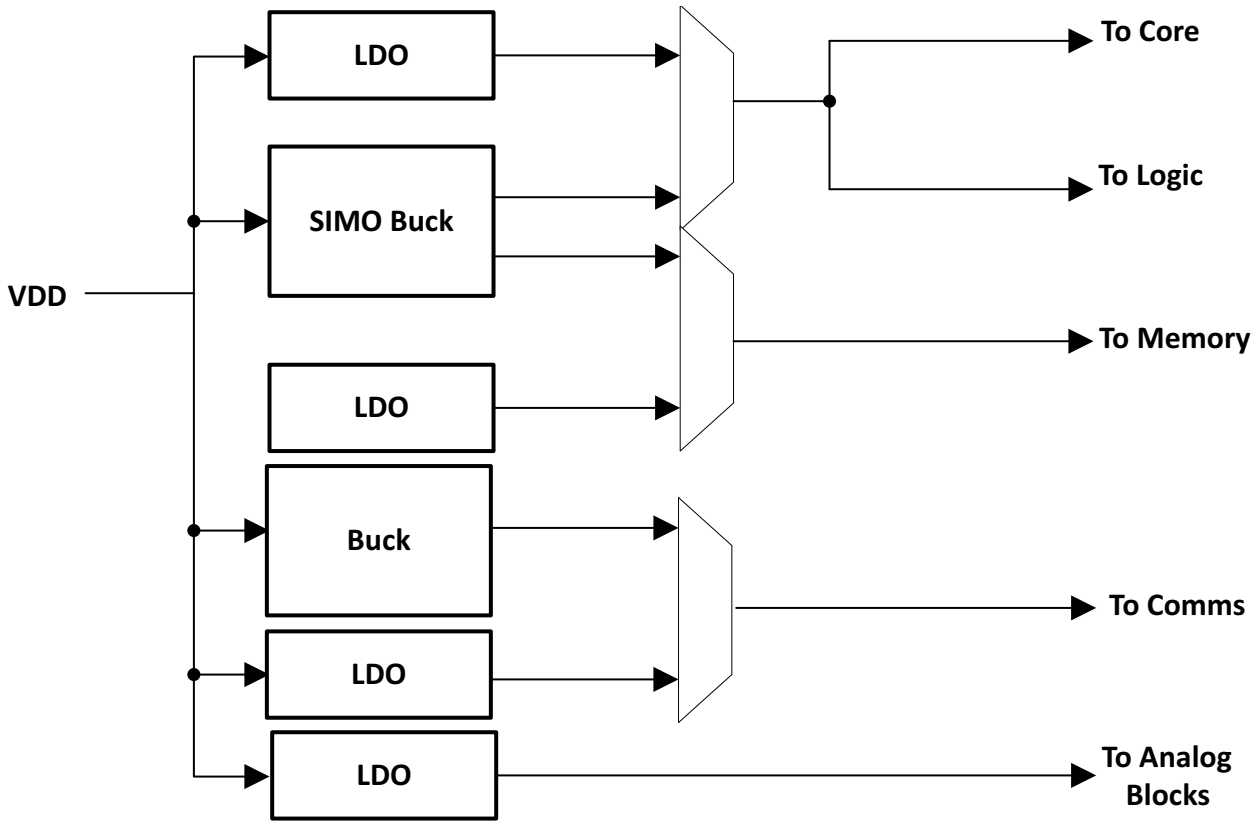


Figure 93. Block Diagram for the Voltage Regulator Module

20.1 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 environments. 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 capacitors/inductors required in buck mode.

The Buck Converters and LDOs of the Voltage Regulator Module are tightly coupled to the various low power modes in the Apollo3 Blue MCU. When the Apollo3 Blue MCU enters deep sleep mode, the Buck Converters will switch into a low power mode to provide very high efficiency at low quiescent current

20.2 SIMO Buck

The SIMO buck sources the primary supplies for the core and memory domains. This buck is a very high efficiency, single-inductor/multiple-output design. The SIMO buck is enabled via an OTP CUSTOMER_TRIM setting. Upon initial reset, if enabled, the SIMO buck will be power up and stabilized through hardware control. The status of the SIMO buck can be queried via the PWRCTRL_SUPPLYSTATUS register (See Section Section 3.5.3.1.2.2 on page 79). The SIMO buck has an efficiency ultra low power mode that is entered automatically via hardware control based on active load current of the system.

For cost/area constrained designs, the SIMO buck can be disabled and on-die LDO regulators can be used. In this configuration, the OTP CUSTOMER_TRIM setting must have the SIMO_BUCK_enable set to '0'. In this configuration, the SIMO buck will remained powered down.

The SIMO buck cannot be dynamically enabled/disabled after initial device reset.

There is also a zero length detect circuit to ensure the regulated voltages from the SIMO buck do not drop out.

20.3 BLE/Burst Buck

The BLE/Burst buck sources the supplies to the BLE radio subsystem as well as the higher voltage required to support the burst mode operation. The BLE/Burst buck must be enabled prior to enabling either the BLE or the burst mode features. The BLE/Burst buck enable bit (See Section Section 3.5.3.1.2.1 on page 79) can be set at any point after reset in software but should be set prior to enabling either the BLE or burst mode features. The status of the BLE/Burst buck can be queried via the PWRCTRL_SUPPLYSTATUS register (See Section Section 3.5.3.1.2.2 on page 79).

For systems that require fast ramp times for the BLE subsystem, the BLE/Burst buck can be enabled by default at reset by setting the BLE_BUCK_enable and BLE_FEATURE_enable bits in the OTP CUSTOMER_TRIM field. If set, hardware will control powering up the buck and sequencing the regulation circuitry as needed at initial power on.

The BLE and burst mode features can be enabled via the FEATUREENABLE register (See Section Section 3.8.2.7 on page 128). Once enabled, hardware controls all sequencing required to enter/exit the various power modes of the BLE/Burst regulators regardless of the configuration.

For cost/area constrained designs, the BLE/Burst buck can be disabled and on-die LDO regulators can be used. In this configuration, the OTP CUSTOMER_TRIM setting must have the BLE_BUCK_enable set to '0'. In this configuration, the SIMO buck will remained powered down.

20.3.1 BLE/Burst Buck Ton Adjustment

Calibration logic within the clock generator block works to check the frequency of the Ton clocks going to the Buck. If the frequency of the Ton clocks is lower than the configured threshold, then the adjustment logic will reduce the Buck charging time for each cycle, which has the effect of increasing the frequency of the charging cycles. If the Ton clocks are higher than the configured threshold, then the adjustment logic will increase the Buck charging time, which has the effect of reducing the frequency of the charging cycle.

The following steps are required to enable the BLE Ton Adjustment:

1. Set the TONADJUSTEN bit field in REG_CLK_GEN_BLEBUCKTONADJ to 0. This will disable the Adjustment until the programming is done.
2. Set the TONADJUSTPERIOD bits field to the adjustment period required. The longer the adjustment period, the more accurate is the adjustment. The shorter the adjustment period, the faster will be the adjustment.
3. Based on the TONADJUSTPERIOD, set the TONHIGHTHRESHOLD and TONLOWTHRESHOLD. The suggested values for the high threshold are #15(94KHz) #2A(47Khz) #A6(12Khz) #29A(3Khz). The suggested values for the low threshold are #A(94KHz) #15(47KHz) #53(12Khz) #14D(3Khz).

Set the TONADJUSTEN bit field in REG_CLK_GEN_BLEBUCKTONADJ to 1. This will enable the Adjustment to start.

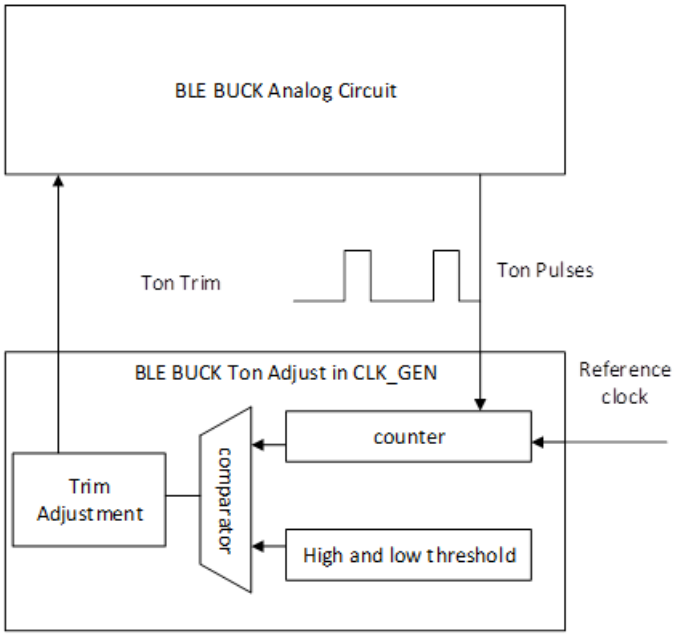


Figure 94. BLE/Burst Buck Ton Adjustment Diagram

20.3.2 BLE/Burst Buck zero length detect

In addition to the Ton adjustment, there is a zero length detect circuit to ensure the regulated voltage does not drop out. The zero length detect is a mechanism to detect the length of time the buck is indicating that the buck voltage is below a certain threshold. If the indicator is continuously asserted beyond a certain time, this indicates that the Buck has not been able to pull the voltage above the threshold. The zero length detect logic will send a flag to the reset generator logic. This will result in either a reset, or an interrupt. The REG_MCU_CTRL_BO_DISABLE register (BODBRDE) controls the BLE/Burst buck local brown out disables.

21. Electrical Characteristics

For all tables T_A = -40ºC to 85ºC, Typical values at 25ºC, 1.8v, unless otherwise specified.

IMPORTANT NOTICE

Specifications and other information in this advanced version of the Apollo3 Blue MCU Datasheet should be regarded as preliminary and subject to change.

21.1 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 1143: Absolute Maximum Ratings

Symbol	Parameter	Test Conditions	Min	Max	Unit
V_{DDP}	Pad supply voltage		-	3.63	V
V_{DDH}	Digital supply voltage		-	3.63	V
V_{DDA}	Analog supply voltage		-	3.63	V
V_{IO}	Voltage on all input and output pins		0	VDDH	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	ºC
T_J	Junction temperature		TBD	85.7	ºC
T_{OP}	Operating temperature		-40	85	ºC
θ_{JA}	Thermal resistance, junction to ambient	BGA Package on 4 layer PCB in still air, 3mW power dissipation		76.2	ºC/W
θ_{JC}	Thermal resistance, junction to package case	BGA Package on 4 layer PCB in still air, 3mW power dissipation		17.0	ºC/W
T_{REFLOW}	Reflow temperature	Reflow Profile per JEDEC J-STD-020D.1		260	ºC
I_{LU}	Latch-up current			100	mA
V_{ESDHBM}	ESD Human Body Model (HBM)			2000	V

Table 1143: Absolute Maximum Ratings

V_{ESDCDM}	ESD Charged Device Model (CDM)			250	V

(1) High side power switches are available on PAD3 and PAD36

(2) A low side power switch is available on PAD37 and PAD41

21.2 Recommended Operating Conditions

Table 1144: Recommended Operating Conditions[a]

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_{DDB}	BLE/Burst Buck Converter supply voltage	1.755		3.63	V
V_{CC}	RF supply voltage	1.755		3.63	V
T_A	Ambient operating temperature	-40		85	ºC
F_{HFRC_LP}	High Frequency RC (HFRC) oscillator frequency - Low Power		48		MHz
F_{HFRC_BRST}	High Frequency RC (HFRC) oscillator frequency - High Performance Burst Mode		96		MHz
F_{LFRC}	Low Frequency RC oscillator frequency (LFRC)		1.024		kHz
F_{XTAL}	Crystal frequency		32.768		kHz

a. $V_{DD} = V_{DDP} = V_{DDA} = V_{DDH}$

21.3 Current Consumption

Table 1145: Current Consumption

Symbol	Parameter	Test Conditions[a,b]	VDD*	Min	Typ	Max	Unit
$I_{RUNLPFB}$	Flash program run current, bucks enabled, Low Power Mode	Executing Coremark from internal Flash memory, cache enabled, HFRC=48MHz, all peripherals disabled, buck converters enabled, 8K SRAM, Flash1 OFF	3.3V		10.3		µA/MHz
			1.8		18.2		µA/MHz
$I_{RUNHPFB}$	Flash program run current, bucks enabled, High Performance Mode	Executing Coremark from internal Flash memory, cache enabled, HFRC=96Hz, all peripherals disabled, buck converters enabled, 8K SRAM, Flash1 OFF	3.3v		27		µA/MHz
			1.8v		41		µA/MHz
$I_{RUNLPFB}$	Flash program run current, bucks enabled, Low Power Mode	Executing Prime Number factorization from internal Flash Memory, cache enabled, HFRC=48MHz, all peripherals disabled, buck converters enabled, 8K SRAM, Flash1 OFF	3.3v		8		µA/MHz
			1.8v		14		µA/MHz
$I_{RUNHPFB}$	Flash program run current, bucks enabled, High Performance Mode	Executing Prime Number factorization from internal Flash Memory, cache enabled, HFRC=96MHz, all peripherals disabled, buck converters enabled, 8K SRAM, Flash1 OFF	3.3v		23		µA/MHz
			1.8v		34		µA/MHz
$I_{RUNWLPFB}$	Flash program run current, bucks enabled, Low Power Mode	Executing while(1) from internal Flash Memory, cache enabled, HFRC=48MHz, all peripherals disabled, buck converters enabled, 8K SRAM, Flash1 OFF	3.3v		6		µA/MHz
			1.8v		10		µA/MHz
I_{SS2}	Sleep mode 2 current	WFI instruction with SLEEP=1, clocks gated, OSC's ON, buck converters enabled, all I/O power domains powered OFF, Flash1 OFF, 8kB SRAM	3.3v		68		µA
			1.8v		80		µA

Table 1145: Current Consumption

$I_{SDS2-384RET}$	System Deep Sleep mode 2 current w/ 384kB retention	WFI instruction with SLEEPDEEP=1, XTAL ON, buck converters enabled in LP mode, all I/O power domains powered OFF, BLE OFF, 384kB SRAM in retention	3.3v		2.7		µA
			1.8v		3.7		µA
$I_{SDS2-8RET}$	System Deep Sleep mode 2 current w/ 8kB retention	WFI instruction with SLEEPDEEP=1, XTAL ON, buck converters enabled in LP mode, all I/O power domains powered OFF, BLE OFF, 8kB SRAM in retention	3.3v		1.4		µA
			1.8v		0.98		µA
I_{SDS3}	System Deep Sleep mode 3 current	WFI instruction with SLEEPDEEP=1, XTAL OFF, buck converters enabled in LP mode, all I/O power domains powered OFF, BLE OFF, all SRAM OFF	3.3v		1.2		µA
			1.8v		0.8		µA
BLE Operating Current							
I_{Active_Rx}	Radio Rx current		3.3V	TBD	3.0	TBD	mA
I_{Active_Tx}	Radio Tx current	Measured at 0 dBm	3.3V	TBD	3.0	TBD	mA
ADC Operating Current[c]							
$I_{ADC\_RUN\_LPM0}$	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				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				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				µA
Flash Memory Operating Current							
$I_{PROGRAM}$	Supply current during a page program		3.3 V			TBD	mA
I_{ERASE}	Supply current during a page erase		3.3 V			TBD	mA
$I_{MASSERASE}$	Supply current during a mass erase		3.3 V			TBD	mA

a. Core clock (HCLK) is 48 MHz for each parameter unless otherwise noted.
b. All values measured at 25°C
c. Base SoC power state is S_{DS2} with minimal SRAM retention. Current represents total current at battery.

21.4 Power Mode Transitions

Table 1146: Power Mode Transitions

Symbol	Parameter	Min	Typ	Max	Unit
Buck mode					
$T_{RUN\_TO\_SLEEP}$	Run to Sleep mode transition time	-	100	-	ns
$T_{RUN\_TO\_DEEPSLEEP}$	Run mode to Deep Sleep mode transition time	-	TBD	-	µs
$T_{SLEEP\_TO\_RUN}$	Sleep to Run mode transition time	-	220	-	ns
$T_{DEEPSLEEP\_TO\_RUN}$	Deep-Sleep to Run mode transition time	-	15	-	µs
LDO mode					
$T_{RUN\_TO\_SLEEP}$	Run to Sleep mode transition time	-	100	-	ns
$T_{RUN\_TO\_DEEPSLEEP}$	Run mode to Deep Sleep mode transition time	-	TBD	-	µs
$T_{SLEEP\_TO\_RUN}$	Sleep to Run mode transition time	-	220	-	ns
$T_{DEEPSLEEP\_TO\_RUN}$	Deep-Sleep to Run mode transition time	-	15	-	µs

21.5 Clocks/Oscillators

Table 1147: Clocks/Oscillators

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{HFRC}	HFRC frequency		-	48	-	MHz
F_{LFRC}	LFRC frequency		-	1024	-	Hz
F_{XT}	XT frequency		-	32.768	-	kHz
DC_{HFRC}	HFRC duty cycle			50		%
C_{INX}	Internal XI/XO pin capacitance			3.4		pF
$C_{EXT\_XT\_TOL}$	Allowed external XI/XO pin capacitance per pin		-	-	7	pF
F_{OF}	XT oscillator failure detection frequency			8		kHz
OA_{XT}	XT oscillation allowance	At 25°C using a 32.768 kHz tuning fork crystal	320		-	kΩ

Table 1148: BLE Crystal Oscillator

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{XTAL}	Crystal frequency			32		MHz
ΔFXTAL	Frequency tolerance	Untrimmed; include initial tolerance/aging/temperature drift	-40		40	ppm
C_L	Crystal load capacitance			8		pF
ESR	Equivalent serial resistance				100	Ohms
T_{XTAL}	Startup time			1		ms

21.6 Bluetooth Low Energy (BLE)

Symbol	Parameter[a]	Test Conditions	Min	Typ	Max	Unit
AC Characteristics - Rx						
R_{SENS}	Receiver sensitivity	1 Mbps BLE ideal transmitter, <=37 bytes, PER < 30.8%	-92	-93	-94	dBm
$R_{SENS, VAR}$	Rx sensitivity variance between channels		-0.5		0.5	dB
$P_{RX, MAX}$	Maximum receiver input power	PER < 30.8%			0	dBm
C/Ico-channel	Co-channel interference	Wanted signal at –67dBm, modulated interferer in channel, PER < 30.8%		7		dB
PB	Out of band blocking	30 MHz to 2000 MHz		-5		dBm
	Out of band blocking	2003 MHz to 2399 MHz		-15		dBm
	Out of band blocking	2484 MHz to 2997 MHz		-15		dBm
	Out of band blocking	3000 MHz to 12.75 GHz		-5		dBm
F_{ET}	Frequency error tolerance		-125		125	kHz
AC Characteristics - Tx						
P_{OUT_PEAK}	Peak output power		3	3.5	4	dBm
P_{OUT_AVG}	Average Tx output power		2.5	3	3.5	dBm
$P_{OUT, VAR}$	Average Tx output power variance between channels		-0.5		0.5	dB
P_{OUT_STEP}	Power control step			TBD		dBm
P_{OUT_HD2}	Second harmonic output power level			-40	-30	dBm
P_{OUT_HD3}	Third harmonic output power level			-40	-30	dBm
P_{OUT_HD4}	Fourth harmonic output power level			-40	-30	dBm

a. FCC and BQB test reports are available upon request.

