MKW40Z/30Z/20Z Reference Manual

Bluetooth Low Energy and IEEE 802.15.4 System on a Chip (SoC)
Reference Manual

Document Number: MKW40Z160RM
Rev. 1.2, 10/2015

Contents

Section number **Title** **Page**

Chapter 1 Introduction

Chapter 2 Signal Multiplexing and Signal Descriptions

Chapter 3 Chip Configuration

Section number Title Page

Section number Title Page

Chapter 4
Memory Map

Chapter 5
Clock Distribution

Chapter 6
Reset and Boot

Section number | Title | Page

Chapter 7
Power Management

Chapter 8
Security

Chapter 9
Debug

Chapter 10
Port control and interrupt (PORT)

Section number | Title | Page

Chapter 11
System Integration Module (SIM)

Section number Title Page

Chapter 12
System Mode Controller (SMC)

Chapter 13
Power Management Controller (PMC)

Chapter 14
DCDC Converter (DCDC)

Section number Title Page

Chapter 15
Low-Leakage Wakeup Unit (LLWU)

Section number Title Page

Chapter 16
Reset Control Module (RCM)

Chapter 17
Bit Manipulation Engine (BME)

Section number Title Page

Chapter 18
Miscellaneous Control Module (MCM)

Chapter 19
Micro Trace Buffer (MTB)

Chapter 20
Crossbar Switch Lite (AXBS-Lite)

Chapter 21
Peripheral Bridge (AIPS-Lite)

Chapter 22
Direct Memory Access Multiplexer (DMAMUX)

Chapter 23
DMA Controller Module

Chapter 24
Multipurpose Clock Generator (MCG)

Section number	Title	Page

Section number Title Page

Chapter 25
32 kHz Oscillator (32kRTC)

Chapter 26
Flash Memory Controller (FMC)

Chapter 27
Flash Memory Module (FTFA)

Section number Title Page

Chapter 28
Analog-to-digital converter (ADC)

Chapter 29
Comparator (CMP)

Chapter 30
12-bit digital-to-analog converter (DAC)

Chapter 31
Front End Module Interface

Chapter 32
Timer/PWM Module (TPM)

Chapter 33
Periodic interrupt timer (PIT)

Section number Title Page

Chapter 34
Low-power timer (LPTMR)

Chapter 35
Real Time Clock (RTC)

Chapter 36
Serial Peripheral Interface (SPI)

Chapter 37
Inter-Integrated Circuit (I2C)

Section number Title Page

Section number | Title | Page

Chapter 38
Universal Asynchronous Receiver/Transmitter (UART0)

Section number | Title | Page

Chapter 39
Carrier Modulator Transmitter (CMT)

Chapter 40
General-purpose input/output (GPIO)

Section number Title Page

Chapter 41
Touch Sensing Input (TSI)

Chapter 42
LP Trusted Cryptography (LTC)

Section number Title Page

Chapter 43
True Random Number Generator (TRNG)

Chapter 44
2.4 GHz Radio

Chapter 45
Clock and Reset Module

Section number Title Page

Chapter 46
Radio System Integration Module (RSIM)

Chapter 47
PLL Digital

Chapter 48
RX Digital

Chapter 49
TX Digital

Section number Title Page

Chapter 50
Transceiver Sequence Manager (TSM)

Section number Title Page

Chapter 51
Oscillator

Chapter 52
Transceiver Registers

Section number Title Page

Section number | Title | Page

Section number Title Page

Chapter 53
BLE Link Layer Registers

Chapter 54
BLE RF Registers

Chapter 55
Zigbee Registers

Section number Title Page

Chapter 1
Introduction

This section provides high-level descriptions of the modules available on the devices covered by this document.

1.1 Introduction

The KW4x is an ultra low power, highly integrated single-chip device that enables Bluetooth low energy (BLE) IEEE Std. 802.15.4/ZigBee RF connectivity for portable, extremely low-power embedded systems. Applications include portable health care devices, wearable sports and fitness devices, AV remote controls, computer keyboards and mice, gaming controllers, access control, security systems, smart energy and home area networks.

The KW4x SoC integrates a radio transceiver operating in the 2.36GHz to 2.48GHz range supporting a range of FSK/GFSK and O-QPSK modulations, an ARM Cortex-M0+ CPU, 160 KB Flash and 20 KB SRAM, BLE Link Layer hardware, 802.15.4 packet processor hardware and peripherals optimized to meet the requirements of the target applications.

The KW4x's radio frequency transceiver is compliant with Bluetooth version 4.1 for Low Energy (aka Bluetooth Smart), and the IEEE 802.15.4-2011 standard using O-QPSK in the 2.4 GHz ISM band and the IEEE 802.15.4j MBAN frequency range spanning from 2.36 GHz to 2.40 GHz. In addition, the KW4x allows the Bluetooth Low Energy protocol to be used in the MBAN frequency range for proprietary applications.

The KW4x can be used in applications as a "BlackBox" modem in order to add BLE or IEEE Std. 802.15.4 connectivity to an existing embedded controller system, or may be used as a stand-alone smart wireless sensor with embedded application where no host controller is required.

KW4x Flash and SRAM memory is available for applications and communication protocols using a choice of Freescale or 3rd party software development tools.

The RF section of the KW4x is optimized to require very few external components, achieving the smallest RF footprint possible on a printed circuit board.

Extremely long battery life is achieved though efficiency of code execution in the Cortex-M0+ CPU core and the multiple low power operating modes of the KW4x. Additionally, an integrated DC-DC converter enables a wide operating range from 0.9V to 4.2V. The DC-DC in Buck mode allows KW4x to operate from a single coin cell battery with a significant reduction of peak Rx and Tx current consumption. The DC-DC in boost mode allows a single alkaline battery to be used throughout its entire useful voltage range of 0.9V to 1.8V.

Table 1-1. KW4x Part Numbers

Part Number	Protocols Supported	Package	Size
MKW20Z160VHT4/R	IEEE 802.15.4/ZigBee	48-pin MAPLGA	7 mm x 7 mm, 0.5 mm pitch
MKW30Z160VHM4/R	Bluetooth LE	32-pin MAPLGA	5 mm x 5 mm, 0.5 mm pitch
MKW40Z160VHT4/R	Bluetooth LE, and IEEE 802.15.4/ZigBee	48-pin MAPLGA	7 mm x 7 mm, 0.5 mm pitch

1.2 Features Overview

The following section lists the features of the devices.

32-bit Cortex M0+ (enhanced M0) Central Processor Unit (CPU)

- Up to 48MHz core frequency across temperature range of –40 °C to 85 °C
- Supports up to 32 interrupt request sources
- 2-stage pipeline microarchitecture for reduced power consumption and improved architectural performance (cycles per instruction)
- Bit Manipulation Engine (BME) for improved bit handling of peripheral modules
- Binary compatible instruction set architecture with the CM0 core
- Thumb instruction set combines high code density with 32-bit performance
- Serial Wire Debug (SWD) reduces the number of pins required for debugging
- Micro Trace Buffer (MTB) provides lightweight program trace capabilities using system RAM as the destination memory

Input Voltage operation

- DCDC in Buck configurations supports 1.8V to 4.2 V
- DCDC in Boost configuration supports 0.9V to 1.8V
- Bypass Mode 1.7 V to 3.6 V (1.45V - 3.6V for the RF and 32MHz OSC supplies)

On-chip Memory

- 160 KB of Flash memory, read/program/erase over full operating voltage and temperature
- 20 KB of Low Power Random access memory (SRAM). Memory retention in most low power modes
- Security circuitry to prevent unauthorized access to SRAM and Flash contents

Power-Saving

- Multiple power modes including low leakage state-retention and memory-retention modes
- Peripheral clock enable registers can disable clocks to unused modules, reducing currents.

System Clock Source Options

- Reference Oscillator — crystal reference oscillator, supports 32MHz
- 32kHz Oscillator —32.768 kHz crystal reference oscillator
- Multipurpose Clock Generator(MCG)
  - Frequency-locked loop (FLL) controlled by internal or external reference
    - 20 MHz to 48 MHz FLL output
  - Internal reference clocks — Can be used as a clock source for other on-chip peripherals
    - On-chip RC oscillator range of 31.25 kHz to 39.0625 kHz with 2% accuracy across full temperature range
    - On-chip 4MHz oscillator with 5% accuracy across full temperature range.

System Protection

- Standard Watchdog reset with option to run from dedicated 1 kHz internal clock source or bus clock
- Low-voltage detection with reset or interrupt; selectable trip points
- HardFault exception on attempts to execute undefined instructions or access to undefined memory space
- LOCKUP reset resource from core
- Flash Read\Write protection
- Firmware distribution protection: Flash can be marked execute-only on a per-sector (4KB) basis to prevent firmware contents from being read by 3rd parties.

Development Support

- Two-wire Serial Wire Debug interface
- Breakpoint unit supporting up to 2 hardware breakpoints
- Watchpoint unit supporting up to 2 watchpoints
- Micro Trace Buffer provides program trace capabilities

Peripherals

- DMA — 4-channel DMA. Bus master provides very configurable source-to-destination data movement capabilities supporting either software-triggered or peripheral-paced transfers
- ADC — up to 5 external channels, 16-bit resolution analog-to-digit converter, fully functional in the entire voltage range. Performance for the ADC depends on package pinout.
- DAC — 12-bit resolution.
- HSCMP — high speed comparator with internal 6-bit DAC
- PIT — 2-channel 32-bit timer module that can be used to assert interrupts or to provide one more time base.
- LPTPM — One 4-channel, and two 2-channel; Basic TPM function. Timer/Pulse-Width Modulator Module supporting input capture, output compare. Quadrature decode feature supported on the two 2-channel LPTPMs
- LPTMR — Low Power Timer that can wakeup CPU from all low power modes
- RTC — Robust 32-bit Real Timer Clock with hardware compensation
- CMT — Carrier Modulation Timer used to drive IR communications.
- AESA — AES Accelerator with DMA support
- UART — Serial Communication Interface with DMA support and hardware flow control (RTS\CTS)
- SPI — Two Serial Peripheral Interfaces with DMA support
- I2C — Two Inter-Integrated Circuit modules with SMBUS 2.0 and DMA support
- GPIO — Port interrupt capability on all the GPIO pins.
- TRNG - True Random Number Generator
- Capacitive Touch Sensing Interface (TSI) with full low-power support and minimal current adder when enabled

Radio

- 2.4GHz ISM band (2400-2483.5MHz) and MBAN 2360-2400MHz operation
- (G)FSK and O-QPSK Modulation
- Supported Standards
  - Bluetooth v4.1 Low Energy
  - 802.15.4-2011 O-QPSK in the 2.45 GHz band, and O-QPSK in the 2.38GHz band (802.15.4j)
  - ZigBee RF4CE and ZigBee PRO
  - Continua Health Alliance
- Receiver performance
  - Receive sensitivity of -92 dBm for BLE
  - Receive sensitivity of -102 dBm typical for IEEE Std. 802.15.4 (Zigbee)
  - Accurate RSSI indication for RF inputs from -102 dBm to 0 dBm

- Programmable output power from -20 dBm to +5 dBm for bypass and buck modes of operation
- Bluetooth Low Energy Link Layer hardware
- Hardware acceleration for IEEE Std. 802.15.4 packet processing
- 32 MHz crystal reference oscillator
- Differential RF port shared by both transmit and receive, used with either an external balun or passing matching network
- Supports diversity antenna options for 802.15.4
- Supports dual PAN for 802.15.4 with hardware-assisted address matching acceleration
- Low external component count
- Supports external PA and LNA

1.3 Feature Summary

The following table lists the features integrated on device.

Table 1-2. Feature Summary

Feature	Device
Hardware Characteristics	
Package	48-pin Laminate QFN (7 x 7 mm, 0.5 mm pitch) 32-pin Laminate QFN (5 x 5 mm, 0.5 mm pitch)
System	
Central processing unit (CPU)	ARM Cortex-M0+ core (32 bit)
Max. CPU frequency	48MHz (4MHz in VLPR mode)
Max. Bus frequency	24MHz (1MHz in VLPR mode)
Nested vectored Interrupt controller (NVIC)	32 vectored interrupts 4 programmable interrupt priority levels
Low Power Modes	Run, Wait, Stop and Partial Stop Compute operation mode Very low power run (VLPR), wait (VLPW), and stop (VLPS) Low Leakage Stop (LLS3, LLS2) Very Low Leakage Stop (VLLS3, VLL2, VLLS1, VLLS0)
Low-leakage Wakeup Unit (LLWU)	external wake-up pins with digital glitch filter as well as internal wake-up sources
Non-maskable interrupt (NMI)	Yes
Software COP (COP)	Yes
Debug and Trace	2-pin serial wire debug (SWD) Micro trace buffer (MTB) + Data Watchpoint and Trace (DWT)

Table continues on the next page...