21.7 Analog-to-Digital Converter (ADC)

Table 1149: 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

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
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
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	-	.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_W-START_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

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
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
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,[a] 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						

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
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
E_{ADC_OFFEST}	Offset error		1		1	%FS
E_{ADC_GAIN}	Gain error				1	%FS
INTERNAL TEMPERATURE SENSOR						
E_{TEMP}	Temperature sensor accuracy			+/- 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

a. Buck Mode not supported on wafer package

21.8 Buck Converter

Table 1150: SIMO Buck Converter

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
L_{SBUCK}	Buck converter inductance (V_{SIMO})			2.2		µH
C_{BUCK}	Buck converter output capacitance (2) (V_{DDC}, V_{DDF})			2.2		µF

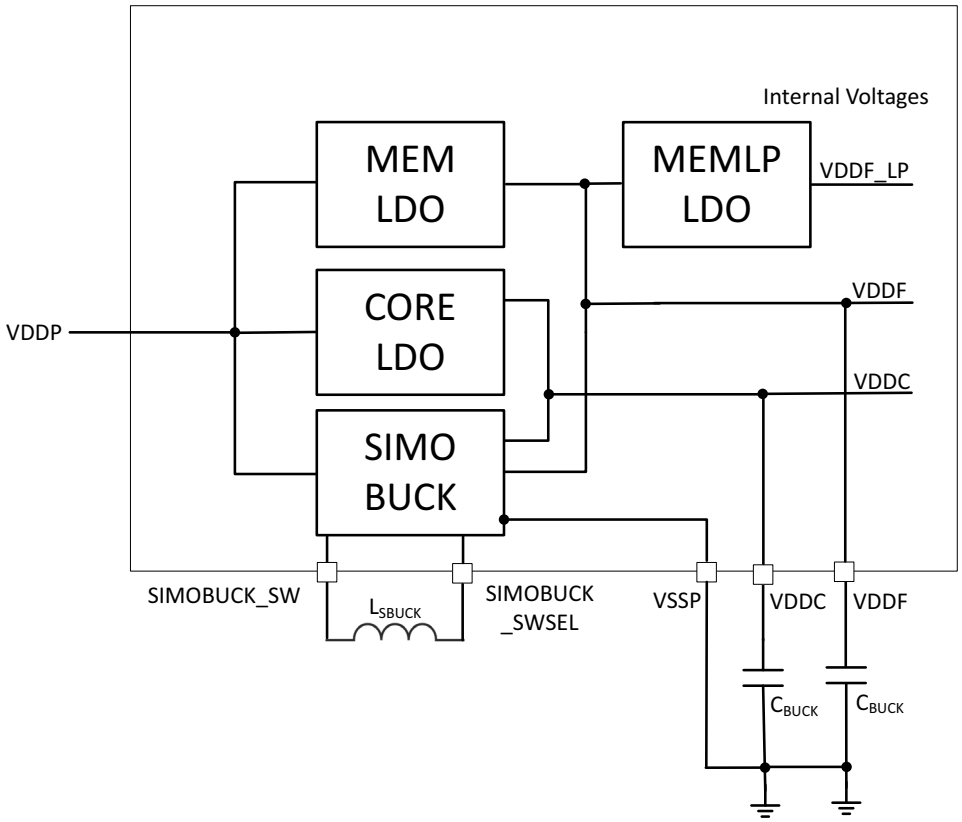


Figure 95. External Components for SIMO Buck

Table 1151: BLE Buck Converter

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
$L_{BLEBUCK}$	Buck converter inductance (V_{DDBH})			1.0		µH
$C_{BLEBUCK}$	Buck converter output capacitance (V_{DDBH})			4.7		µF
$C_{DCDCREF}$	BLE ref voltage capacitance ($V_{VDCDCRF}$)			1.0		µF

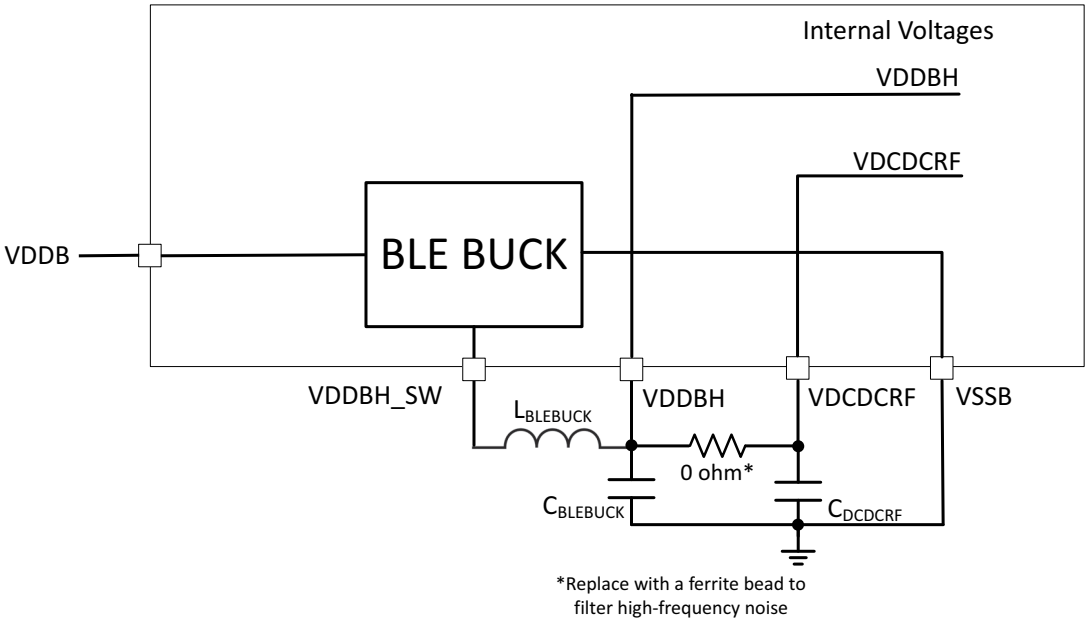


Figure 96. External Components for BLE Buck

21.9 Power-On RESET (POR) and Brown-Out Detector (BOD)

Table 1152: Power-On Reset (POR) and Brown-Out Detector (BOD)

Symbol	Parameter	Min	Typ	Max	Unit
V_{POR_RISING}	POR rising threshold voltage	1.62		1.755	V
$V_{BODL\_FALLING}$	Brownout detection low falling threshold voltage	1.62		1.755	V

21.10Resets

Table 1153: Resets

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
T_{RST}	nRST pulse width to guarantee reset assertion	2KV/s supply slew rate		TBD		µs
T_{POR}	POR detect to nRST deassertion delay	2KV/s supply slew rate	TBD	TBD	TBD	µs
$T_{POR2HRST}$	Delay from nRST deassertion to HRESET deassertion	2KV/s supply slew rate	TBD		TBD	µs
T_{RSTDLY}	nRST reset delay from internal BODL	2KV/s supply slew rate		TBD		µs
T_{SOFT}	Software initiated reset delay	2KV/s supply slew rate		TBD		µs

21.11Voltage Comparator (VCOMP)

Table 1154: 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			TBD		V
I_{COMPIN_LEAK}	Input leakage current			1		nA
T_{COMP_RTRIG}	Rising voltage trigger response time				38	µs
T_{COMP_FTRIG}	Falling voltage trigger response time				12	µs
V_{HYST}	Hysteresis		30			mV

21.12 Inter-Integrated Circuit (I^2C) Interface

Table 1155: Inter-Integrated Circuit (I^2C) Interface

Symbol	Parameter	VCC	Min	Typ	Max	Unit
f_{SCL}	SCL input clock frequency	1.7 V - 3.6 V	10		1000	kHz
t_{LOW}	Low period of SCL clock	1.7 V - 3.6 V	1.3			µs
t_{HIGH}	High period of SCL clock	1.7 V - 3.6 V	600			ns
t_{RISE}	Rise time of SDA and SCL	1.7 V - 3.6 V			300	ns
t_{FALL}	Fall time of SDA and SCL	1.7 V - 3.6 V			300	ns
$t_{HD:STA}$	START condition hold time	1.7 V - 3.6 V	600			ns
$t_{SU:STA}$	START condition setup time	1.7 V - 3.6 V	600			ns
$t_{SU:DAT}$	SDA setup time	1.7 V - 3.6 V	100			ns
$t_{HD:DAT}$	SDA hold time	1.7 V - 3.6 V	0			ns
$t_{SU:STO}$	STOP condition setup time	1.7 V - 3.6 V	600			ns
t_{BUF}	Bus free time before a new transmission	1.7 V - 3.6 V	1.3			µs

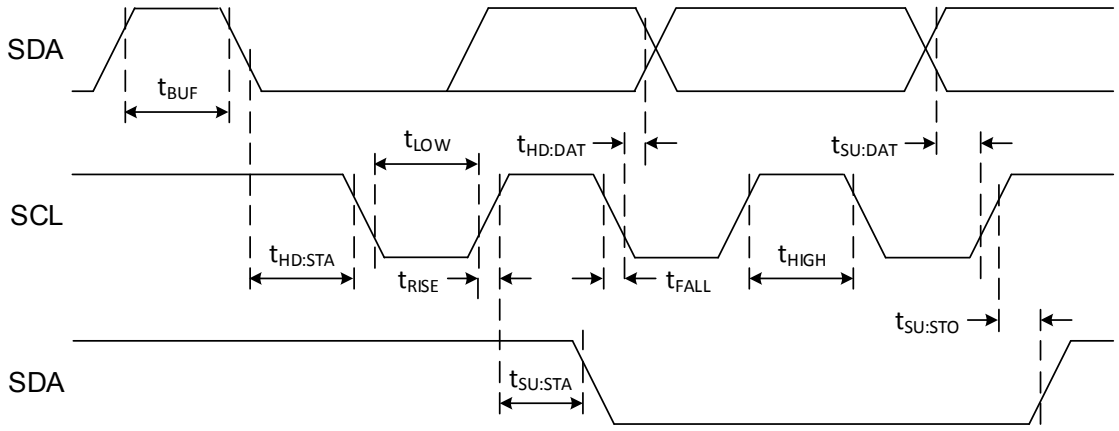


Figure 97. I^2C Timing

21.13 Serial Peripheral Interface (SPI) Master Interface

Table 1156: Serial Peripheral Interface (SPI) Master Interface

Symbol	Parameter	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range	-	8	24	MHz
B_{FIFO}	FIFO size		32		Bytes
T_{SCLK_LO}	Clock low time	$1/2F_{S-CLK(max)}$	-	-	s
T_{SCLK_HI}	Clock high time	$1/2F_{S-CLK(max)}$	-	-	s
T_{SU_MI}	MISO input data setup time	-	-	-	ns
T_{HD_MI}	MISO input data hold time	-	-	-	ns
T_{HD_MO}	MOSI output data hold time	-	-	-	ns
$T_{VALID,MO}$	MOSI output data valid time	-	-	-	ns

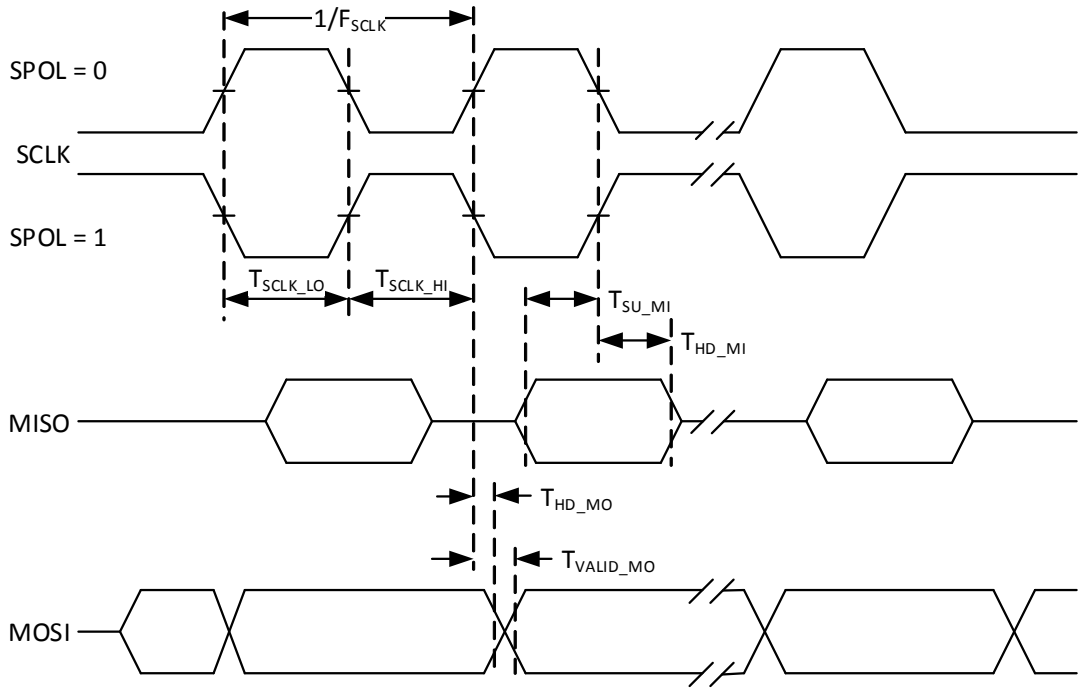


Figure 98. SPI Master Mode, Phase = 0

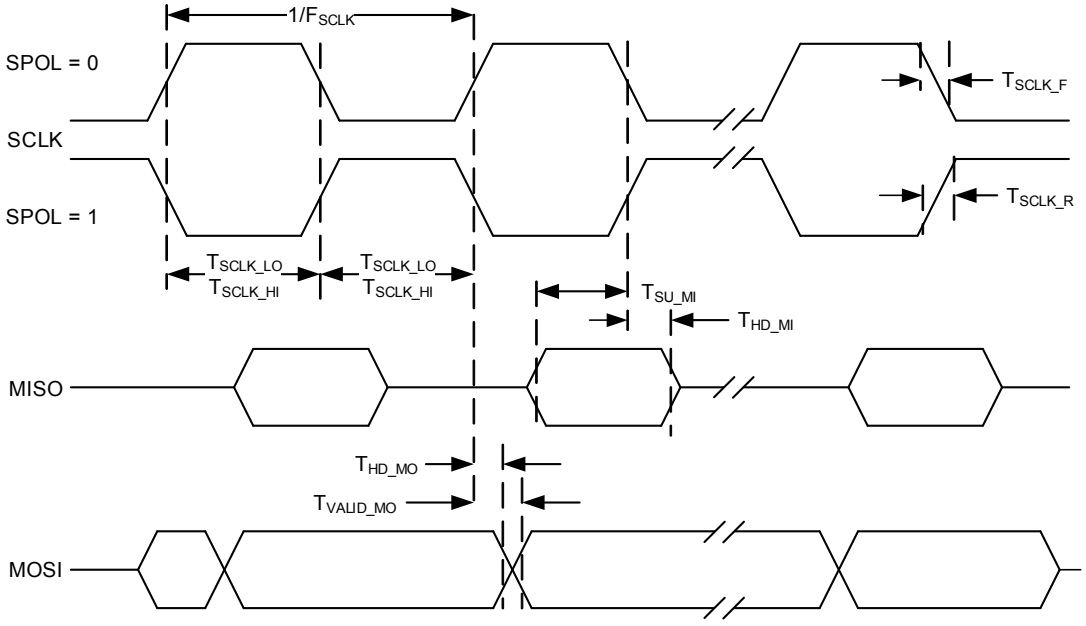


Figure 99. SPI Master Mode, Phase = 1

21.14Serial Peripheral Interface (SPI) Slave Interface

Table 1157: Serial Peripheral Interface (SPI) Slave Interface

Symbol	Parameter	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range	-	-	8	MHz
B_{FIFO}	FIFO size		128		Bytes
T_{SCLK_LO}	Clock low time	$1/2F_{S-CLK(max)}$	-	-	s
T_{SCLK_HI}	Clock high time	$1/2F_{S-CLK(max)}$	-	-	s
T_{CE_LEAD}	Chip enable low to first SCLK edge	-	-	-	ns
T_{CE_LAG}	Chip enable high to last SCLK edge	-	-	-	ns
T_{CE_SDO}	Chip enable low to MISO data output	-	-	-	ns
T_{CE_SDZ}	Chip enable high to MISO data tri-state	-	-	-	ns
T_{SU_SI}	MOSI input data setup time	-	-	-	ns
T_{HD_SI}	MOSI input data hold time	-	-	-	ns
T_{HD_SO}	MISO output data hold time	-	-	-	ns
$T_{VALID,SO}$	MISO output data valid time	-	-	-	ns

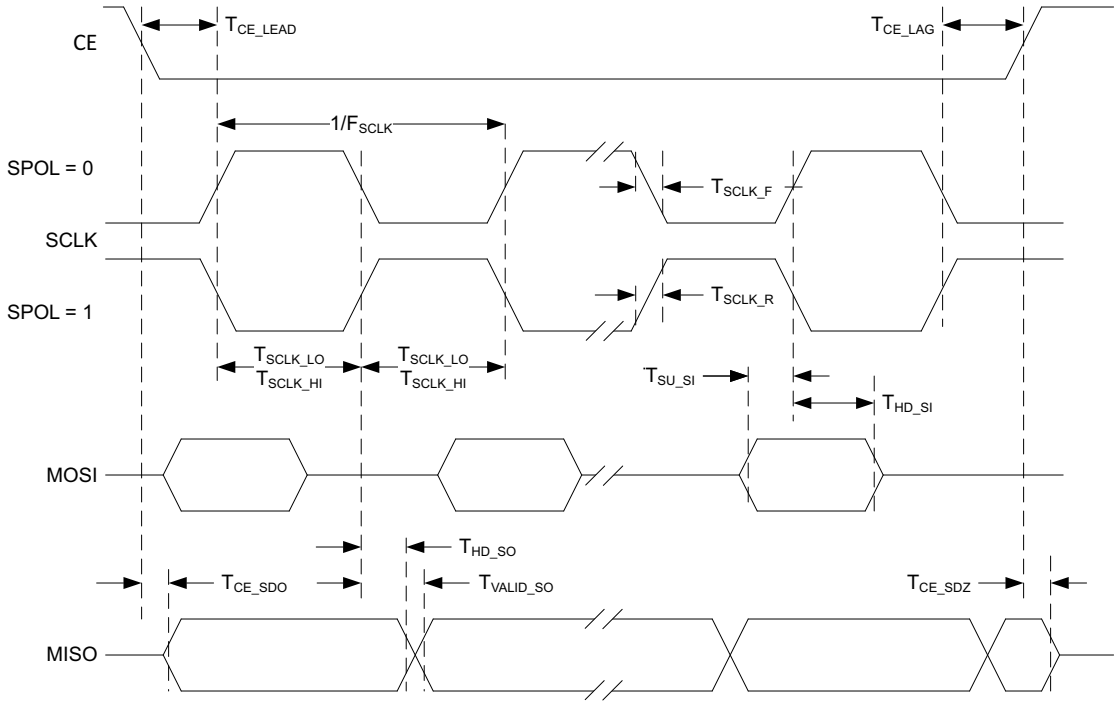


Figure 100. SPI Slave Mode, Phase = 0

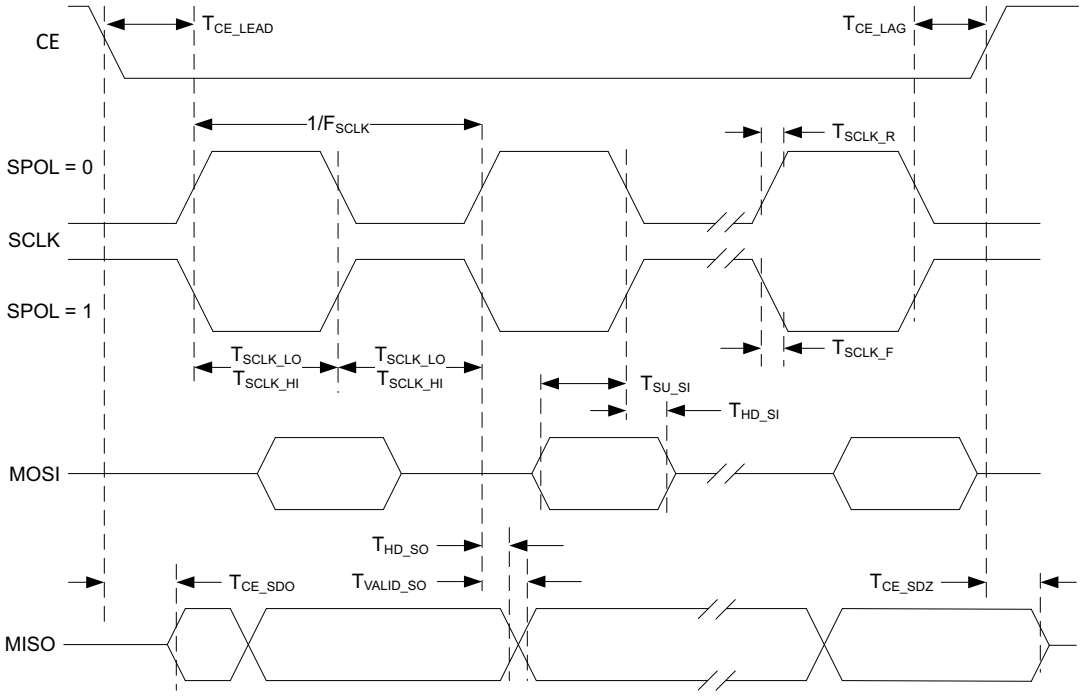


Figure 101. SPI Slave Mode, Phase = 1

21.15PDM Interface

Table 1158: Pulse Density Modulation (PDM) Interface

Symbol	Parameter	Min	Typ	Max	Unit
DC_{PDMCLK}	PDM clock duty cycle[a]	45	-	55	%
DC_{PDMCLK_HI}	PDM high frequency clock duty cycle[b]	48	-	52	%
T_{PDM_RISE}	PDM clock and data rise time	-	-	TBD	ns
T_{PDM_FALL}	PDM clock and data fall time	-	-	TBD	ns
T_{SU_PDM}	PDM input data setup time	-	-	TBD	ns
T_{HD_PDM}	PDM input data hold time	TBD	-	-	ns

a. 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%.

b. 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%. Also, using Pad 37 for PDM_CLK supports only the lower frequency DC range (DC_{PDMCLK}) and is not guaranteed to meet the higher frequency DC range.

21.16I2S Interface

Table 1159: Inter-Integrated Serial (I2S) Interface

Symbol	Parameter	Min	Typ	Max	Unit
F_{BCLK}	I2S input BCLK frequency range	TBD		TBD	MHz
F_{WDCLK}	I2S input WDCLK frequency range	TBD		TBD	MHz
DC_{BCLK}	I2S BCLK duty cycle	40	-	60	%
DC_{WDCLK}	I2S WDCLK duty cycle	40	-	69	%
T_{I2S_RISE}	I2S clock and data rise time	-	-	TBD	ns
T_{I2S_FALL}	I2S clock and data fall time	-	-	TBD	ns
T_{SU_I2S}	I2S input data setup time	-	-	TBD	ns
T_{HD_I2S}	I2S input data hold time	TBD	-	-	ns

21.17Universal Asynchronous Receiver/Transmitter (UART)

Table 1160: Universal Asynchronous Receiver/Transmitter (UART)

Symbol	Parameter	Min	Typ	Max	Unit
F_{BAUD}	UART baud rate		-	921600	bps

21.18Counter/Timer (CTIMER)

Table 1161: Counter/Timer (CTIMER)

Symbol	Parameter	Min	Typ	Max	Unit
F_{CTIMER}	Input frequency	-	-	24	MHz
T_{CTIMER}	Capture pulse width		-	-	

21.19Flash Memory

Table 1162: Flash Memory

Symbol	Parameter	Min	Typ	Max	Unit
PE_{CYC}	Program/erase cycles before failure	10,000	-	-	cycles
T_{FDR}	Data retention @85C	10	-	-	years
T_{PAGE_ERASE}	Single page erase time (8192 bytes)	10 (TBD)	-	20 (TBD)	ms
T_{MASS_ERASE}	Mass erase time	10 (TBD)	-	20 (TBD)	ms

21.20General Purpose Input/Output (GPIO)

All GPIOs have Schmitt trigger inputs.

Table 1163: General Purpose Input/Output (GPIO)

Symbol	Parameter	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	-		TBD	pF
R_{PU}	Pull-up resistance	-	63	-	kΩ
R_{PD}	Pull-down resistance	-	63	-	kΩ
$R_{PUI2C00}$	I2C pad pull-up resistance, RSEL = 0x00	-	2	-	kΩ
$R_{PUI2C01}$	I2C pad pull-up resistance, RSEL = 0x01	-	6	-	kΩ
$R_{PUI2C10}$	I2C pad pull-up resistance, RSEL = 0x10	-	12	-	kΩ
$R_{PUI2C11}$	I2C pad pull-up resistance, RSEL = 0x11	-	24	-	kΩ
I_{IN}	Input pin leakage current	-	1	-	nA
I_{INOD}	Open drain output leakage current	-	1	-	nA
STANDARD GPIOs					

Table 1163: General Purpose Input/Output (GPIO)

T_{RISE_STD}	Rise time	-	-	TBD	ns
T_{FALL_STD}	Fall time	-	-	TBD	ns
I_{SRC_STD}	Output source current, 2mA drive strength	-	3.5	-	mA
I_{SNK_STD}	Output sink current, 2mA drive strength	-	3.4	-	mA
I_{SRC_STD}	Output source current, 4mA drive strength	-	7.1	-	mA
I_{SNK_STD}	Output sink current, 4mA drive strength	-	6.8	-	mA
I_{SRC_STD}	Output source current, 8mA drive strength	-	14.1	-	mA
I_{SNK_STD}	Output sink current, 8mA drive strength	-	13.5	-	mA
I_{SRC_STD}	Output source current, 12mA drive strength	-	21.1	-	mA
I_{SNK_STD}	Output sink current, 12mA drive strength	-	20.2	-	mA
POWER SWITCH GPIOs					
R_{SRC_PWR}	High side power switch resistance	-	1.02	1.7	Ω
I_{SRC_PWR}	High side power switch source current	-	-	100	mA
$I_{SRC\_P-WR\_LKG}$	High side power switch source leakage current	TBD	2.58	TBD	nA
R_{SNK_PWR}	Low side power switch resistance	-	1	TBD	Ω
I_{SNK_PWR}	Low side power switch sink current	-	-	30	mA
$I_{SNK\_P-WR\_LKG}$	Low side power switch source leakage current	TBD	TBD	TBD	nA

21.21Serial Wire Debug (SWD)

Table 1164: Serial Wire Debug (SWD)

Symbol	Parameter	Min	Typ	Max	Unit
T_{SWDCK_HI}	SWDCK clock high period				µs
T_{SWDCK_LO}	SWDCK clock low period				µs
T_{OS_SWD}	SWDIO output skew to falling edge of SWDCLK				ns
T_{SU_SWD}	Input setup time between SWDIO and rising edge SWDCK				ns
T_{HD_SWD}	Input hold time between SWDIO and rising edge SWDCK				ns

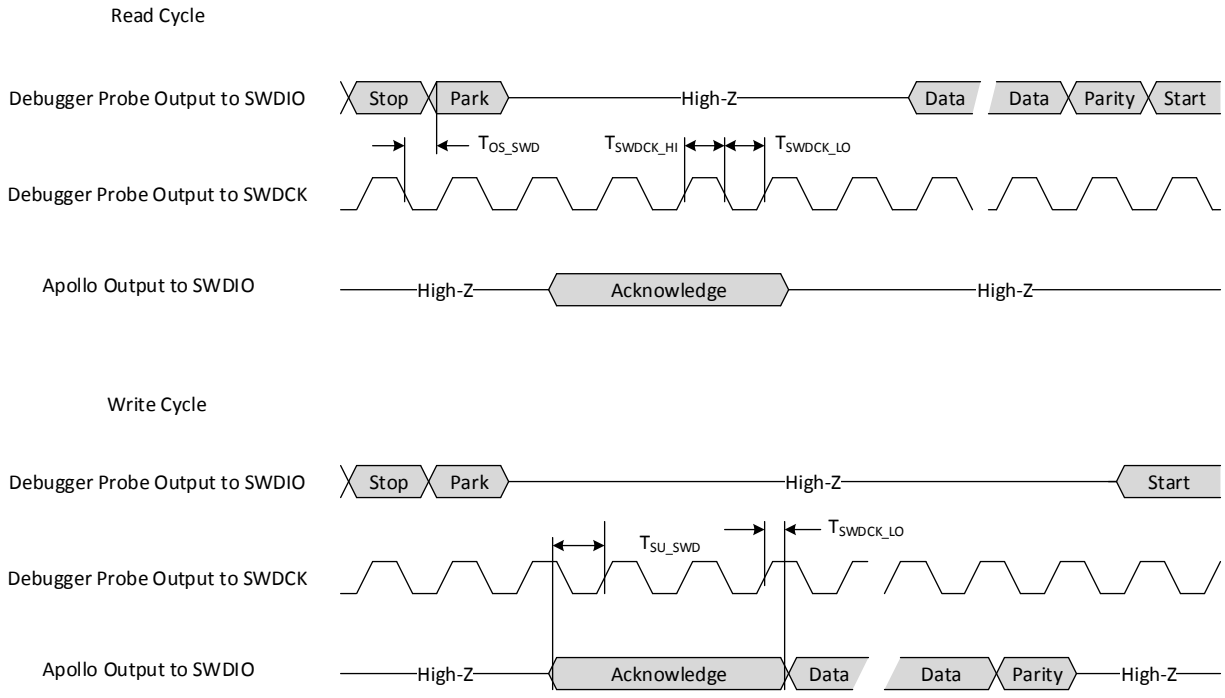
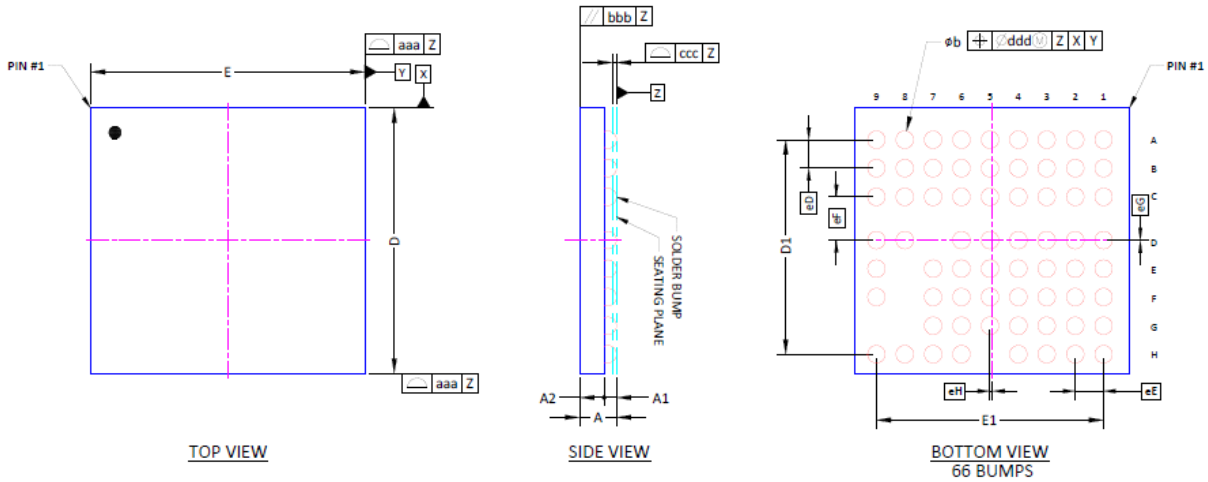