Table 1-2. Feature Summary (continued)

Feature	Device
Unique Identification (ID) Numbers	80-bit wide unique device ID Additional 40bit unique value for creating MAC address
Memory	
Flash memory	160KB with 64 byte flash cache
Random-access memory (RAM)	20 KB
System Register File	32 bytes
Clocks	
Reference crystal oscillator or resonator	Crystal reference oscillator, ability to bypass oscillator with external clock. Supports 32MHz
32 kHz External crystal oscillator	Supports 32kHz or 32.768 kHz crystal/resonator, or external 32/32.768kHz clock
Internal clock references	31.25 to 39.063 kHz oscillator with ±2% max. deviation across temperature 4MHz oscillator with ±5% max. deviation across temperature 1 kHz oscillator
Frequency-locked loop (FLL)	20 - 48 MHz
Human-Machine Interface (HMI)	
General-purpose input/output (GPIO)	Default to disabled (no leakage) Hysteresis and configurable pull up/down device on all input pins Configurable drive strength and/or slew rate on some pins up to 6 pins (package dependent) with 20 mA high current drive ability Single cycle GPIO control via IOPORT
Touch Sensor Input (TSI)	Up to 16-channels (package dependent) Selectable single channel wakeup source available in all low power modes DMA support
General Purpose Analog	
Power management controller (PMC)	Low voltage warning and detect with selectable trip points 1 kHz LPO
16-bit analog-to-digital converter (ADC)	Up to 5 external channels Linear successive approximation algorithm Internal Temp Sensor and Battery Monitor DMA support
High speed comparator (HSCMP) with internal 6-bit digital-to-analog converter (DAC)	Up to 5 external channels
12-bit DAC	2x16bit data buffer DMA support
Timers	
16-bit TPM timer (LPTPM x3)	one 4-channel without quadrature decode two 2-channel with quadrature decode basic TPM function

Table continues on the next page...

Table 1-2. Feature Summary (continued)

Feature	Device
	PWM generation built in
32-bit Programmable interrupt timer (PIT)	2 channel
Real-time clock (RTC)	32-bit seconds counter 16-bit prescaler with compensation
Low-power Timer (LPTMR)	1-channel, 16-bit pulse counter or periodic interrupt functional in all power modes
Communication Interfaces	
Serial peripheral interface (2x SPI)	Two DSPI with 4-entry TX/CMD and RX FIFOs Master mode and slave mode functions Supports multiple chip selects in master mode Programmable transfer lengths DMA Support
Inter-Integrated Circuit (2x I^{2}C)	Two I2C modules
Universal asynchronous receiver/transmitter (LPUART)	Standard features Tx pin pseudo open drain with enable/disable programmable configurable x4 to x32 oversampling Functional in STOP/VLPS modes Hardware Flow Control (RTS/CTS) DMA Support
Carrier Modulation Timer (CMT)	Direct drive of IR LED.
Radio	
Operating Frequency range	ISM: 2400 - 2483.5 MHz MBAN: 2360 - 2400MHz
Antenna and RF match support	Support for External, PCB and Ceramic chip antenna
Common Rx/Tx antenna terminals	Use same 2 pins for Rx and Tx
Extended range options	Shall be capable of supporting an external PA of +30dB gain Shall support +10 dB external LNA
Security Support	
Encryption	AES Accelerator supporting ECB, CBC, CTR, CCM and CCM*, CMAC, and XCBC-MAC modes
TRNG	True Random Number Generator
ESD/EFT	
Human Body Model (HBM) JEDEC STD 2, method A114	>+ / -2000 V All package pins,including RF pins.
Charge Device Model (CDM) JEDEC STD 2, method C101	+ / -500 V All package pins,including RF pins.