Figure 102. Serial Wire Debug Timing

22. Package Mechanical Information

22.1 CSP Package



Control dimensions are in millimeter

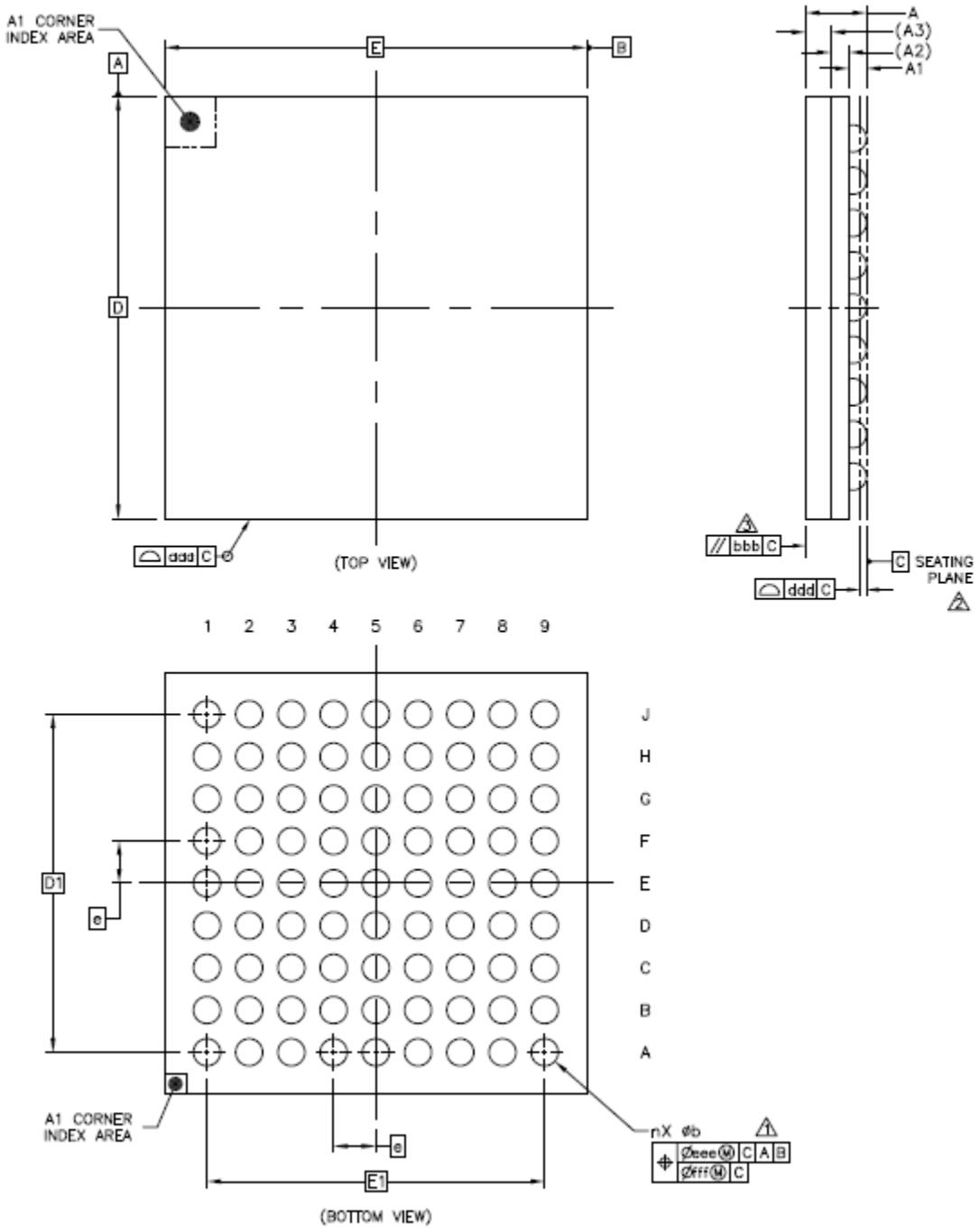
Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	0.397	0.452	0.507	0.016	0.018	0.020
A1	0.122	0.152	0.182	0.005	0.006	0.007
A2	0.275	0.300	0.325	0.011	0.012	0.013
D	3.2320	3.2520	3.2720	0.127	0.128	0.129
E	3.3554	3.3754	3.3954	0.132	0.133	0.134
D1	-	2.629	-	-	0.103	-
E1	-	2.797	-	-	0.110	-
eD, eE	-	0.350	-	-	0.014	-
eF	-	0.531	-	-	0.021	-
eG	-0.004	0.006	0.016	-0.0002	0.000	0.0006
eH	0.022	0.032	0.042	0.0009	0.001	0.0017
b	0.193	0.218	0.243	0.008	0.009	0.010
aaa	0.10			0.004		
bbb	0.10			0.004		
ccc	0.05			0.002		
ddd	0.05			0.002		

Notes:

1. **Dimension b is measured at the maximum solder bump diameter, parallel to primary datum C.**
2. **This pod is for device Apollo3 without backside coating.**
3. **eF is the distance between the center lines of row C and row D of balls.**
4. **eG is the vertical offset from center of package to center of D5 ball.**
5. **eH is the horizontal offset from center of package to center of D5 ball.**
6. **In the bottom view, the D5 ball center is above and to the left of the center of package, making the overall D5 offset from the center of package toward the top-left quadrant (A9 corner) of the package.**

Figure 103. CSP Package Drawing

22.2 BGA Package[1]



1. All dimensions in mm unless otherwise noted.

	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	0.8
STAND OFF	A1	0.16	---	0.26
SUBSTRATE THICKNESS	A2		0.21	REF
MOLD THICKNESS	A3		0.3	REF
BODY SIZE	D		5	BSC
	E		5	BSC
BALL DIAMETER			0.3	
BALL OPENING			0.275	
BALL WIDTH	b	0.27	---	0.37
BALL PITCH	e		0.5	BSC
BALL COUNT	n		81	
EDGE BALL CENTER TO CENTER	D1		4	BSC
	E1		4	BSC
BODY CENTER TO CONTACT BALL	SD		---	BSC
	SE		---	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.1	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

Figure 104. BGA Package Drawing

23. Appendix 1. FLASH OTP 0 Customer Info Space (Info0)

23.1 Flash OTP INSTANCE0 INFO0 Words

Customer OTP Block 0 of Instance 0.

INSTANCE 0 BASE ADDRESS:0x00000000

This is the detailed description of the contents of the Customer OTP for Apollo3.

23.1.1 Register Memory Map

Table 1165: Flash OTP INSTANCE0 INFO0 Register Map

Address(s)	Register Name	Description
0x00000000	SIGNATURE0	INFO0 Signature
0x00000004	SIGNATURE1	INFO0 Signature
0x00000008	SIGNATURE2	INFO0 Signature
0x0000000C	SIGNATURE3	INFO0 Signature
0x00000010	SECURITY	Security protection bits
0x00000014	CUSTOMERTRIM	Customer trim values
0x00000018	CUSTOMERTRIM2	Customer trim values word2
0x00000020	SECURITYOVR	Security Override configuration bits
0x00000024	SECURITYWIREDCFG	Security Wired configuration bits
0x00000028	SECURITYWIREDIFCCFG0	Security Wired Interface configuration word0
0x0000002C	SECURITYWIREDIFCCFG1	Security Wired Interface configuration word1
0x00000030	SECURITYWIREDIFCCFG2	Security Wired Interface configuration word2
0x00000034	SECURITYWIREDIFCCFG3	Security Wired Interface configuration word3
0x00000038	SECURITYWIREDIFCCFG4	Security Wired Interface configuration word4
0x0000003C	SECURITYWIREDIFCCFG5	Security Wired Interface configuration word5
0x00000040	SECURITYVERSION	Security version field
0x00000050	SECURITYSRAMRESV	SRAM Reserved for Application Scratch space
0x000001F8	WRITEPROTECTL	Flash write-protection bits.
0x000001FC	WRITEPROTECTH	Flash write-protection bits.
0x00000200	COPYPROTECTL	Flash copy/read-protection bits.
0x00000204	COPYPROTECTH	Flash copy/read-protection bits.
0x000009F8	WRITEPROTECTSBLL	Flash write-protection bits.
0x000009FC	WRITEPROTECTSBLH	Flash write-protection bits.
0x00000A00	COPYPROTECTSBLL	Flash copy/read-protection bits.
0x00000A04	COPYPROTECTSBLH	Flash copy/read-protection bits.
0x00000C00	MAINPTR0	main firmware pointer 0
0x00000C04	MAINPTR1	main firmware pointer 1
0x00000C08	KREVTRACK	KEK Revocation Tracker
0x00000C0C	AREVTRACK	AUTH Revocation Tracker
0x00000C10	OTADESCRIPTOR	OTA Descriptor Pointer
0x00000FF8	MAINCNT0	main Index Counter 0
0x00000FFC	MAINCNT1	main Index Counter 1
0x00001800	CUSTKEKW0	Customer KEK Word0
0x00001804	CUSTKEKW1	Customer KEK Word1
0x00001808	CUSTKEKW2	Customer KEK Word2
0x0000180C	CUSTKEKW3	Customer KEK Word3

Table 1165: Flash OTP INSTANCE0 INFO0 Register Map

Address(s)	Register Name	Description
0x00001810	CUSTKEKW4	Customer KEK Word4
0x00001814	CUSTKEKW5	Customer KEK Word5
0x00001818	CUSTKEKW6	Customer KEK Word6
0x0000181C	CUSTKEKW7	Customer KEK Word7
0x00001820	CUSTKEKW8	Customer KEK Word0
0x00001824	CUSTKEKW9	Customer KEK Word9
0x00001828	CUSTKEKW10	Customer KEK Word10
0x0000182C	CUSTKEKW11	Customer KEK Word11
0x00001830	CUSTKEKW12	Customer KEK Word12
0x00001834	CUSTKEKW13	Customer KEK Word13
0x00001838	CUSTKEKW14	Customer KEK Word14
0x0000183C	CUSTKEKW15	Customer KEK Word15
0x00001840	CUSTKEKW16	Customer KEK Word16
0x00001844	CUSTKEKW17	Customer KEK Word17
0x00001848	CUSTKEKW18	Customer KEK Word18
0x0000184C	CUSTKEKW19	Customer KEK Word19
0x00001850	CUSTKEKW20	Customer KEK Word20
0x00001854	CUSTKEKW21	Customer KEK Word21
0x00001858	CUSTKEKW22	Customer KEK Word22
0x0000185C	CUSTKEKW23	Customer KEK Word23
0x00001860	CUSTKEKW24	Customer KEK Word24
0x00001864	CUSTKEKW25	Customer KEK Word25
0x00001868	CUSTKEKW26	Customer KEK Word26
0x0000186C	CUSTKEKW27	Customer KEK Word27
0x00001870	CUSTKEKW28	Customer KEK Word28
0x00001874	CUSTKEKW29	Customer KEK Word29
0x00001878	CUSTKEKW30	Customer KEK Word30
0x0000187C	CUSTKEKW31	Customer KEK Word31
0x00001880	CUSTAUTHW0	Customer AUTH Key Word0
0x00001884	CUSTAUTHW1	Customer AUTH Key Word1
0x00001888	CUSTAUTHW2	Customer AUTH Key Word2
0x0000188C	CUSTAUTHW3	Customer AUTH Key Word3
0x00001890	CUSTAUTHW4	Customer AUTH Key Word4
0x00001894	CUSTAUTHW5	Customer AUTH Key Word5
0x00001898	CUSTAUTHW6	Customer AUTH Key Word6
0x0000189C	CUSTAUTHW7	Customer AUTH Key Word7
0x000018A0	CUSTAUTHW8	Customer AUTH Key Word0
0x000018A4	CUSTAUTHW9	Customer AUTH Key Word9

Table 1165: Flash OTP INSTANCE0 INFO0 Register Map

Address(s)	Register Name	Description
0x000018A8	CUSTAUTHW10	Customer AUTH Key Word10
0x000018AC	CUSTAUTHW11	Customer AUTH Key Word11
0x000018B0	CUSTAUTHW12	Customer AUTH Key Word12
0x000018B4	CUSTAUTHW13	Customer AUTH Key Word13
0x000018B8	CUSTAUTHW14	Customer AUTH Key Word14
0x000018BC	CUSTAUTHW15	Customer AUTH Key Word15
0x000018C0	CUSTAUTHW16	Customer AUTH Key Word16
0x000018C4	CUSTAUTHW17	Customer AUTH Key Word17
0x000018C8	CUSTAUTHW18	Customer AUTH Key Word18
0x000018CC	CUSTAUTHW19	Customer AUTH Key Word19
0x000018D0	CUSTAUTHW20	Customer AUTH Key Word20
0x000018D4	CUSTAUTHW21	Customer AUTH Key Word21
0x000018D8	CUSTAUTHW22	Customer AUTH Key Word22
0x000018DC	CUSTAUTHW23	Customer AUTH Key Word23
0x000018E0	CUSTAUTHW24	Customer AUTH Key Word24
0x000018E4	CUSTAUTHW25	Customer AUTH Key Word25
0x000018E8	CUSTAUTHW26	Customer AUTH Key Word26
0x000018EC	CUSTAUTHW27	Customer AUTH Key Word27
0x000018F0	CUSTAUTHW28	Customer AUTH Key Word28
0x000018F4	CUSTAUTHW29	Customer AUTH Key Word29
0x000018F8	CUSTAUTHW30	Customer AUTH Key Word30
0x000018FC	CUSTAUTHW31	Customer AUTH Key Word31
0x00001900	CUSTPUBKEYW0	Customer Public Key Word0
0x00001904	CUSTPUBKEYW1	Customer Public Key Word1
0x00001908	CUSTPUBKEYW2	Customer Public Key Word2
0x0000190C	CUSTPUBKEYW3	Customer Public Key Word3
0x00001910	CUSTPUBKEYW4	Customer Public Key Word4
0x00001914	CUSTPUBKEYW5	Customer Public Key Word5
0x00001918	CUSTPUBKEYW6	Customer Public Key Word6
0x0000191C	CUSTPUBKEYW7	Customer Public Key Word7
0x00001920	CUSTPUBKEYW8	Customer Public Key Word0
0x00001924	CUSTPUBKEYW9	Customer Public Key Word9
0x00001928	CUSTPUBKEYW10	Customer Public Key Word10
0x0000192C	CUSTPUBKEYW11	Customer Public Key Word11
0x00001930	CUSTPUBKEYW12	Customer Public Key Word12
0x00001934	CUSTPUBKEYW13	Customer Public Key Word13
0x00001938	CUSTPUBKEYW14	Customer Public Key Word14
0x0000193C	CUSTPUBKEYW15	Customer Public Key Word15

Table 1165: Flash OTP INSTANCE0 INFO0 Register Map

Address(s)	Register Name	Description
0x00001940	CUSTPUBKEYW16	Customer Public Key Word16
0x00001944	CUSTPUBKEYW17	Customer Public Key Word17
0x00001948	CUSTPUBKEYW18	Customer Public Key Word18
0x0000194C	CUSTPUBKEYW19	Customer Public Key Word19
0x00001950	CUSTPUBKEYW20	Customer Public Key Word20
0x00001954	CUSTPUBKEYW21	Customer Public Key Word21
0x00001958	CUSTPUBKEYW22	Customer Public Key Word22
0x0000195C	CUSTPUBKEYW23	Customer Public Key Word23
0x00001960	CUSTPUBKEYW24	Customer Public Key Word24
0x00001964	CUSTPUBKEYW25	Customer Public Key Word25
0x00001968	CUSTPUBKEYW26	Customer Public Key Word26
0x0000196C	CUSTPUBKEYW27	Customer Public Key Word27
0x00001970	CUSTPUBKEYW28	Customer Public Key Word28
0x00001974	CUSTPUBKEYW29	Customer Public Key Word29
0x00001978	CUSTPUBKEYW30	Customer Public Key Word30
0x0000197C	CUSTPUBKEYW31	Customer Public Key Word31
0x00001980	CUSTPUBKEYW32	Customer Public Key Word32
0x00001984	CUSTPUBKEYW33	Customer Public Key Word33
0x00001988	CUSTPUBKEYW34	Customer Public Key Word34
0x0000198C	CUSTPUBKEYW35	Customer Public Key Word35
0x00001990	CUSTPUBKEYW36	Customer Public Key Word36
0x00001994	CUSTPUBKEYW37	Customer Public Key Word37
0x00001998	CUSTPUBKEYW38	Customer Public Key Word38
0x0000199C	CUSTPUBKEYW39	Customer Public Key Word39
0x000019A0	CUSTPUBKEYW40	Customer Public Key Word40
0x000019A4	CUSTPUBKEYW41	Customer Public Key Word41
0x000019A8	CUSTPUBKEYW42	Customer Public Key Word42
0x000019AC	CUSTPUBKEYW43	Customer Public Key Word43
0x000019B0	CUSTPUBKEYW44	Customer Public Key Word44
0x000019B4	CUSTPUBKEYW45	Customer Public Key Word45
0x000019B8	CUSTPUBKEYW46	Customer Public Key Word46
0x000019BC	CUSTPUBKEYW47	Customer Public Key Word47
0x000019C0	CUSTPUBKEYW48	Customer Public Key Word48
0x000019C4	CUSTPUBKEYW49	Customer Public Key Word49
0x000019C8	CUSTPUBKEYW50	Customer Public Key Word50
0x000019CC	CUSTPUBKEYW51	Customer Public Key Word51
0x000019D0	CUSTPUBKEYW52	Customer Public Key Word52
0x000019D4	CUSTPUBKEYW53	Customer Public Key Word53

Table 1165: Flash OTP INSTANCE0 INFO0 Register Map

Address(s)	Register Name	Description
0x000019D8	CUSTPUBKEYW54	Customer Public Key Word54
0x000019DC	CUSTPUBKEYW55	Customer Public Key Word55
0x000019E0	CUSTPUBKEYW56	Customer Public Key Word56
0x000019E4	CUSTPUBKEYW57	Customer Public Key Word57
0x000019E8	CUSTPUBKEYW58	Customer Public Key Word58
0x000019EC	CUSTPUBKEYW59	Customer Public Key Word59
0x000019F0	CUSTPUBKEYW60	Customer Public Key Word60
0x000019F4	CUSTPUBKEYW61	Customer Public Key Word61
0x000019F8	CUSTPUBKEYW62	Customer Public Key Word62
0x000019FC	CUSTPUBKEYW63	Customer Public Key Word63
0x00001A00	CUSTOMERKEY0	128-bit customer key. Customer SW can access the protected upper half of info0 by writing this key value into the security lock register
0x00001A04	CUSTOMERKEY1	128-bit customer key.
0x00001A08	CUSTOMERKEY2	128-bit customer key.
0x00001A0C	CUSTOMERKEY3	128-bit customer key.
0x00001A10	CUSTPUBHASHW0	Customer Public Key Hash Word0
0x00001A14	CUSTPUBHASHW1	Customer Public Key Hash Word1
0x00001A18	CUSTPUBHASHW2	Customer Public Key Hash Word2
0x00001A1C	CUSTPUBHASHW3	Customer Public Key Hash Word3

23.1.2 Flash OTP INSTANCE0 INFO0 Words

23.1.2.1 SIGNATURE0 Register

INFO0 Signature

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x00000000

Word 0 (low word, bits 31:0) of the 128-bit INFO0 signature.

Table 1166: SIGNATURE0 Register

31	30	29	28	27	26	25	24	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 1167: SIGNATURE0 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG0	0xffffffff		INFO0 signature word 0 (low 32-bits).

23.1.2.2 SIGNATURE1 Register

INFO0 Signature

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x00000004

Word 1 (bits 63:32) of the 128-bit INFO0 signature.

Table 1168: SIGNATURE1 Register

31	30	29	28	27	26	25	24	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 1169: SIGNATURE1 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG1	0xffffffff		INFO0 signature word 1.

23.1.2.3 SIGNATURE2 Register

INFO0 Signature

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x00000008

Word 2 (bits 95:64) of the 128-bit INFO0 signature.

Table 1170: SIGNATURE2 Register

31	30	29	28	27	26	25	24	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 1171: SIGNATURE2 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG2	0xffffffff		INFO0 signature word 2.

23.1.2.4 SIGNATURE3 Register

INFO0 Signature

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x0000000C

Word 3 (high word, bits 127:96) of the 128-bit INFO0 signature.

Table 1172: SIGNATURE3 Register

31	30	29	28	27	26	25	24	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 1173: SIGNATURE3 Register Bits

Bit	Name	Reset	RW	Description
31:0	SIG3	0xffffffff		INFO0 signature word 3 (high 32 bits).

23.1.2.5 SECURITY Register

Security protection bits

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x00000010

This 32-bit word contains the customer programmable security.

Table 1174: SECURITY Register

31	30	29	28	27	26	25	24	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					SECPOL			KEYWRAP				RSVD	SECBOOTONRST			RSVD	SECBOOT			PLONEXIT	SDBG	BOOTLOADER_AT_RESET	EN_CUST_INFO_ERASE	EN_CUST_INFO_PROG					SECURE_LOCK	SRAM_WIPE	SWO_CTRL	DEBUG_PROT

Table 1175: SECURITY Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x3f		Reserved.
26:24	SECPOL	0x7		Defines the minimum security level required DIS = 0x0 - No policy enforced AUTH = 0x1 - If bit[0] is set, Authentication is required ENC = 0x2 - If bit[1] is set, Encryption is required ARB = 0x4 - If bit[2] is set, Anti-Rollback is required (future support)
23:20	KEYWRAP	0xf		Key wrap method used to validate customer program image. (used by boot-loader SW) NOWRAP = 0x0 - No key wrap XORWRAP = 0x1 - XOR based key wrap AES128WRAP = 0x2 - AES-128 based key wrap
19	RSVD	0x1		Reserved.
18:16	SEC-BOOTONRST	0x7		Enable secure boot at warm reset. All other encodings not listed will result in an error. SBOREN = 0x2 - Secure boot on reset enable SBORDIS = 0x5 - Secure boot on reset disabled
15	RSVD	0x1		Reserved.
14:12	SECBOOT	0x7		Enable secure boot. All other encodings not listed will result in an error. SBEN = 0x2 - Secure boot enable SBDIS = 0x5 - Secure boot disabled
11	PLONEXIT	0x1		Flash Protection Lock on bootloader exit. (used by bootloader SW) PLNS = 0x0 - Protection lock will remain not set, allowing customer firmware to set. PLS = 0x1 - Flash Protection lock will be set before handoff to customer firmware.

Table 1175: SECURITY Register Bits

Bit	Name	Reset	RW	Description
10	SDBG	0x1		Secure Debug Lock. Should be set to 0 for production parts. (used by boot-loader SW) LOCK = 0x0 - Prevents debugger control while PROTLOCK is not set. UNLOCK = 0x1 - Debugger is allowed to connect during secure boot stage.
9	BOOTLOAD-ER_AT_RESET	0x1		Enable bootloader action at reset. BARDS = 0x0 - bootloader can go to deep sleep. BARNODS = 0x1 - bootloader spins in an infinite while loop
8	EN_CUST_IN-FO_ERASE	0x1		Enable customer INFO space erasing.
7:4	EN_CUST_IN-FO_PROG	0xf		Enable customer INFO space programming.
3	SECURE_LOCK	0x1		Locks the device preventing any further updates or overrides to locked parti-tions based on EN_CUST_INFO_ERASE, EN_CUST_INFO_PROG, WRITE_PROTECT* and COPY_PROTECT*.
2	SRAM_WIPE	0x1		SRAM wipe.
1	SWO_CTRL	0x1		SWO Control.
0	DEBUG_PROT	0x1		Debugger protection.

23.1.2.6 CUSTOMERTRIM Register

Customer trim values

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x00000014

Customer Programmable trim overrides. Bits in this register are loaded into hardware registers at reset.

Table 1176: CUSTOMERTRIM Register

31	30	29	28	27	26	25	24	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																													BLE_FEATURE_enable	BLE_BUCK_enable	SIMO_BUCK_enable

Table 1177: CUSTOMERTRIM Register Bits

Bit	Name	Reset	RW	Description
31:3	RSVD	0x3fffffff		Reserved
2	BLE_FEATURE_enable	0x1		BLE Feature Enable Bit
1	BLE_BUCK_enable	0x1		BLE Buck Enable Bit
0	SIMO_BUCK_enable	0x1		SIMO Buck Enable Bit

23.1.2.7 CUSTOMERTRIM2 Register

Customer trim values word2

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x00000018

Customer Programmable trim overrides. Bits in this register are used by software for hardware configuration.

Table 1178: CUSTOMERTRIM2 Register

31	30	29	28	27	26	25	24	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																					XO32M_FREQ_TRIM										

Table 1179: CUSTOMERTRIM2 Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x1fffff		Reserved
10:0	XO32M_FREQ_TRIM	0x7ff		XO32 frequency trim. This field can optionally be adjusted to provide better interoperability performance based on crystal and board design.

23.1.2.8 SECURITYOVR Register

Security Override configuration bits

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x00000020

This 32-bit word contains the override configuration for forcing GPIO-based firmware update.

Table 1180: SECURITYOVR Register

31	30	29	28	27	26	25	24	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																								POL	GPIO						

Table 1181: SECURITYOVR Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0xffffff		Reserved
7	POL	0x1		GPIO polarity to indicate update. POL_HIGH = 0x1 - Polarity set to High or Logic 1 POL_LOW = 0x0 - Polarity set to Low or Logic 0
6:0	GPIO	0x7f		GPIO port to be used to indicate forced update. A value of 0x7F disables this feature.

23.1.2.9 SECURITYWIREDCFG Register

Security Wired configuration bits

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x00000024

This 32-bit word contains the configuration for the wired update interface.

Table 1182: SECURITYWIREDCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TIMEOUT																I2CADDR								SLVINTPIN						IFC	

Table 1183: SECURITYWIREDCFG Register Bits

Bit	Name	Reset	RW	Description
31:16	TIMEOUT	0xffff		Timeout for wired interface poll (in milliseconds)
15:9	I2CADDR	0x7f		I2C Address
8:3	SLVINTPIN	0x3f		Slave interrupt pin

Table 1183: SECURITYWIREDCFG Register Bits

Bit	Name	Reset	RW	Description
2:0	IFC	0x7		Wired interface configuration IFC_UART = 0x1 - UART interface IFC_SPI = 0x2 - SPI interface IFC_I2C = 0x4 - I2C interface

23.1.2.10SECURITYWIREDIFCCFG0 Register

Security Wired Interface configuration word0

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x00000028

This 32-bit word contains the interface configuration word0 for the UART wired update.

Table 1184: SECURITYWIREDIFCCFG0 Register

31	30	29	28	27	26	25	24	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				BAUDRATE																				DATALEN		2STOP	EVEN	PAR	CTS	RTS	UART

Table 1185: SECURITYWIREDIFCCFG0 Register Bits

Bit	Name	Reset	RW	Description
31:28	RSVD	0xf		Reserved
27:8	BAUDRATE	0xfffff		UART Baudrate
7:6	DATALEN	0x3		Number of Data Bits 5BIT = 0x0 - 5 bit 6BIT = 0x1 - 6 bit 7BIT = 0x2 - 7 bit 8BIT = 0x3 - 8 bit
5	2STOP	0x1		2 Stop Bits
4	EVEN	0x1		Even Parity
3	PAR	0x1		Enable Parity
2	CTS	0x1		Enable CTS
1	RTS	0x1		Enable RTS

Table 1185: SECURITYWIREDIFCCFG0 Register Bits

Bit	Name	Reset	RW	Description
0	UART	0x1		UART Module

23.1.2.11SECURITYWIREDIFCCFG1 Register

Security Wired Interface configuration word1

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x0000002C

This 32-bit word contains the interface configuration word1 for the UART wired update.

Table 1186: SECURITYWIREDIFCCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PIN3								PIN2								PIN1								PIN0							

Table 1187: SECURITYWIREDIFCCFG1 Register Bits

Bit	Name	Reset	RW	Description
31:24	PIN3	0xff		Pin 3 for UART interface
23:16	PIN2	0xff		Pin 2 for UART interface
15:8	PIN1	0xff		Pin 1 for UART interface
7:0	PIN0	0xff		Pin 0 for UART interface

23.1.2.12SECURITYWIREDIFCCFG2 Register

Security Wired Interface configuration word2

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x00000030

This 32-bit word contains the raw Pin configuration for the UART wired interface pin 0.

Table 1188: SECURITYWIREDIFCCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD2								ALTPADCFG								RSVD1				GPIOCFG				PADCFG							

Table 1189: SECURITYWIREDIFCCFG2 Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD2	0xff		Reserved
23:16	ALTPADCFG	0xff		8 bit value representing the raw ALTPADCFG bits for this pin
15:12	RSVD1	0xf		Reserved
11:8	GPIOCFG	0xf		4 bit value representing the raw GPIOCFG bits for this pin
7:0	PADCFG	0xff		8 bit value representing the raw PADREG bits for this pin

23.1.2.13SECURITYWIREDIFCCFG3 Register

Security Wired Interface configuration word3

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x00000034

This 32-bit word contains the raw Pin configuration for the UART wired interface pin 1.

Table 1190: SECURITYWIREDIFCCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD2								ALTPADCFG								RSVD1				GPIOCFG				PADCFG							

Table 1191: SECURITYWIREDIFCCFG3 Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD2	0xff		Reserved
23:16	ALTPADCFG	0xff		8 bit value representing the raw ALTPADCFG bits for this pin
15:12	RSVD1	0xf		Reserved
11:8	GPIOCFG	0xf		4 bit value representing the raw GPIOCFG bits for this pin
7:0	PADCFG	0xff		8 bit value representing the raw PADREG bits for this pin

23.1.2.14SECURITYWIREDIFCCFG4 Register

Security Wired Interface configuration word4

OFFSET: 0x00000038

INSTANCE 0 ADDRESS: 0x00000038

This 32-bit word contains the raw Pin configuration for the UART wired interface pin 2.

Table 1192: SECURITYWIREDIFCCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD2								ALTPADCFG								RSVD1				GPIOCFG				PADCFG							

Table 1193: SECURITYWIREDIFCCFG4 Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD2	0xff		Reserved
23:16	ALTPADCFG	0xff		8 bit value representing the raw ALTPADCFG bits for this pin
15:12	RSVD1	0xf		Reserved
11:8	GPIOCFG	0xf		4 bit value representing the raw GPIOCFG bits for this pin
7:0	PADCFG	0xff		8 bit value representing the raw PADREG bits for this pin

23.1.2.15SECURITYWIREDIFCCFG5 Register

Security Wired Interface configuration word5

OFFSET: 0x0000003C

INSTANCE 0 ADDRESS: 0x0000003C

This 32-bit word contains the raw Pin configuration for the UART wired interface pin 3.

Table 1194: SECURITYWIREDIFCCFG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD2								ALTPADCFG								RSVD1				GPIOCFG				PADCFG							

Table 1195: SECURITYWIREDIFCCFG5 Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD2	0xff		Reserved
23:16	ALTPADCFG	0xff		8 bit value representing the raw ALTPADCFG bits for this pin

Table 1195: SECURITYWIREDIFCCFG5 Register Bits

Bit	Name	Reset	RW	Description
15:12	RSVD1	0xf		Reserved
11:8	GPIOCFG	0xf		4 bit value representing the raw GPIOCFG bits for this pin
7:0	PADCFG	0xff		8 bit value representing the raw PADREG bits for this pin

23.1.2.16SECURITYVERSION Register

Security version field

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x00000040

This 32-bit word contains the version ID used for revision control

Table 1196: SECURITYVERSION Register

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

Table 1197: SECURITYVERSION Register Bits

Bit	Name	Reset	RW	Description
31:0	VERSION	0xffffffff		Version ID

23.1.2.17SECURITYSRAMRESV Register

SRAM Reserved for Application Scratch space

OFFSET: 0x00000050

INSTANCE 0 ADDRESS: 0x00000050

This 32-bit word indicates the amount of SRAM to keep reserved for application scratch space. This reserves the specified memory at the top end of SRAM memory address range. This memory is not disturbed by the Secure Boot Loader

Table 1198: SECURITYSRAMRESV Register

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

Table 1199: SECURITYSRAMRESV Register Bits

Bit	Name	Reset	RW	Description
31:0	SRAM_RESV	0xffffffff		SRAM Reservation

23.1.2.18WRITEPROTECTL Register

Flash write-protection bits.

OFFSET: 0x000001F8

INSTANCE 0 ADDRESS: 0x000001F8

These bits write-protect flash in 16KB chunks.

Table 1200: WRITEPROTECTL Register

31	30	29	28	27	26	25	24	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 1201: WRITEPROTECTL 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.

23.1.2.19WRITEPROTECTH Register

Flash write-protection bits.

OFFSET: 0x000001FC

INSTANCE 0 ADDRESS: 0x000001FC

These bits write-protect flash in 16KB chunks.

Table 1202: WRITEPROTECTH Register

31	30	29	28	27	26	25	24	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 1203: WRITEPROTECTH 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.

23.1.2.20COPYPROTECTL Register

Flash copy/read-protection bits.

OFFSET: 0x00000200

INSTANCE 0 ADDRESS: 0x00000200

These bits read-protect flash in 16KB chunks.

Table 1204: COPYPROTECTL Register

31	30	29	28	27	26	25	24	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 1205: COPYPROTECTL 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.

23.1.2.21COPYPROTECTH Register

Flash copy/read-protection bits.

OFFSET: 0x00000204

INSTANCE 0 ADDRESS: 0x00000204

These bits read-protect flash in 16KB chunks.

Table 1206: COPYPROTECTH Register

31	30	29	28	27	26	25	24	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 1207: COPYPROTECTH 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.

23.1.2.22WRITEPROTECTSBLL Register

Flash write-protection bits.

OFFSET: 0x000009F8

INSTANCE 0 ADDRESS: 0x000009F8

These bits write-protect flash in 16KB chunks. Only SBL can override these through Secure OTA

Table 1208: WRITEPROTECTSBLL Register

31	30	29	28	27	26	25	24	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 1209: WRITEPROTECTSBLL 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.

23.1.2.23WRITEPROTECTSBLH Register

Flash write-protection bits.

OFFSET: 0x000009FC

INSTANCE 0 ADDRESS: 0x000009FC

These bits write-protect flash in 16KB chunks. Only SBL can override these through Secure OTA

Table 1210: WRITEPROTECTSBLH Register

31	30	29	28	27	26	25	24	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 1211: WRITEPROTECTSBLH 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.

23.1.2.24COPYPROTECTSBLL Register

Flash copy/read-protection bits.

OFFSET: 0x00000A00

INSTANCE 0 ADDRESS: 0x00000A00

These bits read-protect flash in 16KB chunks.

Table 1212: COPYPROTECTSBLL Register

31	30	29	28	27	26	25	24	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 1213: COPYPROTECTSBLL 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.

23.1.2.25COPYPROTECTSBLH Register

Flash copy/read-protection bits.

OFFSET: 0x00000A04

INSTANCE 0 ADDRESS: 0x00000A04

These bits read-protect flash in 16KB chunks. Only SBL can override these through Secure OTA

Table 1214: COPYPROTECTSBLH Register

31	30	29	28	27	26	25	24	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 1215: COPYPROTECTSBLH 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.

23.1.2.26MAINPTR0 Register

main firmware pointer 0

OFFSET: 0x00000C00

INSTANCE 0 ADDRESS: 0x00000C00

This is the main/sbl_main firmware pointer 0 referenced by sbl_init for pointing to the main or sbl_main base address

Table 1216: MAINPTR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PTR0																															

Table 1217: MAINPTR0 Register Bits

Bit	Name	Reset	RW	Description
31:0	PTR0	0xffffffff		main pointer 0

23.1.2.27MAINPTR1 Register

main firmware pointer 1

OFFSET: 0x00000C04

INSTANCE 0 ADDRESS: 0x00000C04

This is the main/sbl_main firmware pointer 1 referenced by sbl_init for pointing to the main or sbl_main base address

Table 1218: MAINPTR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PTR1																															

Table 1219: MAINPTR1 Register Bits

Bit	Name	Reset	RW	Description
31:0	PTR1	0xffffffff		main pointer 1

23.1.2.28KREVTRACK Register

KEK Revocation Tracker

OFFSET: 0x00000C08

INSTANCE 0 ADDRESS: 0x00000C08

KEK Key Revocation Tracker. Monotonic counter where each bit indicates if that respective word in the KEK key bank is valid. For example, if KEK0 is not valid but KEK1-7 are valid (for a 128-bit KEK configuration), the KREVTRACK would be 0xFFFFFFF0.

Table 1220: KREVTRACK Register

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

Table 1221: KREVTRACK Register Bits

Bit	Name	Reset	RW	Description
31:0	KTRCKER	0xffffffff		KEK Revocation Tracker

23.1.2.29AREVTRACK Register

AUTH Revocation Tracker

OFFSET: 0x00000C0C

INSTANCE 0 ADDRESS: 0x00000C0C

AUTH Key Revocation Tracker. Monotonic counter where each bit indicates if that respective word in the AUTH key bank is valid. For example, if AUTH0 is not valid but AUTH1-7 are valid (for a 128-bit AUTH configuration), the AREVTRACK would be 0xFFFFFFF0.