1.4 Block Diagram

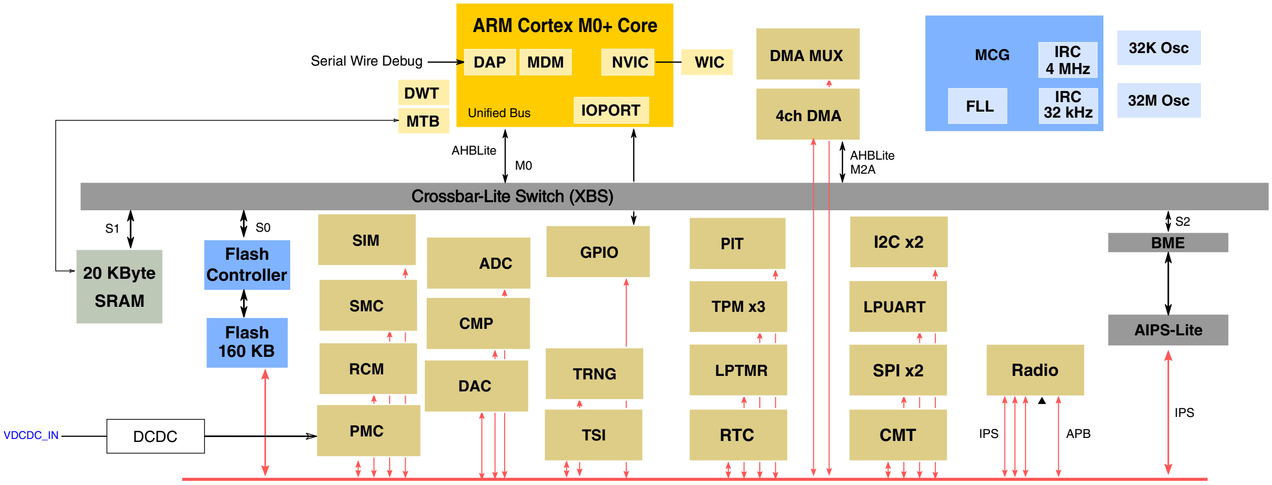


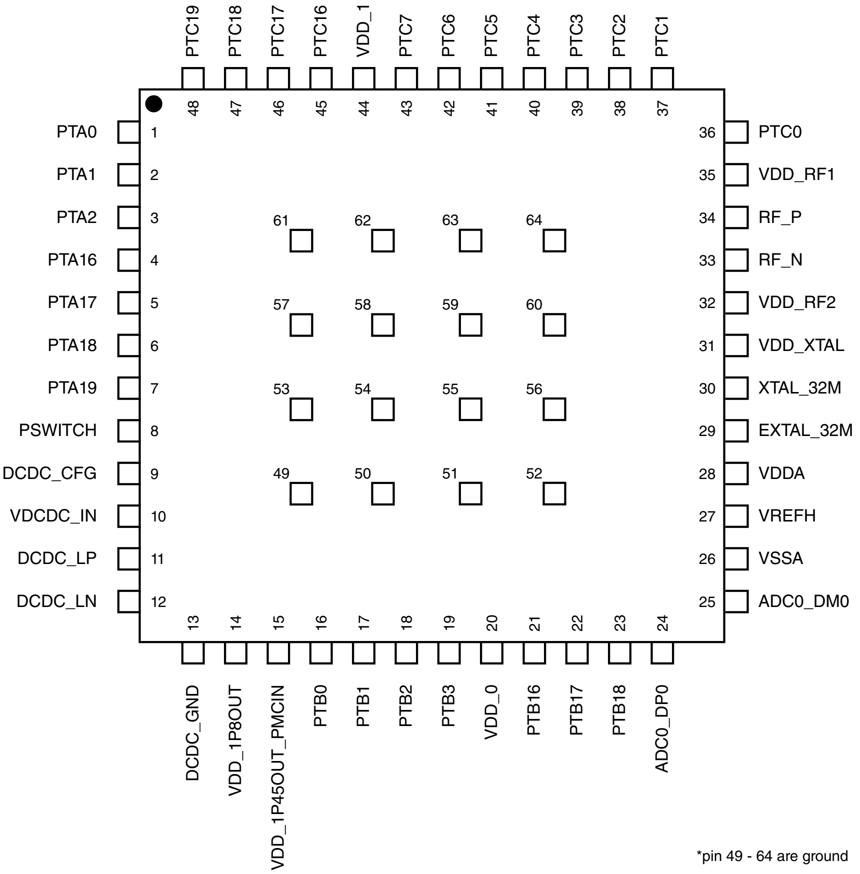
Figure 1-1. KW40Z Detailed Block Diagram

Chapter 2
Signal Multiplexing and Signal Descriptions

This section illustrates which of this device's signals are multiplexed on which external pin.

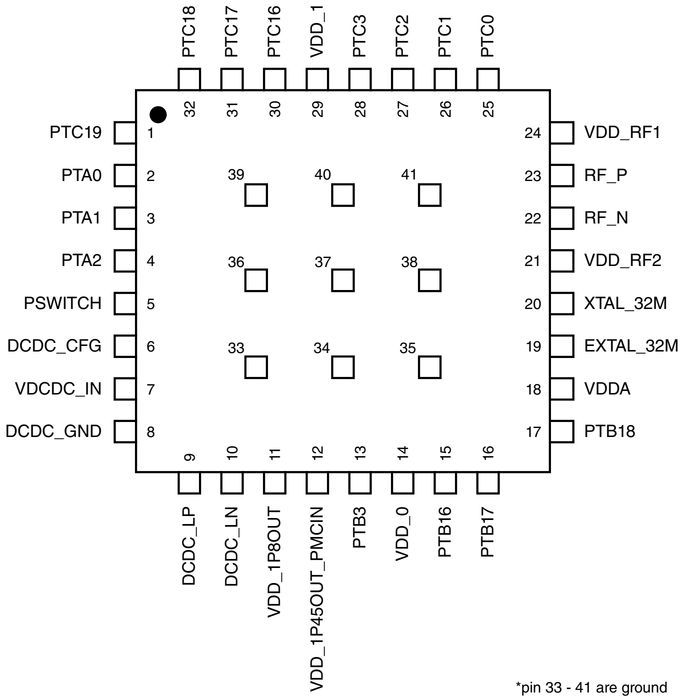
2.1 Pinouts

Device pinout are shown in figures below.



*pin 49 - 64 are ground

Figure 2-1. 48-pin Laminate QFN pinout diagram



*pin 33 - 41 are ground

Figure 2-2. 32-pin Laminate QFN pinout diagram

2.2 Signal Multiplexing and Pin Assignments

The following table shows the signals available on each pin and locations of these pins on the packages supported by this device. The Port Control Module is responsible for selecting which ALT functional is available on each PTxy pin.

Table 2-1. KW40Z Pin Assignments

48 Laminate QFN	32 Laminate QFN	Pin Name[1]	DEFAULT	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
1	2	PTA0	SWD_DIO	TSI0_CH8	PTA0	SPI0_PCS1	—	—	TPM1_CH0	—	SWD_DIO
2	3	PTA1	SWD_CLK	TSI0_CH9	PTA1	—	—	—	TPM1_CH1	—	SWD_CLK
3	4	PTA2	RESET_b	—	PTA2	—	—	—	TMP0_CH3	—	RESET_b
4	—	PTA16	DISABLED	TSI0_CH10	PTA16/LLWU_P4	SPI1_SOUT	—	—	TPM0_CH0	—	—

Table continues on the next page...

Table 2-1. KW40Z Pin Assignments (continued)

48 Laminate QFN	32 Laminate QFN	Pin Name[1]	DEFAULT	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
5	—	PTA17	DISABLED	TSI0_CH11	PTA17/LLWU_P5	SPI1_SIN	—	—	TPM_CLKIN1	—	—
6	—	PTA18	DISABLED	TSI0_CH12	PTA18/LLWU_P6	SPI1_SCK	—	—	TPM2_CH0	—	—
7	—	PTA19	DISABLED	TSI0_CH13	PTA19/LLWU_P7	SPI1_PCS0	—	—	TPM2_CH1	—	—
8	5	PSWITCH	PSWITCH	PSWITCH	—	—	—	—	—	—	—
9	6	DCDC_CFG	DCDC_CFG	DCDC_CFG	—	—	—	—	—	—	—
10	7	VDCDC_IN	VDCDC_IN	VDCDC_IN	—	—	—	—	—	—	—
11	9	DCDC_LP	DCDC_LP	DCDC_LP	—	—	—	—	—	—	—
13	8	DCDC_GND	DCDC_GND	DCDC_GND	—	—	—	—	—	—	—
14	11	VDD_1P8OUT	VDD_1P8OUT	VDD_1P8OUT	—	—	—	—	—	—	—
12	10	DCDC_LN	DCDC_LN	DCDC_LN	—	—	—	—	—	—	—
15	12	VDD_1P45OUT_PMCIN	VDD_1P45OUT_PMCIN	VDD_1P45OUT_PMCIN	—	—	—	—	—	—	—
16	—	PTB0	DISABLED	—	PTB0/LLWU_P8	—	I2C0_SCL	CMP0_OUT	TPM0_CH1	—	CLKOUT
17	—	PTB1	ADC0_SE1/CMP0_IN5	ADC0_SE1/CMP0_IN5	PTB1	—	I2C0_SDA	LPTMR0_ALT1	TPM0_CH2	—	CMT_IRO
18	—	PTB2	ADC0_SE3/CMP0_IN3	ADC0_SE3/CMP0_IN3	PTB2	—	—	—	TPM1_CH0	—	—
19	13	PTB3	ADC0_SE2/CMP0_IN4	ADC0_SE2/CMP0_IN4	PTB3	—	—	CLKOUT	TPM1_CH1	—	RTC_CLKOUT
20	14	VDD_0	—	—	—	—	—	—	—	—	—
21	15	PTB16	EXTAL32K	EXTAL32K	PTB16	—	I2C1_SCL	—	TPM2_CH0	—	—
22	16	PTB17	XTAL32K	XTAL32K	PTB17	—	I2C1_SDA	—	TPM2_CH1	—	—
23	17	PTB18	NMI_b	DAC0_OUT/ADC0_SE4/CMP0_IN2	PTB18	—	I2C1_SCL	TPM_CLKIN0	TPM0_CH0	—	NMI_b
24	—	ADC0_DP0	ADC0_DP0/CMP0_IN0	ADC0_DP0/CMP0_IN0	—	—	—	—	—	—	—
25	—	ADC0_DM0	—	ADC0_DM0/CMP0_IN1	—	—	—	—	—	—	—