Table 1222: AREVTRACK Register

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

Table 1223: AREVTRACK Register Bits

Bit	Name	Reset	RW	Description
31:0	ATRCKER	0xffffffff		AUTH Revocation Tracker

23.1.2.30OTADESCRIPTOR Register

OTA Descriptor Pointer

OFFSET: 0x00000C10

INSTANCE 0 ADDRESS: 0x00000C10

This field is used to track the OTA DESCRIPTOR pointer to ensure proper OTA update.

Table 1224: OTADESCRIPTOR Register

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

Table 1225: OTADESCRIPTOR Register Bits

Bit	Name	Reset	RW	Description
31:0	DESCRIPTOR	0xffffffff		OTA Descriptor Pointer

23.1.2.31MAINCNT0 Register

main Index Counter 0

OFFSET: 0x00000FF8

INSTANCE 0 ADDRESS: 0x00000FF8

Index counter for main or sbl_main firmware. Counter is used to indicate which pointer to reference, MAINPTR1 or MAINPTR2

Table 1226: MAINCNT0 Register

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

Table 1227: MAINCNT0 Register Bits

Bit	Name	Reset	RW	Description
31:0	INDXCNTR	0xffffffff		main Index Counter

23.1.2.32MAINCNT1 Register

main Index Counter 1

OFFSET: 0x00000FFC

INSTANCE 0 ADDRESS: 0x00000FFC

Index counter for main or sbl_main firmware. Counter is used to indicate which pointer to reference, MAINPTR1 or MAINPTR2

Table 1228: MAINCNT1 Register

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

Table 1229: MAINCNT1 Register Bits

Bit	Name	Reset	RW	Description
31:0	INDXCNTR	0xffffffff		main Index Counter

23.1.2.33CUSTKEKW0 Register

Customer KEK Word0

OFFSET: 0x00001800

INSTANCE 0 ADDRESS: 0x00001800

This is the Customer KEK Word0.

Table 1230: CUSTKEKW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W0																															

Table 1231: CUSTKEKW0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W0	0xffffffff		Customer KEK Word0

23.1.2.34CUSTKEKW1 Register

Customer KEK Word1

OFFSET: 0x00001804

INSTANCE 0 ADDRESS: 0x00001804

This is the Customer KEK Word1.

Table 1232: CUSTKEKW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W1																															

Table 1233: CUSTKEKW1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W1	0xffffffff		Customer KEK Word1

23.1.2.35CUSTKEKW2 Register

Customer KEK Word2

OFFSET: 0x00001808

INSTANCE 0 ADDRESS: 0x00001808

This is the Customer KEK Word2.

Table 1234: CUSTKEKW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W2																															

Table 1235: CUSTKEKW2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W2	0xffffffff		Customer KEK Word2

23.1.2.36CUSTKEKW3 Register

Customer KEK Word3

OFFSET: 0x0000180C

INSTANCE 0 ADDRESS: 0x0000180C

This is the Customer KEK Word3.

Table 1236: CUSTKEKW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W3																															

Table 1237: CUSTKEKW3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W3	0xffffffff		Customer KEK Word3

23.1.2.37CUSTKEKW4 Register

Customer KEK Word4

OFFSET: 0x00001810

INSTANCE 0 ADDRESS: 0x00001810

This is the Customer KEK Word4.

Table 1238: CUSTKEKW4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W4																															

Table 1239: CUSTKEKW4 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W4	0xffffffff		Customer KEK Word4

23.1.2.38CUSTKEKW5 Register

Customer KEK Word5

OFFSET: 0x00001814

INSTANCE 0 ADDRESS: 0x00001814

This is the Customer KEK Word5.

Table 1240: CUSTKEKW5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W5																															

Table 1241: CUSTKEKW5 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W5	0xffffffff		Customer KEK Word5

23.1.2.39CUSTKEKW6 Register

Customer KEK Word6

OFFSET: 0x00001818

INSTANCE 0 ADDRESS: 0x00001818

This is the Customer KEK Word6.

Table 1242: CUSTKEKW6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W6																															

Table 1243: CUSTKEKW6 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W6	0xffffffff		Customer KEK Word6

23.1.2.40CUSTKEKW7 Register

Customer KEK Word7

OFFSET: 0x0000181C

INSTANCE 0 ADDRESS: 0x0000181C

This is the Customer KEK Word7.

Table 1244: CUSTKEKW7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W7																															

Table 1245: CUSTKEKW7 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W7	0xffffffff		Customer KEK Word7

23.1.2.41CUSTKEKW8 Register

Customer KEK Word0

OFFSET: 0x00001820

INSTANCE 0 ADDRESS: 0x00001820

This is the Customer KEK Word8.

Table 1246: CUSTKEKW8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W8																															

Table 1247: CUSTKEKW8 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W8	0xffffffff		Customer KEK Word8

23.1.2.42CUSTKEKW9 Register

Customer KEK Word9

OFFSET: 0x00001824

INSTANCE 0 ADDRESS: 0x00001824

This is the Customer KEK Word9.

Table 1248: CUSTKEKW9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W9																															

Table 1249: CUSTKEKW9 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W9	0xffffffff		Customer KEK Word9

23.1.2.43CUSTKEKW10 Register

Customer KEK Word10

OFFSET: 0x00001828

INSTANCE 0 ADDRESS: 0x00001828

This is the Customer KEK Word10.

Table 1250: CUSTKEKW10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W10																															

Table 1251: CUSTKEKW10 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W10	0xffffffff		Customer KEK Word10

23.1.2.44CUSTKEKW11 Register

Customer KEK Word11

OFFSET: 0x0000182C

INSTANCE 0 ADDRESS: 0x0000182C

This is the Customer KEK Word11.

Table 1252: CUSTKEKW11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W11																															

Table 1253: CUSTKEKW11 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W11	0xffffffff		Customer KEK Word11

23.1.2.45CUSTKEKW12 Register

Customer KEK Word12

OFFSET: 0x00001830

INSTANCE 0 ADDRESS: 0x00001830

This is the Customer KEK Word12.

Table 1254: CUSTKEKW12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W12																															

Table 1255: CUSTKEKW12 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W12	0xffffffff		Customer KEK Word12

23.1.2.46CUSTKEKW13 Register

Customer KEK Word13

OFFSET: 0x00001834

INSTANCE 0 ADDRESS: 0x00001834

This is the Customer KEK Word13.

Table 1256: CUSTKEKW13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W13																															

Table 1257: CUSTKEKW13 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W13	0xffffffff		Customer KEK Word13

23.1.2.47CUSTKEKW14 Register

Customer KEK Word14

OFFSET: 0x00001838

INSTANCE 0 ADDRESS: 0x00001838

This is the Customer KEK Word14.

Table 1258: CUSTKEKW14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W14																															

Table 1259: CUSTKEKW14 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W14	0xffffffff		Customer KEK Word14

23.1.2.48CUSTKEKW15 Register

Customer KEK Word15

OFFSET: 0x0000183C

INSTANCE 0 ADDRESS: 0x0000183C

This is the Customer KEK Word15.

Table 1260: CUSTKEKW15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W15																															

Table 1261: CUSTKEKW15 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W15	0xffffffff		Customer KEK Word15

23.1.2.49CUSTKEKW16 Register

Customer KEK Word16

OFFSET: 0x00001840

INSTANCE 0 ADDRESS: 0x00001840

This is the Customer KEK Word16.

Table 1262: CUSTKEKW16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W16																															

Table 1263: CUSTKEKW16 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W16	0xffffffff		Customer KEK Word16

23.1.2.50CUSTKEKW17 Register

Customer KEK Word17

OFFSET: 0x00001844

INSTANCE 0 ADDRESS: 0x00001844

This is the Customer KEK Word17.

Table 1264: CUSTKEKW17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W17																															

Table 1265: CUSTKEKW17 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W17	0xffffffff		Customer KEK Word17

23.1.2.51CUSTKEKW18 Register

Customer KEK Word18

OFFSET: 0x00001848

INSTANCE 0 ADDRESS: 0x00001848

This is the Customer KEK Word18.

Table 1266: CUSTKEKW18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W18																															

Table 1267: CUSTKEKW18 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W18	0xffffffff		Customer KEK Word18

23.1.2.52CUSTKEKW19 Register

Customer KEK Word19

OFFSET: 0x0000184C

INSTANCE 0 ADDRESS: 0x0000184C

This is the Customer KEK Word19.

Table 1268: CUSTKEKW19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W19																															

Table 1269: CUSTKEKW19 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W19	0xffffffff		Customer KEK Word19

23.1.2.53CUSTKEKW20 Register

Customer KEK Word20

OFFSET: 0x00001850

INSTANCE 0 ADDRESS: 0x00001850

This is the Customer KEK Word20.

Table 1270: CUSTKEKW20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W20																															

Table 1271: CUSTKEKW20 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W20	0xffffffff		Customer KEK Word20

23.1.2.54CUSTKEKW21 Register

Customer KEK Word21

OFFSET: 0x00001854

INSTANCE 0 ADDRESS: 0x00001854

This is the Customer KEK Word21.

Table 1272: CUSTKEKW21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W21																															

Table 1273: CUSTKEKW21 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W21	0xffffffff		Customer KEK Word21

23.1.2.55CUSTKEKW22 Register

Customer KEK Word22

OFFSET: 0x00001858

INSTANCE 0 ADDRESS: 0x00001858

This is the Customer KEK Word22.

Table 1274: CUSTKEKW22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W22																															

Table 1275: CUSTKEKW22 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W22	0xffffffff		Customer KEK Word22

23.1.2.56CUSTKEKW23 Register

Customer KEK Word23

OFFSET: 0x0000185C

INSTANCE 0 ADDRESS: 0x0000185C

This is the Customer KEK Word23.

Table 1276: CUSTKEKW23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W23																															

Table 1277: CUSTKEKW23 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W23	0xffffffff		Customer KEK Word23

23.1.2.57CUSTKEKW24 Register

Customer KEK Word24

OFFSET: 0x00001860

INSTANCE 0 ADDRESS: 0x00001860

This is the Customer KEK Word24.

Table 1278: CUSTKEKW24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W24																															

Table 1279: CUSTKEKW24 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W24	0xffffffff		Customer KEK Word24

23.1.2.58CUSTKEKW25 Register

Customer KEK Word25

OFFSET: 0x00001864

INSTANCE 0 ADDRESS: 0x00001864

This is the Customer KEK Word25.

Table 1280: CUSTKEKW25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W25																															

Table 1281: CUSTKEKW25 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W25	0xffffffff		Customer KEK Word25

23.1.2.59CUSTKEKW26 Register

Customer KEK Word26

OFFSET: 0x00001868

INSTANCE 0 ADDRESS: 0x00001868

This is the Customer KEK Word26.

Table 1282: CUSTKEKW26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W26																															

Table 1283: CUSTKEKW26 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W26	0xffffffff		Customer KEK Word26

23.1.2.60CUSTKEKW27 Register

Customer KEK Word27

OFFSET: 0x0000186C

INSTANCE 0 ADDRESS: 0x0000186C

This is the Customer KEK Word27.

Table 1284: CUSTKEKW27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W27																															

Table 1285: CUSTKEKW27 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W27	0xffffffff		Customer KEK Word27

23.1.2.61CUSTKEKW28 Register

Customer KEK Word28

OFFSET: 0x00001870

INSTANCE 0 ADDRESS: 0x00001870

This is the Customer KEK Word28.

Table 1286: CUSTKEKW28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W28																															

Table 1287: CUSTKEKW28 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W28	0xffffffff		Customer KEK Word28

23.1.2.62CUSTKEKW29 Register

Customer KEK Word29

OFFSET: 0x00001874

INSTANCE 0 ADDRESS: 0x00001874

This is the Customer KEK Word29.

Table 1288: CUSTKEKW29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W29																															

Table 1289: CUSTKEKW29 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W29	0xffffffff		Customer KEK Word29

23.1.2.63CUSTKEKW30 Register

Customer KEK Word30

OFFSET: 0x00001878

INSTANCE 0 ADDRESS: 0x00001878

This is the Customer KEK Word30.

Table 1290: CUSTKEKW30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W30																															

Table 1291: CUSTKEKW30 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W30	0xffffffff		Customer KEK Word30

23.1.2.64CUSTKEKW31 Register

Customer KEK Word31

OFFSET: 0x0000187C

INSTANCE 0 ADDRESS: 0x0000187C

This is the Customer KEK Word31.

Table 1292: CUSTKEKW31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTKEK_W31																															

Table 1293: CUSTKEKW31 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTKEK_W31	0xffffffff		Customer KEK Word31

23.1.2.65CUSTAUTHW0 Register

Customer AUTH Key Word0

OFFSET: 0x00001880

INSTANCE 0 ADDRESS: 0x00001880

This is the Customer AUTH Key Word0.

Table 1294: CUSTAUTHW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W0																															

Table 1295: CUSTAUTHW0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W0	0xffffffff		Customer AUTH Key Word0

23.1.2.66CUSTAUTHW1 Register

Customer AUTH Key Word1

OFFSET: 0x00001884

INSTANCE 0 ADDRESS: 0x00001884

This is the Customer AUTH Key Word1.

Table 1296: CUSTAUTHW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W1																															

Table 1297: CUSTAUTHW1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W1	0xffffffff		Customer AUTH Key Word1

23.1.2.67CUSTAUTHW2 Register

Customer AUTH Key Word2

OFFSET: 0x00001888

INSTANCE 0 ADDRESS: 0x00001888

This is the Customer AUTH Key Word2.

Table 1298: CUSTAUTHW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W2																															

Table 1299: CUSTAUTHW2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W2	0xffffffff		Customer AUTH Key Word2

23.1.2.68CUSTAUTHW3 Register

Customer AUTH Key Word3

OFFSET: 0x0000188C

INSTANCE 0 ADDRESS: 0x0000188C

This is the Customer AUTH Key Word3.

Table 1300: CUSTAUTHW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W3																															

Table 1301: CUSTAUTHW3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W3	0xffffffff		Customer AUTH Key Word3

23.1.2.69CUSTAUTHW4 Register

Customer AUTH Key Word4

OFFSET: 0x00001890

INSTANCE 0 ADDRESS: 0x00001890

This is the Customer AUTH Key Word4.

Table 1302: CUSTAUTHW4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W4																															

Table 1303: CUSTAUTHW4 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W4	0xffffffff		Customer AUTH Key Word4

23.1.2.70CUSTAUTHW5 Register

Customer AUTH Key Word5

OFFSET: 0x00001894

INSTANCE 0 ADDRESS: 0x00001894

This is the Customer AUTH Key Word5.

Table 1304: CUSTAUTHW5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W5																															

Table 1305: CUSTAUTHW5 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W5	0xffffffff		Customer AUTH Key Word5

23.1.2.71CUSTAUTHW6 Register

Customer AUTH Key Word6

OFFSET: 0x00001898

INSTANCE 0 ADDRESS: 0x00001898

This is the Customer AUTH Key Word6.

Table 1306: CUSTAUTHW6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W6																															

Table 1307: CUSTAUTHW6 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W6	0xffffffff		Customer AUTH Key Word6

23.1.2.72CUSTAUTHW7 Register

Customer AUTH Key Word7

OFFSET: 0x0000189C

INSTANCE 0 ADDRESS: 0x0000189C

This is the Customer AUTH Key Word7.

Table 1308: CUSTAUTHW7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W7																															

Table 1309: CUSTAUTHW7 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W7	0xffffffff		Customer AUTH Key Word7

23.1.2.73CUSTAUTHW8 Register

Customer AUTH Key Word0

OFFSET: 0x000018A0

INSTANCE 0 ADDRESS: 0x000018A0

This is the Customer AUTH Key Word8.

Table 1310: CUSTAUTHW8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W8																															

Table 1311: CUSTAUTHW8 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W8	0xffffffff		Customer AUTH Key Word8

23.1.2.74CUSTAUTHW9 Register

Customer AUTH Key Word9

OFFSET: 0x000018A4

INSTANCE 0 ADDRESS: 0x000018A4

This is the Customer AUTH Key Word9.

Table 1312: CUSTAUTHW9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W9																															

Table 1313: CUSTAUTHW9 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAUTH_W9	0xffffffff		Customer AUTH Key Word9

23.1.2.75CUSTAUTHW10 Register

Customer AUTH Key Word10

OFFSET: 0x000018A8

INSTANCE 0 ADDRESS: 0x000018A8

This is the Customer AUTH Key Word10.

Table 1314: CUSTAUTHW10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W10																															

Table 1315: CUSTAUTHW10 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W10	0xffffffff		Customer AUTH Key Word10

23.1.2.76CUSTAUTHW11 Register

Customer AUTH Key Word11

OFFSET: 0x000018AC

INSTANCE 0 ADDRESS: 0x000018AC

This is the Customer AUTH Key Word11.

Table 1316: CUSTAUTHW11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W11																															

Table 1317: CUSTAUTHW11 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W11	0xffffffff		Customer AUTH Key Word11

23.1.2.77CUSTAUTHW12 Register

Customer AUTH Key Word12

OFFSET: 0x000018B0

INSTANCE 0 ADDRESS: 0x000018B0

This is the Customer AUTH Key Word12.

Table 1318: CUSTAUTHW12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W12																															

Table 1319: CUSTAUTHW12 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W12	0xffffffff		Customer AUTH Key Word12

23.1.2.78CUSTAUTHW13 Register

Customer AUTH Key Word13

OFFSET: 0x000018B4

INSTANCE 0 ADDRESS: 0x000018B4

This is the Customer AUTH Key Word13.

Table 1320: CUSTAUTHW13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W13																															

Table 1321: CUSTAUTHW13 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W13	0xffffffff		Customer AUTH Key Word13

23.1.2.79CUSTAUTHW14 Register

Customer AUTH Key Word14

OFFSET: 0x000018B8

INSTANCE 0 ADDRESS: 0x000018B8

This is the Customer AUTH Key Word14.

Table 1322: CUSTAUTHW14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W14																															

Table 1323: CUSTAUTHW14 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W14	0xffffffff		Customer AUTH Key Word14

23.1.2.80CUSTAUTHW15 Register

Customer AUTH Key Word15

OFFSET: 0x000018BC

INSTANCE 0 ADDRESS: 0x000018BC

This is the Customer AUTH Key Word15.

Table 1324: CUSTAUTHW15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W15																															

Table 1325: CUSTAUTHW15 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W15	0xffffffff		Customer AUTH Key Word15

23.1.2.81CUSTAUTHW16 Register

Customer AUTH Key Word16

OFFSET: 0x000018C0

INSTANCE 0 ADDRESS: 0x000018C0

This is the Customer AUTH Key Word16.

Table 1326: CUSTAUTHW16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W16																															

Table 1327: CUSTAUTHW16 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W16	0xffffffff		Customer AUTH Key Word16

23.1.2.82CUSTAUTHW17 Register

Customer AUTH Key Word17

OFFSET: 0x000018C4

INSTANCE 0 ADDRESS: 0x000018C4

This is the Customer AUTH Key Word17.

Table 1328: CUSTAUTHW17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W17																															

Table 1329: CUSTAUTHW17 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W17	0xffffffff		Customer AUTH Key Word17

23.1.2.83CUSTAUTHW18 Register

Customer AUTH Key Word18

OFFSET: 0x000018C8

INSTANCE 0 ADDRESS: 0x000018C8

This is the Customer AUTH Key Word18.

Table 1330: CUSTAUTHW18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W18																															

Table 1331: CUSTAUTHW18 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W18	0xffffffff		Customer AUTH Key Word18

23.1.2.84CUSTAUTHW19 Register

Customer AUTH Key Word19

OFFSET: 0x000018CC

INSTANCE 0 ADDRESS: 0x000018CC

This is the Customer AUTH Key Word19.

Table 1332: CUSTAUTHW19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W19																															

Table 1333: CUSTAUTHW19 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W19	0xffffffff		Customer AUTH Key Word19

23.1.2.85CUSTAUTHW20 Register

Customer AUTH Key Word20

OFFSET: 0x000018D0

INSTANCE 0 ADDRESS: 0x000018D0

This is the Customer AUTH Key Word20.

Table 1334: CUSTAUTHW20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W20																															

Table 1335: CUSTAUTHW20 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W20	0xffffffff		Customer AUTH Key Word20

23.1.2.86CUSTAUTHW21 Register

Customer AUTH Key Word21

OFFSET: 0x000018D4

INSTANCE 0 ADDRESS: 0x000018D4

This is the Customer AUTH Key Word21.

Table 1336: CUSTAUTHW21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W21																															

Table 1337: CUSTAUTHW21 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W21	0xffffffff		Customer AUTH Key Word21

23.1.2.87CUSTAUTHW22 Register

Customer AUTH Key Word22

OFFSET: 0x000018D8

INSTANCE 0 ADDRESS: 0x000018D8

This is the Customer AUTH Key Word22.

Table 1338: CUSTAUTHW22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W22																															

Table 1339: CUSTAUTHW22 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W22	0xffffffff		Customer AUTH Key Word22

23.1.2.88CUSTAUTHW23 Register

Customer AUTH Key Word23

OFFSET: 0x000018DC

INSTANCE 0 ADDRESS: 0x000018DC

This is the Customer AUTH Key Word23.

Table 1340: CUSTAUTHW23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W23																															

Table 1341: CUSTAUTHW23 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W23	0xffffffff		Customer AUTH Key Word23

23.1.2.89CUSTAUTHW24 Register

Customer AUTH Key Word24

OFFSET: 0x000018E0

INSTANCE 0 ADDRESS: 0x000018E0

This is the Customer AUTH Key Word24.

Table 1342: CUSTAUTHW24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W24																															

Table 1343: CUSTAUTHW24 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W24	0xffffffff		Customer AUTH Key Word24

23.1.2.90CUSTAUTHW25 Register

Customer AUTH Key Word25

OFFSET: 0x000018E4

INSTANCE 0 ADDRESS: 0x000018E4

This is the Customer AUTH Key Word25.

Table 1344: CUSTAUTHW25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W25																															

Table 1345: CUSTAUTHW25 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W25	0xffffffff		Customer AUTH Key Word25

23.1.2.91CUSTAUTHW26 Register

Customer AUTH Key Word26

OFFSET: 0x000018E8

INSTANCE 0 ADDRESS: 0x000018E8

This is the Customer AUTH Key Word26.

Table 1346: CUSTAUTHW26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W26																															

Table 1347: CUSTAUTHW26 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W26	0xffffffff		Customer AUTH Key Word26

23.1.2.92CUSTAUTHW27 Register

Customer AUTH Key Word27

OFFSET: 0x000018EC

INSTANCE 0 ADDRESS: 0x000018EC

This is the Customer AUTH Key Word27.

Table 1348: CUSTAUTHW27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W27																															

Table 1349: CUSTAUTHW27 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W27	0xffffffff		Customer AUTH Key Word27

23.1.2.93CUSTAUTHW28 Register

Customer AUTH Key Word28

OFFSET: 0x000018F0

INSTANCE 0 ADDRESS: 0x000018F0

This is the Customer AUTH Key Word28.

Table 1350: CUSTAUTHW28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W28																															

Table 1351: CUSTAUTHW28 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W28	0xffffffff		Customer AUTH Key Word28

23.1.2.94CUSTAUTHW29 Register

Customer AUTH Key Word29

OFFSET: 0x000018F4

INSTANCE 0 ADDRESS: 0x000018F4

This is the Customer AUTH Key Word29.

Table 1352: CUSTAUTHW29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W29																															

Table 1353: CUSTAUTHW29 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W29	0xffffffff		Customer AUTH Key Word29

23.1.2.95CUSTAUTHW30 Register

Customer AUTH Key Word30

OFFSET: 0x000018F8

INSTANCE 0 ADDRESS: 0x000018F8

This is the Customer AUTH Key Word30.

Table 1354: CUSTAUTHW30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W30																															

Table 1355: CUSTAUTHW30 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W30	0xffffffff		Customer AUTH Key Word30

23.1.2.96CUSTAUTHW31 Register

Customer AUTH Key Word31

OFFSET: 0x000018FC

INSTANCE 0 ADDRESS: 0x000018FC

This is the Customer AUTH Key Word31.

Table 1356: CUSTAUTHW31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTAUTH_W31																															

Table 1357: CUSTAUTHW31 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTAU-TH_W31	0xffffffff		Customer AUTH Key Word31

23.1.2.97CUSTPUBKEYW0 Register

Customer Public Key Word0

OFFSET: 0x00001900

INSTANCE 0 ADDRESS: 0x00001900

This is the Customer Public Key Word0.

Table 1358: CUSTPUBKEYW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W0																															

Table 1359: CUSTPUBKEYW0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W0	0xffffffff		Customer Public Key Word0

23.1.2.98CUSTPUBKEYW1 Register

Customer Public Key Word1

OFFSET: 0x00001904

INSTANCE 0 ADDRESS: 0x00001904

This is the Customer Public Key Word1.

Table 1360: CUSTPUBKEYW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W1																															

Table 1361: CUSTPUBKEYW1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W1	0xffffffff		Customer Public Key Word1

23.1.2.99CUSTPUBKEYW2 Register

Customer Public Key Word2

OFFSET: 0x00001908

INSTANCE 0 ADDRESS: 0x00001908

This is the Customer Public Key Word2.

Table 1362: CUSTPUBKEYW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W2																															

Table 1363: CUSTPUBKEYW2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W2	0xffffffff		Customer Public Key Word2

23.1.2.100CUSTPUBKEYW3 Register

Customer Public Key Word3

OFFSET: 0x0000190C

INSTANCE 0 ADDRESS: 0x0000190C

This is the Customer Public Key Word3.

Table 1364: CUSTPUBKEYW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W3																															

Table 1365: CUSTPUBKEYW3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W3	0xffffffff		Customer Public Key Word3

23.1.2.101CUSTPUBKEYW4 Register

Customer Public Key Word4

OFFSET: 0x00001910

INSTANCE 0 ADDRESS: 0x00001910

This is the Customer Public Key Word4.

Table 1366: CUSTPUBKEYW4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W4																															

Table 1367: CUSTPUBKEYW4 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W4	0xffffffff		Customer Public Key Word4

23.1.2.102CUSTPUBKEYW5 Register

Customer Public Key Word5

OFFSET: 0x00001914

INSTANCE 0 ADDRESS: 0x00001914

This is the Customer Public Key Word5.

Table 1368: CUSTPUBKEYW5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W5																															

Table 1369: CUSTPUBKEYW5 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W5	0xffffffff		Customer Public Key Word5

23.1.2.103CUSTPUBKEYW6 Register

Customer Public Key Word6

OFFSET: 0x00001918

INSTANCE 0 ADDRESS: 0x00001918

This is the Customer Public Key Word6.

Table 1370: CUSTPUBKEYW6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W6																															

Table 1371: CUSTPUBKEYW6 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W6	0xffffffff		Customer Public Key Word6

23.1.2.104CUSTPUBKEYW7 Register

Customer Public Key Word7

OFFSET: 0x0000191C

INSTANCE 0 ADDRESS: 0x0000191C

This is the Customer Public Key Word7.

Table 1372: CUSTPUBKEYW7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W7																															

Table 1373: CUSTPUBKEYW7 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W7	0xffffffff		Customer Public Key Word7

23.1.2.105CUSTPUBKEYW8 Register

Customer Public Key Word0

OFFSET: 0x00001920

INSTANCE 0 ADDRESS: 0x00001920

This is the Customer Public Key Word8.

Table 1374: CUSTPUBKEYW8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W8																															

Table 1375: CUSTPUBKEYW8 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W8	0xffffffff		Customer Public Key Word8

23.1.2.106CUSTPUBKEYW9 Register

Customer Public Key Word9

OFFSET: 0x00001924

INSTANCE 0 ADDRESS: 0x00001924

This is the Customer Public Key Word9.

Table 1376: CUSTPUBKEYW9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W9																															

Table 1377: CUSTPUBKEYW9 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W9	0xffffffff		Customer Public Key Word9

23.1.2.107CUSTPUBKEYW10 Register

Customer Public Key Word10

OFFSET: 0x00001928

INSTANCE 0 ADDRESS: 0x00001928

This is the Customer Public Key Word10.

Table 1378: CUSTPUBKEYW10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W10																															

Table 1379: CUSTPUBKEYW10 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W10	0xffffffff		Customer Public Key Word10

23.1.2.108CUSTPUBKEYW11 Register

Customer Public Key Word11

OFFSET: 0x0000192C

INSTANCE 0 ADDRESS: 0x0000192C

This is the Customer Public Key Word11.

Table 1380: CUSTPUBKEYW11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W11																															

Table 1381: CUSTPUBKEYW11 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W11	0xffffffff		Customer Public Key Word11

23.1.2.109CUSTPUBKEYW12 Register

Customer Public Key Word12

OFFSET: 0x00001930

INSTANCE 0 ADDRESS: 0x00001930

This is the Customer Public Key Word12.

Table 1382: CUSTPUBKEYW12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W12																															

Table 1383: CUSTPUBKEYW12 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W12	0xffffffff		Customer Public Key Word12

23.1.2.110CUSTPUBKEYW13 Register

Customer Public Key Word13

OFFSET: 0x00001934

INSTANCE 0 ADDRESS: 0x00001934

This is the Customer Public Key Word13.

Table 1384: CUSTPUBKEYW13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W13																															

Table 1385: CUSTPUBKEYW13 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W13	0xffffffff		Customer Public Key Word13

23.1.2.111CUSTPUBKEYW14 Register

Customer Public Key Word14

OFFSET: 0x00001938

INSTANCE 0 ADDRESS: 0x00001938

This is the Customer Public Key Word14.

Table 1386: CUSTPUBKEYW14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W14																															

Table 1387: CUSTPUBKEYW14 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W14	0xffffffff		Customer Public Key Word14

23.1.2.112CUSTPUBKEYW15 Register

Customer Public Key Word15

OFFSET: 0x0000193C

INSTANCE 0 ADDRESS: 0x0000193C

This is the Customer Public Key Word15.

Table 1388: CUSTPUBKEYW15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W15																															

Table 1389: CUSTPUBKEYW15 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W15	0xffffffff		Customer Public Key Word15

23.1.2.113CUSTPUBKEYW16 Register

Customer Public Key Word16

OFFSET: 0x00001940

INSTANCE 0 ADDRESS: 0x00001940

This is the Customer Public Key Word16.

Table 1390: CUSTPUBKEYW16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W16																															

Table 1391: CUSTPUBKEYW16 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W16	0xffffffff		Customer Public Key Word16

23.1.2.114CUSTPUBKEYW17 Register

Customer Public Key Word17

OFFSET: 0x00001944

INSTANCE 0 ADDRESS: 0x00001944

This is the Customer Public Key Word17.

Table 1392: CUSTPUBKEYW17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W17																															

Table 1393: CUSTPUBKEYW17 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W17	0xffffffff		Customer Public Key Word17

23.1.2.115CUSTPUBKEYW18 Register

Customer Public Key Word18

OFFSET: 0x00001948

INSTANCE 0 ADDRESS: 0x00001948

This is the Customer Public Key Word18.

Table 1394: CUSTPUBKEYW18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W18																															

Table 1395: CUSTPUBKEYW18 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W18	0xffffffff		Customer Public Key Word18

23.1.2.116CUSTPUBKEYW19 Register

Customer Public Key Word19

OFFSET: 0x0000194C

INSTANCE 0 ADDRESS: 0x0000194C

This is the Customer Public Key Word19.

Table 1396: CUSTPUBKEYW19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W19																															

Table 1397: CUSTPUBKEYW19 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W19	0xffffffff		Customer Public Key Word19

23.1.2.117CUSTPUBKEYW20 Register

Customer Public Key Word20

OFFSET: 0x00001950

INSTANCE 0 ADDRESS: 0x00001950

This is the Customer Public Key Word20.

Table 1398: CUSTPUBKEYW20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W20																															

Table 1399: CUSTPUBKEYW20 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W20	0xffffffff		Customer Public Key Word20

23.1.2.118CUSTPUBKEYW21 Register

Customer Public Key Word21

OFFSET: 0x00001954

INSTANCE 0 ADDRESS: 0x00001954

This is the Customer Public Key Word21.

Table 1400: CUSTPUBKEYW21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W21																															

Table 1401: CUSTPUBKEYW21 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W21	0xffffffff		Customer Public Key Word21

23.1.2.119CUSTPUBKEYW22 Register

Customer Public Key Word22

OFFSET: 0x00001958

INSTANCE 0 ADDRESS: 0x00001958

This is the Customer Public Key Word22.

Table 1402: CUSTPUBKEYW22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W22																															

Table 1403: CUSTPUBKEYW22 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W22	0xffffffff		Customer Public Key Word22

23.1.2.120CUSTPUBKEYW23 Register

Customer Public Key Word23

OFFSET: 0x0000195C

INSTANCE 0 ADDRESS: 0x0000195C

This is the Customer Public Key Word23.

Table 1404: CUSTPUBKEYW23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W23																															

Table 1405: CUSTPUBKEYW23 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W23	0xffffffff		Customer Public Key Word23

23.1.2.121CUSTPUBKEYW24 Register

Customer Public Key Word24

OFFSET: 0x00001960

INSTANCE 0 ADDRESS: 0x00001960

This is the Customer Public Key Word24.

Table 1406: CUSTPUBKEYW24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W24																															

Table 1407: CUSTPUBKEYW24 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W24	0xffffffff		Customer Public Key Word24

23.1.2.122CUSTPUBKEYW25 Register

Customer Public Key Word25

OFFSET: 0x00001964

INSTANCE 0 ADDRESS: 0x00001964

This is the Customer Public Key Word25.

Table 1408: CUSTPUBKEYW25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W25																															

Table 1409: CUSTPUBKEYW25 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W25	0xffffffff		Customer Public Key Word25

23.1.2.123CUSTPUBKEYW26 Register

Customer Public Key Word26

OFFSET: 0x00001968

INSTANCE 0 ADDRESS: 0x00001968

This is the Customer Public Key Word26.

Table 1410: CUSTPUBKEYW26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W26																															

Table 1411: CUSTPUBKEYW26 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W26	0xffffffff		Customer Public Key Word26

23.1.2.124CUSTPUBKEYW27 Register

Customer Public Key Word27

OFFSET: 0x0000196C

INSTANCE 0 ADDRESS: 0x0000196C

This is the Customer Public Key Word27.

Table 1412: CUSTPUBKEYW27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W27																															

Table 1413: CUSTPUBKEYW27 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W27	0xffffffff		Customer Public Key Word27

23.1.2.125CUSTPUBKEYW28 Register

Customer Public Key Word28

OFFSET: 0x00001970

INSTANCE 0 ADDRESS: 0x00001970

This is the Customer Public Key Word28.

Table 1414: CUSTPUBKEYW28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W28																															

Table 1415: CUSTPUBKEYW28 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W28	0xffffffff		Customer Public Key Word28

23.1.2.126CUSTPUBKEYW29 Register

Customer Public Key Word29

OFFSET: 0x00001974

INSTANCE 0 ADDRESS: 0x00001974

This is the Customer Public Key Word29.

Table 1416: CUSTPUBKEYW29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W29																															

Table 1417: CUSTPUBKEYW29 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W29	0xffffffff		Customer Public Key Word29

23.1.2.127CUSTPUBKEYW30 Register

Customer Public Key Word30

OFFSET: 0x00001978

INSTANCE 0 ADDRESS: 0x00001978

This is the Customer Public Key Word30.

Table 1418: CUSTPUBKEYW30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W30																															

Table 1419: CUSTPUBKEYW30 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W30	0xffffffff		Customer Public Key Word30

23.1.2.128CUSTPUBKEYW31 Register

Customer Public Key Word31

OFFSET: 0x0000197C

INSTANCE 0 ADDRESS: 0x0000197C

This is the Customer Public Key Word31.