Table continues on the next page...

Table 2-1. KW40Z Pin Assignments (continued)

48 Laminate QFN	32 Laminate QFN	Pin Name[1]	DEFAULT	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
26	—	VSSA	VSSA	VSSA	—	—	—	—	—	—	—
27	—	VREFH	VREFH	VREFH	—	—	—	—	—	—	—
28	18	VDDA	VDDA	VDDA	—	—	—	—	—	—	—
29	19	EXTAL_32M	EXTAL_32M	EXTAL_32M	—	—	—	—	—	—	—
30	20	XTAL_32M	XTAL_32M	XTAL_32M	—	—	—	—	—	—	—
31	—	VDD_XTAL	VDD_XTAL	VDD_XTAL	—	—	—	—	—	—	—
32	21	VDD_RF2	VDD_RF2	VDD_RF2	—	—	—	—	—	—	—
33	22	RF_N	RF_N	RF_N	—	—	—	—	—	—	—
34	23	RF_P	RF_P	RF_P	—	—	—	—	—	—	—
35	24	VDD_RF1	VDD_RF1	VDD_RF1	—	—	—	—	—	—	—
36	25	PTC0	DISABLED	—	PTC0/LLWU_P9	ANT_A	I2C0_SCL	UART0_CTS_b	TPM0_CH1	—	—
37	26	PTC1	DISABLED	—	PTC1	ANT_B	I2C0_SDA	UART0_RTS_b	TPM0_CH2	—	BLE_ACTIVE
38	27	PTC2	DISABLED	TSI0_CH14	PTC2/LLWU_P10	TX_SWITCH	I2C1_SCL	UART0_RX	CMT_IRO	—	DTM_RX
39	28	PTC3	DISABLED	TSI0_CH15	PTC3/LLWU_P11	RX_SWITCH	I2C1_SDA	UART0_TX	—	—	DTM_TX
40	—	PTC4	DISABLED	TSI0_CH0	PTC4/LLWU_P12	—	EXTRG_IN	UART0_CTS_b	TPM1_CH0	—	—
41	—	PTC5	DISABLED	TSI0_CH1	PTC5/LLWU_P13	—	LPTMR0_ALT2	UART0_RTS_b	TPM1_CH1	—	—
42	—	PTC6	DISABLED	TSI0_CH2	PTC6/LLWU_P14	—	I2C1_SCL	UART0_RX	TPM2_CH0	—	—
43	—	PTC7	DISABLED	TSI0_CH3	PTC7/LLWU_P15	SPI0_PCS2	I2C1_SDA	UART0_TX	TPM2_CH1	—	—
44	29	VDD_1	VDD	—	—	—	—	—	—	—	—
45	30	PTC16	DISABLED	TSI0_CH4	PTC16/LLWU_P0	SPI0_SCK	I2C0_SDA	UART0_RTS_b	TPM0_CH3	—	—
46	31	PTC17	DISABLED	TSI0_CH5	PTC17/LLWU_P1	SPI0_SOUT	—	UART0_RX	—	—	DTM_RX

Table continues on the next page...

Table 2-1. KW40Z Pin Assignments (continued)

48 Laminate QFN	32 Laminate QFN	Pin Name[1]	DEFAULT	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
47	32	PTC18	DISABLED	TSI0_CH6	PTC18/LLWU_P2	SPI0_SIN	—	UART0_TX	—	—	DTM_TX
48	1	PTC19	DISABLED	TSI0_CH7	PTC19/LLWU_P3	SPI0_PCS0	I2C0_SCL	UART0_CTS_b	—	—	BLE_ACTIVE
49-64	33-41	Ground	NA	NA	NA	NA	NA	NA	NA	NA	NA

1. LLWU_Px signals are active in LLS/VLLSx power modes

2.3 KW40Z SoC Signal Descriptions

Signal Descriptions

Table 2-2. KW40Z SoC Signal Descriptions

Signal Name	Description	Type
ADC0_DM0	ADC Channel 0 Differential Input Negative	A
ADC0_DP0	ADC Channel 0 Differential Input Positive	A
ADC0_SE0	ADC Channel 0 Single-ended Input 0	A
ADC0_SE1	ADC Channel 0 Single-ended Input 1	A
ADC0_SE2	ADC Channel 0 Single-ended Input 2	A
ADC0_SE3	ADC Channel 0 Single-ended Input 3	A
ADC0_SE4	ADC Channel 0 Single-ended Input 4	A
ANT_A	TSM Trigger Output	D
ANT_B	TSM Trigger Output	D
BLE_ACTIVE	BLE Radio Active Indicator Output	D
CLKOUT	Internal Clocks Monitoring	D
CMP0_IN0	Comparator0 Input 0	D
CMP0_IN1	Comparator0 Input 0	D
CMP0_IN2	Comparator0 Input 0	D
CMP0_IN3	Comparator0 Input 0	D
CMP0_IN4	Comparator0 Input 0	D
CMP0_IN5	Comparator0 Input 0	D
CMP0_OUT	Comparator0 Output	D
CMT_IRO	Carrier Modulator Transmitter Infrared Output	A
DAC0_OUT	DAC0 Output	A
DCDC_CFG	DCDC Switch Mode Select	D

Table continues on the next page...