Table 1420: CUSTPUBKEYW31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W31																															

Table 1421: CUSTPUBKEYW31 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W31	0xffffffff		Customer Public Key Word31

23.1.2.129CUSTPUBKEYW32 Register

Customer Public Key Word32

OFFSET: 0x00001980

INSTANCE 0 ADDRESS: 0x00001980

This is the Customer Public Key Word32.

Table 1422: CUSTPUBKEYW32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W32																															

Table 1423: CUSTPUBKEYW32 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W32	0xffffffff		Customer Public Key Word32

23.1.2.130CUSTPUBKEYW33 Register

Customer Public Key Word33

OFFSET: 0x00001984

INSTANCE 0 ADDRESS: 0x00001984

This is the Customer Public Key Word33.

Table 1424: CUSTPUBKEYW33 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W33																															

Table 1425: CUSTPUBKEYW33 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W33	0xffffffff		Customer Public Key Word33

23.1.2.131CUSTPUBKEYW34 Register

Customer Public Key Word34

OFFSET: 0x00001988

INSTANCE 0 ADDRESS: 0x00001988

This is the Customer Public Key Word34.

Table 1426: CUSTPUBKEYW34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W34																															

Table 1427: CUSTPUBKEYW34 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W34	0xffffffff		Customer Public Key Word34

23.1.2.132CUSTPUBKEYW35 Register

Customer Public Key Word35

OFFSET: 0x0000198C

INSTANCE 0 ADDRESS: 0x0000198C

This is the Customer Public Key Word35.

Table 1428: CUSTPUBKEYW35 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W35																															

Table 1429: CUSTPUBKEYW35 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W35	0xffffffff		Customer Public Key Word35

23.1.2.133CUSTPUBKEYW36 Register

Customer Public Key Word36

OFFSET: 0x00001990

INSTANCE 0 ADDRESS: 0x00001990

This is the Customer Public Key Word36.

Table 1430: CUSTPUBKEYW36 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W36																															

Table 1431: CUSTPUBKEYW36 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W36	0xffffffff		Customer Public Key Word36

23.1.2.134CUSTPUBKEYW37 Register

Customer Public Key Word37

OFFSET: 0x00001994

INSTANCE 0 ADDRESS: 0x00001994

This is the Customer Public Key Word37.

Table 1432: CUSTPUBKEYW37 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W37																															

Table 1433: CUSTPUBKEYW37 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W37	0xffffffff		Customer Public Key Word37

23.1.2.135CUSTPUBKEYW38 Register

Customer Public Key Word38

OFFSET: 0x00001998

INSTANCE 0 ADDRESS: 0x00001998

This is the Customer Public Key Word38.

Table 1434: CUSTPUBKEYW38 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W38																															

Table 1435: CUSTPUBKEYW38 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W38	0xffffffff		Customer Public Key Word38

23.1.2.136CUSTPUBKEYW39 Register

Customer Public Key Word39

OFFSET: 0x0000199C

INSTANCE 0 ADDRESS: 0x0000199C

This is the Customer Public Key Word39.

Table 1436: CUSTPUBKEYW39 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W39																															

Table 1437: CUSTPUBKEYW39 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W39	0xffffffff		Customer Public Key Word39

23.1.2.137CUSTPUBKEYW40 Register

Customer Public Key Word40

OFFSET: 0x000019A0

INSTANCE 0 ADDRESS: 0x000019A0

This is the Customer Public Key Word40.

Table 1438: CUSTPUBKEYW40 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W40																															

Table 1439: CUSTPUBKEYW40 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W40	0xffffffff		Customer Public Key Word40

23.1.2.138CUSTPUBKEYW41 Register

Customer Public Key Word41

OFFSET: 0x000019A4

INSTANCE 0 ADDRESS: 0x000019A4

This is the Customer Public Key Word41.

Table 1440: CUSTPUBKEYW41 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W41																															

Table 1441: CUSTPUBKEYW41 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W41	0xffffffff		Customer Public Key Word41

23.1.2.139CUSTPUBKEYW42 Register

Customer Public Key Word42

OFFSET: 0x000019A8

INSTANCE 0 ADDRESS: 0x000019A8

This is the Customer Public Key Word42.

Table 1442: CUSTPUBKEYW42 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W42																															

Table 1443: CUSTPUBKEYW42 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W42	0xffffffff		Customer Public Key Word42

23.1.2.140CUSTPUBKEYW43 Register

Customer Public Key Word43

OFFSET: 0x000019AC

INSTANCE 0 ADDRESS: 0x000019AC

This is the Customer Public Key Word43.

Table 1444: CUSTPUBKEYW43 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W43																															

Table 1445: CUSTPUBKEYW43 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W43	0xffffffff		Customer Public Key Word43

23.1.2.141CUSTPUBKEYW44 Register

Customer Public Key Word44

OFFSET: 0x000019B0

INSTANCE 0 ADDRESS: 0x000019B0

This is the Customer Public Key Word44.

Table 1446: CUSTPUBKEYW44 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W44																															

Table 1447: CUSTPUBKEYW44 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W44	0xffffffff		Customer Public Key Word44

23.1.2.142CUSTPUBKEYW45 Register

Customer Public Key Word45

OFFSET: 0x000019B4

INSTANCE 0 ADDRESS: 0x000019B4

This is the Customer Public Key Word45.

Table 1448: CUSTPUBKEYW45 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W45																															

Table 1449: CUSTPUBKEYW45 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W45	0xffffffff		Customer Public Key Word45

23.1.2.143CUSTPUBKEYW46 Register

Customer Public Key Word46

OFFSET: 0x000019B8

INSTANCE 0 ADDRESS: 0x000019B8

This is the Customer Public Key Word46.

Table 1450: CUSTPUBKEYW46 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W46																															

Table 1451: CUSTPUBKEYW46 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W46	0xffffffff		Customer Public Key Word46

23.1.2.144CUSTPUBKEYW47 Register

Customer Public Key Word47

OFFSET: 0x000019BC

INSTANCE 0 ADDRESS: 0x000019BC

This is the Customer Public Key Word47.

Table 1452: CUSTPUBKEYW47 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W47																															

Table 1453: CUSTPUBKEYW47 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W47	0xffffffff		Customer Public Key Word47

23.1.2.145CUSTPUBKEYW48 Register

Customer Public Key Word48

OFFSET: 0x000019C0

INSTANCE 0 ADDRESS: 0x000019C0

This is the Customer Public Key Word48.

Table 1454: CUSTPUBKEYW48 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W48																															

Table 1455: CUSTPUBKEYW48 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W48	0xffffffff		Customer Public Key Word48

23.1.2.146CUSTPUBKEYW49 Register

Customer Public Key Word49

OFFSET: 0x000019C4

INSTANCE 0 ADDRESS: 0x000019C4

This is the Customer Public Key Word49.

Table 1456: CUSTPUBKEYW49 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W49																															

Table 1457: CUSTPUBKEYW49 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W49	0xffffffff		Customer Public Key Word49

23.1.2.147CUSTPUBKEYW50 Register

Customer Public Key Word50

OFFSET: 0x000019C8

INSTANCE 0 ADDRESS: 0x000019C8

This is the Customer Public Key Word50.

Table 1458: CUSTPUBKEYW50 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W50																															

Table 1459: CUSTPUBKEYW50 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W50	0xffffffff		Customer Public Key Word50

23.1.2.148CUSTPUBKEYW51 Register

Customer Public Key Word51

OFFSET: 0x000019CC

INSTANCE 0 ADDRESS: 0x000019CC

This is the Customer Public Key Word51.

Table 1460: CUSTPUBKEYW51 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W51																															

Table 1461: CUSTPUBKEYW51 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W51	0xffffffff		Customer Public Key Word51

23.1.2.149CUSTPUBKEYW52 Register

Customer Public Key Word52

OFFSET: 0x000019D0

INSTANCE 0 ADDRESS: 0x000019D0

This is the Customer Public Key Word52.

Table 1462: CUSTPUBKEYW52 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W52																															

Table 1463: CUSTPUBKEYW52 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W52	0xffffffff		Customer Public Key Word52

23.1.2.150CUSTPUBKEYW53 Register

Customer Public Key Word53

OFFSET: 0x000019D4

INSTANCE 0 ADDRESS: 0x000019D4

This is the Customer Public Key Word53.

Table 1464: CUSTPUBKEYW53 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W53																															

Table 1465: CUSTPUBKEYW53 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W53	0xffffffff		Customer Public Key Word53

23.1.2.151CUSTPUBKEYW54 Register

Customer Public Key Word54

OFFSET: 0x000019D8

INSTANCE 0 ADDRESS: 0x000019D8

This is the Customer Public Key Word54.

Table 1466: CUSTPUBKEYW54 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W54																															

Table 1467: CUSTPUBKEYW54 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W54	0xffffffff		Customer Public Key Word54

23.1.2.152CUSTPUBKEYW55 Register

Customer Public Key Word55

OFFSET: 0x000019DC

INSTANCE 0 ADDRESS: 0x000019DC

This is the Customer Public Key Word55.

Table 1468: CUSTPUBKEYW55 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W55																															

Table 1469: CUSTPUBKEYW55 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W55	0xffffffff		Customer Public Key Word55

23.1.2.153CUSTPUBKEYW56 Register

Customer Public Key Word56

OFFSET: 0x000019E0

INSTANCE 0 ADDRESS: 0x000019E0

This is the Customer Public Key Word56.

Table 1470: CUSTPUBKEYW56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W56																															

Table 1471: CUSTPUBKEYW56 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W56	0xffffffff		Customer Public Key Word56

23.1.2.154CUSTPUBKEYW57 Register

Customer Public Key Word57

OFFSET: 0x000019E4

INSTANCE 0 ADDRESS: 0x000019E4

This is the Customer Public Key Word57.

Table 1472: CUSTPUBKEYW57 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W57																															

Table 1473: CUSTPUBKEYW57 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W57	0xffffffff		Customer Public Key Word57

23.1.2.155CUSTPUBKEYW58 Register

Customer Public Key Word58

OFFSET: 0x000019E8

INSTANCE 0 ADDRESS: 0x000019E8

This is the Customer Public Key Word58.

Table 1474: CUSTPUBKEYW58 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W58																															

Table 1475: CUSTPUBKEYW58 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W58	0xffffffff		Customer Public Key Word58

23.1.2.156CUSTPUBKEYW59 Register

Customer Public Key Word59

OFFSET: 0x000019EC

INSTANCE 0 ADDRESS: 0x000019EC

This is the Customer Public Key Word59.

Table 1476: CUSTPUBKEYW59 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W59																															

Table 1477: CUSTPUBKEYW59 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W59	0xffffffff		Customer Public Key Word59

23.1.2.157CUSTPUBKEYW60 Register

Customer Public Key Word60

OFFSET: 0x000019F0

INSTANCE 0 ADDRESS: 0x000019F0

This is the Customer Public Key Word60.

Table 1478: CUSTPUBKEYW60 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W60																															

Table 1479: CUSTPUBKEYW60 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W60	0xffffffff		Customer Public Key Word60

23.1.2.158CUSTPUBKEYW61 Register

Customer Public Key Word61

OFFSET: 0x000019F4

INSTANCE 0 ADDRESS: 0x000019F4

This is the Customer Public Key Word61.

Table 1480: CUSTPUBKEYW61 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W61																															

Table 1481: CUSTPUBKEYW61 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W61	0xffffffff		Customer Public Key Word61

23.1.2.159CUSTPUBKEYW62 Register

Customer Public Key Word62

OFFSET: 0x000019F8

INSTANCE 0 ADDRESS: 0x000019F8

This is the Customer Public Key Word62.

Table 1482: CUSTPUBKEYW62 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W62																															

Table 1483: CUSTPUBKEYW62 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W62	0xffffffff		Customer Public Key Word62

23.1.2.160CUSTPUBKEYW63 Register

Customer Public Key Word63

OFFSET: 0x000019FC

INSTANCE 0 ADDRESS: 0x000019FC

This is the Customer Public Key Word63.

Table 1484: CUSTPUBKEYW63 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBKEY_W63																															

Table 1485: CUSTPUBKEYW63 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-KEY_W63	0xffffffff		Customer Public Key Word63

23.1.2.161CUSTOMERKEY0 Register

128-bit customer key. Customer SW can access the protected upper half of info0 by writing this key value into the security lock register

OFFSET: 0x00001A00

INSTANCE 0 ADDRESS: 0x00001A00

customer_key[31:0]

Table 1486: CUSTOMERKEY0 Register

31	30	29	28	27	26	25	24	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 1487: CUSTOMERKEY0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Word 0 of the customer key.

23.1.2.162CUSTOMERKEY1 Register

128-bit customer key.

OFFSET: 0x00001A04

INSTANCE 0 ADDRESS: 0x00001A04

customer_key[63:32]

Table 1488: CUSTOMERKEY1 Register

31	30	29	28	27	26	25	24	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 1489: CUSTOMERKEY1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Word 1 of the customer key.

23.1.2.163CUSTOMERKEY2 Register

128-bit customer key.

OFFSET: 0x00001A08

INSTANCE 0 ADDRESS: 0x00001A08

customer_key[95:64]

Table 1490: CUSTOMERKEY2 Register

31	30	29	28	27	26	25	24	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 1491: CUSTOMERKEY2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Word 2 of the customer key.

23.1.2.164CUSTOMERKEY3 Register

128-bit customer key.

OFFSET: 0x00001A0C

INSTANCE 0 ADDRESS: 0x00001A0C

customer_key[127:96]

Table 1492: CUSTOMERKEY3 Register

31	30	29	28	27	26	25	24	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 1493: CUSTOMERKEY3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CHUNKS	0xffffffff		Word 3 of the customer key.

23.1.2.165CUSTPUBHASHW0 Register

Customer Public Key Hash Word0

OFFSET: 0x00001A10

INSTANCE 0 ADDRESS: 0x00001A10

This is the Customer Public Key Hash Word0.

Table 1494: CUSTPUBHASHW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBHASH_W0																															

Table 1495: CUSTPUBHASHW0 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-HASH_W0	0xffffffff		Customer Public Key Hash Word0

23.1.2.166CUSTPUBHASHW1 Register

Customer Public Key Hash Word1

OFFSET: 0x00001A14

INSTANCE 0 ADDRESS: 0x00001A14

This is the Customer Public Key Hash Word1.

Table 1496: CUSTPUBHASHW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBHASH_W1																															

Table 1497: CUSTPUBHASHW1 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-HASH_W1	0xffffffff		Customer Public Key Hash Word1

23.1.2.167CUSTPUBHASHW2 Register

Customer Public Key Hash Word2

OFFSET: 0x00001A18

INSTANCE 0 ADDRESS: 0x00001A18

This is the Customer Public Key Hash Word2.

Table 1498: CUSTPUBHASHW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBHASH_W2																															

Table 1499: CUSTPUBHASHW2 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-HASH_W2	0xffffffff		Customer Public Key Hash Word2

23.1.2.168CUSTPUBHASHW3 Register

Customer Public Key Hash Word3

OFFSET: 0x00001A1C

INSTANCE 0 ADDRESS: 0x00001A1C

This is the Customer Public Key Hash Word3.

Table 1500: CUSTPUBHASHW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CUSTPUBHASH_W3																															

Table 1501: CUSTPUBHASHW3 Register Bits

Bit	Name	Reset	RW	Description
31:0	CUSTPUB-HASH_W3	0xffffffff		Customer Public Key Hash Word3

Index

24. Ordering Information

Table 1502: Ordering Information

Orderable Part Number	Flash	RAM	Package	Packing	Temperature Range	Availability
AMA3B1KK-KBR	1 MB	384 KB	81-pin BGA	Tape and Reel	–40 to 85°C	Now
AMA3B1KK-KCR	1 MB	384 KB	66-pin CSP	Tape and Reel	–40 to 85°C	Apr 2019

25. Document Revision History

Table 1503: Document Revision List

Revision	Date	Description
0.6	Feb 2018	Initial alpha release - Updated Appendix
0.7	Jun 2018	Pin Cfg: - VDDA pin on CSP package corrected in Pin List and Function Table - DVDD pin added in BGA and CSP pkg diagrams and in Pin List and Function Table I2C/SPI Master Module: Functional Description details added ADC: ADC register set expanded to include DMA control/status registers CTIMER: Corrected CTIMER Pad Input Connections table GPIO: PADREGA corrected for SLMISO/SLMOSI function selections for PAD1 and PAD2 Electrical: - Updated VESDCDM - ESD Charged Device Model (CDM) - Updated VESDHBM - ESD Human Body Model (HBM) - Updated Latch-up Current
0.8	Aug 2018	Package: CSP Package Drawing added Appendix: Info0 information updated Electrical: BLE specifications updated
0.9	Feb 2019	Pincfg: For CSP Pin Configuration Diagram and Pin List and Function Table, VSSB (C7) and VDDBH_SW (C8) have been interchanged, CSP balls for GPIO4, 9, 10, 13, 14, 15, 21, 23, 24, 40 and 41 have been updated. Preliminary designation removed. System Core (Flash): Updated note about allowable number of program cycles between erase cycles. Clock Gen: Clock tree diagram added. Reset Module: Updated Block diagram for the Reset Generator Module Electricals: - ADC specifications updated - Coremark current at 3.3V/48MHz corrected
0.9.1	TBD	MSPI: Pin Muxing table corrected Ordering Information: CSP package added

Contact Information

Address	Ambiq Micro, Inc. 6500 River Place Blvd. Building 7, Suite 200 Austin, TX 78730-1156
Phone	+1 (512) 879-2850
Website	https://ambiqmicro.com/
General Information	info@ambiqmicro.com
Sales	sales@ambiqmicro.com
Technical Support	https://support.ambiqmicro.com

Legal Information and Disclaimers