Table 2-2. KW40Z SoC Signal Descriptions (continued)

Signal Name	Description	Type
DCDC_GND	DCDC Switch Ground	G
DCDC_LN	DCDC Switch Inductor Input Negative	A
DCDC_LP	DCDC Switch Inductor Input Positive	A
DTM_RX	Direct Test Mode Receive	D
DTM_TX	Direct Test Mode Transmit	D
EXTAL_32M	32 MHz Crystal Input	A
EXTAL32K	32kHz Crystal Input	A
EXTRG_IN	TPM or ADC External Trigger Input	D
I2C0_SCL	I2C0 SCL	D
I2C0_SDA	I2C0 SDA	D
I2C1_SCL	I2C1 SCL	D
I2C1_SDA	I2C1 SDA	D
LPTMR0_ALT1	Low Power Timer0 ALT1	D
LPTMR0_ALT2	Low Power Timer0 ALT2	D
NMI_b	Non Maskable Interrupt Request	D
PSWITCH	DCDC Switch Enable	D
PTA0	GPIO Port A0	D
PTA1	GPIO Port A1	D
PTA16	GPIO Port A16	D
PTA17	GPIO Port A17	D
PTA18	GPIO Port A18	D
PTA19	GPIO Port A19	D
PTA2	GPIO Port A2	D
PTB0	GPIO Port B0	D
PTB1	GPIO Port B1	D
PTB16	GPIO Port B16	D
PTB17	GPIO Port B17	D
PTB18	GPIO Port B18	D
PTB2	GPIO Port B2	D
PTB3	GPIO Port B3	D
PTB4	GPIO Port B4	D
PTC0	GPIO Port C0	D
PTC1	GPIO Port C1	D
PTC16	GPIO Port C16	D
PTC17	GPIO Port C17	D
PTC18	GPIO Port C18	D
PTC19	GPIO Port C19	D
PTC2	GPIO Port C2	D
PTC3	GPIO Port C3	D

Table continues on the next page...

Table 2-2. KW40Z SoC Signal Descriptions (continued)

Signal Name	Description	Type
PTC4	GPIO Port C4	D
PTC5	GPIO Port C5	D
PTC6	GPIO Port C6	D
PTC7	GPIO Port C7	D
RESET_b	MCU Reset	D
RF_N	RF Negative Port	A
RF_P	RF Positive Port	A
RTC_CLKOUT	RTC Clock Out	D
RX_SWITCH	TSM Trigger Output	D
SPI0_PCS0	SPI0 PCS0	D
SPI0_PCS1	SPI0 PCS1	D
SPI0_PCS2	SPI0 PCS2	D
SPI0_SCK	SPI0 Clock	D
SPI0_SIN	SPI0 Input	D
SPI0_SOUT	SPI0 Output	D
SPI1_PCS0	SPI1 PCS0	D
SPI1_SCK	SPI1 Clock	D
SPI1_SIN	SPI1 Input	D
SPI1_SOUT	SPI1 Output	D
SWD_CLK	Serial Wire Debug Clock	D
SWD_DIO	Serial Wire Debug Data Input and Output	D
TPM_CLKIN0	TPM Clock Input 0	D
TPM_CLKIN1	TPM Clock Input 1	D
TPM0_CH0	TPM0 Channel 0	D
TPM0_CH1	TPM0 Channel 1	D
TPM0_CH2	TPM0 Channel 2	D
TPM0_CH3	TPM0 Channel 3	D
TPM1_CH0	TPM1 Channel 0	D
TPM1_CH1	TPM1 Channel 1	D
TPM2_CH0	TPM2 Channel 0	D
TPM2_CH1	TPM2 Channel 1	D
TSI0_CH0	TSI0 Channel 0	A
TSI0_CH1	TSI0 Channel 1	A
TSI0_CH10	TSI0 Channel 10	A
TSI0_CH11	TSI0 Channel 11	A
TSI0_CH12	TSI0 Channel 12	A
TSI0_CH13	TSI0 Channel 13	A
TSI0_CH14	TSI0 Channel 14	A
TSI0_CH15	TSI0 Channel 15	A

Table continues on the next page...

Table 2-2. KW40Z SoC Signal Descriptions (continued)

Signal Name	Description	Type
TSI0_CH2	TSI0 Channel 2	A
TSI0_CH3	TSI0 Channel 3	A
TSI0_CH4	TSI0 Channel 4	A
TSI0_CH5	TSI0 Channel 5	A
TSI0_CH6	TSI0 Channel 6	A
TSI0_CH7	TSI0 Channel 7	A
TSI0_CH8	TSI0 Channel 8	A
TSI0_CH9	TSI0 Channel 9	A
TX_SWITCH	TSM Trigger Output	D
UART0_CTS_b	UART0 Clear To Send	D
UART0_RTS_b	UART0 Request To Send	D
UART0_RX	UART0 Receive	D
UART0_TX	UART0 Transmit	D
VDCDC_IN	DCDC Switch Main Supply	P
VDD_0	Power Supply 0	P
VDD_1	Power Supply 1	P
VDD_1P45OUT_PMCIN	DCDC Pulsed Output 1.45V Regulated Output or PMC Input when DCDC in bypass mode	P
VDD_1P8OUT	DCDC Pulsed 1.8V Regulated Output	P
VDD_RF1_1P45	Power Supply - RF1	P
VDD_RF2_1P45	Power Supply - RF2	P
VDD_XTAL	32MHz Crystal Oscillator Power Supply	P
VDDA	Power Supply - Analog	P
VREFH	ADC reference voltage	P
VSSA	ADC ground	G
XTAL_32M	32 MHz Crystal Input	A
XTAL32K	32kHz Crystal Input	A

Legend

- A - Analog
- D - Digital
- P - Power Supply
- G - Ground

Chapter 3
Chip Configuration

This section provides details on the individual modules of the microcontroller.

3.1 Introduction

This chapter provides details on the individual modules of the microcontroller. It includes:

- Module block diagrams showing immediate connections within the device
- Specific module-to-module interactions not necessarily discussed in the individual module chapters
- Links for more information

3.2 Module to module interconnects

3.2.1 Module to Module Interconnects

The below table captures the Module to module interconnections for this device. KW4x closely follows the L-family definition for module to module interconnects.

Table 3-1. Module to Module Interconnects

Peripheral	Signal		to Peripheral	Use Case	Control	Comment
TPM1	CH0F, CH1F	to	ADC (Trigger)	ADC Triggering (A AND B)	SOPT7_ADCAL TTRGEN = 0	Ch0 is A, and Ch1 is B, selecting this ADC trigger is for supporting A and B triggering. In Stop and VLPS modes, the second

Table continues on the next page...

Table 3-1. Module to Module Interconnects (continued)

Peripheral	Signal		to Peripheral	Use Case	Control	Comment
						trigger must be set to >10us after the first trigger
TPMx	TOF	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field), SOPT7_ADC0PRETRGSEL to select A or B	—
LPTMR	Hardware trigger	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field), ADC0PRETRGSEL to select A or B	—
PIT CHx	TIF0, TIF1	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field), ADC0PRETRGSEL to select A or B	—
RTC	ALARM or SECONDS	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field) ADC0PRETRGSEL to select A or B	—
EXTRG_IN	EXTRG_IN	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field) ADC0PRETRGSEL to select A or B	—
CMP0	CMP0_OUT	to	ADC (Trigger)	ADC Triggering (A or B)	SOPT7_ADC0TRGSEL (4 bit field) ADC0PRETRGSEL to select A or B	—
Radio TSM	sar_adc_trig	to	ADC (Trigger)	ADC Triggering (A or B).	SOPT7_ADC0TRGSEL (4 bit field) ADC0PRETRGSEL to select A or B	This could provide a battery voltage or other ADC channel measurement reading at end of warmup or start of warmdown
CMP0	CMP0_OUT	to	LPTMR_ALT0	Count CMP events	LPTMR_CSR[TPS]	

Table continues on the next page...

Table 3-1. Module to Module Interconnects (continued)

Peripheral	Signal		to Peripheral	Use Case	Control	Comment
CMP0	CMP0_OUT	to	TPM1 CH0	Input capture	SOPT4[TPM1CH0SRC]	
CMP0	CMP0_OUT	to	TPM2 CH0	Input capture	SOPT4[TPM2CH0SRC]	
CMP0	CMP0_OUT	to	UART0_RX	IR interface	SOPT5[UART0RXSRC]	
LPTMR	Hardware trigger	to	CMP0	Low power triggering of the comparator	CMP_CR1[TRIM]	
LPTMR	Hardware trigger	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	—
TPMx	TOF	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	—
PIT CHx	TIF0, TIF1	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	If PIT is triggering the TPM, the TPM clock must be faster than Bus clock.
RTC	ALARM or SECONDS	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	—
EXTRG_IN	EXTRG_IN	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	—
CMP0	CMP0_OUT	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	
Radio TSM	sar_adc_trig[1]	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	
TPM1	Timebase	to	TPMx	TPM Global timebase input	TPMx_CONF[GTBEEN]	—
LPTMR	Hardware trigger	to	TSI	TSI triggering	TSI selects HW trigger	
UART0	UART0_TX	to	Modulated by TPM1 CH0	UART modulation	SOPT5_UART0TXSRC	—
UART0	UART0_TX	to	Modulated by TPM2 CH0	UART modulation	SOPT5_UART0TXSRC	—
PIT	TIF0	to	DAC	Advance DAC FIFO	DAC HWTRG Select	
PIT	TIF0	to	DMA CH0	DMA HW Trigger	DMA MUX register option	
PIT	TIF1	to	DMA CH1	DMA HW Trigger	DMA MUX register option	

1. This is the same TSM signal as shown above providing a trigger to the ADC. Triggering a TPM could provide a time-delayed offset for the TPM to trigger the ADC, or could be used for other purposes.

3.3 Core modules

3.3.1 ARM Cortex-M0+ core configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

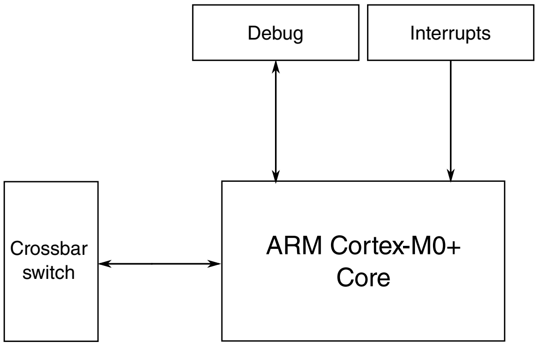


Figure 3-1. Core configuration

Table 3-2. Reference links to related information

Topic	Related module	Reference
Full description	ARM Cortex-M0+ core, r0p1	ARM Cortex-M0+ Technical Reference Manual,
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
System/instruction/data bus module	Crossbar switch	Crossbar switch
Debug	Serial wire debug (SWD)	Debug
Interrupts	Nested vectored interrupt controller (NVIC)	NVIC
Private Peripheral Bus	Miscellaneous control module (MCM)	MCM

3.3.1.1 ARM Cortex M0+ core

This device uses the r0p1 version of the ARM Cortex M0+ core.

The ARM Cortex M0+ parameter settings are as follows:

Table 3-3. ARM Cortex-M0+ parameter settings

Parameter	Value	Description
Arch Clock Gating	1 = Present	Implements architectural clock gating
DAP Slave Port Support	1	Supports any AHB debug access port (like the CM4 DAP)
DAP ROM Table Base	0xF000_2003	Base address for DAP ROM table
Endianess	0	Little endian control for data transfers
Breakpoints	2	Implements 2 breakpoints
Debug Support	1 = Present	—
Halt Event Support	1 = Present	—
I/O Port	1 = Present	Implements single-cycle ld/st accesses to special address space
IRQ Mask Enable	0x00000000	Assume (for now) all 32 IRQs are used (set if IRQ is disabled)
Debug Port Protocol	0 = SWD	SWD protocol, not JTAG
Core Memory Protection	0 = Absent	No MPU
Number of IRQs	32	Assume full NVIC request vector
Reset all registers	0 = Standard	Do not force all registers to be async reset
Multiplier	0 = Fast Mul	Implements single-cycle multiplier
Multi-drop Support	0 = Absent	Do not include serial wire support for multi-drop
System Tick Timer	1 = Present	Implements system tick timer (for CM4 compatibility)
DAP Target ID	0	—
User/Privileged	1 = Present	Implements processor operating modes
Vector Table Offset Register	1 = Present	Implements relocation of exception vector table
WIC Support	1 = Present	Implements WIC interface
WIC Requests	34	Exact number of wake-up IRQs is 34
Watchpoints	2	Implements two watchpoints

For details on the ARM Cortex-M0+ processor core, see the ARM website:arm.com.

3.3.1.2 Buses, Interconnects, and Interfaces

The ARM Cortex-M0+ core has two bus interfaces:

- single 32-bit AMBA-3 AHB-Lite system interface that provides connections to peripherals and all system memory, which includes flash and RAM.
- single 32-bit I/O port bus interfacing to the GPIO with 1-cycle loads and stores.

3.3.1.3 System Tick Timer

The CLKSOURCE bit in SysTick Control and Status register selects either the core clock (when CLKSOURCE = 1) or a divide-by-16 of the core clock (when CLKSOURCE = 0). Because the timing reference is a variable frequency, the TENMS bit in the SysTick Calibration Value Register is always zero.

3.3.1.4 Debug Facilities

This device supports standard ARM 2-pin SWD debug port.

3.3.1.5 Core Privilege Levels

The ARM documentation uses different terms than this document to distinguish between privilege levels.

If you see this term...	it also means this term...
Privileged	Supervisor
Unprivileged or user	User

3.3.2 Nested Vectored Interrupt Controller (NVIC) Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

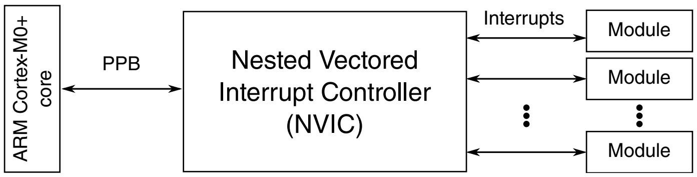


Figure 3-2. NVIC configuration

Table 3-4. Reference links to related information

Topic	Related module	Reference
Full description	Nested Vectored Interrupt Controller (NVIC)	ARM Cortex-M0+ Technical Reference Manual
System memory map		System memory map

Table continues on the next page...

Table 3-4. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock distribution
Power management		Power management
Private Peripheral Bus (PPB)	ARM Cortex-M0+ core	ARM Cortex-M0+ core

3.3.2.1 Interrupt priority levels

This device supports 4 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 2 bits. For example, IPR0 is shown below:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ3		0	0	0	0	0	0	IRQ2		0	0	0	0	0	0	IRQ1		0	0	0	0	0	0	IRQ0		0	0	0	0	0	0
W																																

3.3.2.2 Non-maskable interrupt

The non-maskable interrupt request to the NVIC is controlled by the external $\overline{\text{NMI}}$ signal. The pin the $\overline{\text{NMI}}$ signal is multiplexed on must be configured in order for the $\overline{\text{NMI}}$ function to generate the non-maskable interrupt request. $\overline{\text{NMI}}$ and DAC output share the same physical signal pin (PTB18). The $\overline{\text{NMI}}$ signal have a pull-up enabled and any external device connected to PTB18 will see a VDD voltage. Even if the FOPT is selected to disable the $\overline{\text{NMI}}$ signal, the pull-up will be enabled during Reset.

3.3.2.3 Interrupt channel assignments

The interrupt vector assignments are defined in the following table.

- Vector number — the value stored on the stack when an interrupt is serviced.
- IRQ number — non-core interrupt source count, which is the vector number minus 16.

The IRQ number is used within ARM's NVIC documentation.

Table 3-6. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
ARM Core System Handler Vectors					
0x0000_0000	0	—	—	ARM core	Initial Stack Pointer
0x0000_0004	1	—	—	ARM core	Initial Program Counter
0x0000_0008	2	—	—	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	—	—	ARM core	Hard Fault
0x0000_0010	4	—	—	—	—
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
0x0000_0034	13	—	—	—	—
0x0000_0038	14	—	—	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	—	—	ARM core	System tick timer (SysTick)
Non-Core Vectors					
0x0000_0040	16	0	0	DMA	DMA channel 0 transfer complete and error
0x0000_0044	17	1	0	DMA	DMA channel 1 transfer complete and error
0x0000_0048	18	2	0	DMA	DMA channel 2 transfer complete and error
0x0000_004C	19	3	0	DMA	DMA channel 3 transfer complete and error
0x0000_0050	20	4	1	—	—
0x0000_0054	21	5	1	FTFA	Command complete and read collision
0x0000_0058	22	6	1	PMC and DCDC	PMC: Low-voltage detect, low-voltage warning DCDC: PSWITCH interrupt
0x0000_005C	23	7	1	LLWU	Low Leakage Wakeup
0x0000_0060	24	8	2	I^{2}C0	
0x0000_0064	25	9	2		
0x0000_0068	26	10	2	SPI0	Single interrupt vector for all sources
0x0000_006C	27	11	2	TSI	
0x0000_0070	28	12	3	UART0	Status and error
0x0000_0074	29	13	3	TRNG	TRNG
0x0000_0078	30	14	3	CMT	
0x0000_007C	31	15	3	ADC0	
0x0000_0080	32	16	4	CMP0	

Table continues on the next page...

Table 3-6. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_0084	33	17	4	TPM0	
0x0000_0088	34	18	4	TPM1	
0x0000_008C	35	19	4	TPM2	
0x0000_0090	36	20	5	RTC	Alarm interrupt
0x0000_0094	37	21	5	RTC	Seconds interrupt
0x0000_0098	38	22	5	PIT	Single interrupt vector for all channels
0x0000_009C	39	23	5	LTC0	
0x0000_00A0	40	24	6	BTLL and RSIM	BTLL: combined Bluetooth LL interrupt RSIM: DSM exit interrupt
0x0000_00A4	41	25	6	DAC0	
0x0000_00A8	42	26	6	ZigBee	ZigBee interrupt sources
0x0000_00AC	43	27	6	MCG	
0x0000_00B0	44	28	7	LPTMR0	
0x0000_00B4	45	29	7	SPI1	SPI1
0x0000_00B8	46	30	7	Port control module	Pin detect (Port A)
0x0000_00BC	47	31	7	Port control module	Pin detect (Port B and Port C)

[1] Indicates the NVIC's interrupt source number.
[2] Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4

3.3.2.4 Serialization of memory operations when clearing interrupt flags

When clearing flags associated with an interrupt source in the interrupt service routine (ISR), the flag must be read back before exiting the ISR to ensure the flag is cleared. Otherwise, the interrupt could still be pending, causing the ISR to be entered again inadvertently.

3.3.3 Asynchronous wake-up interrupt controller (AWIC) configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

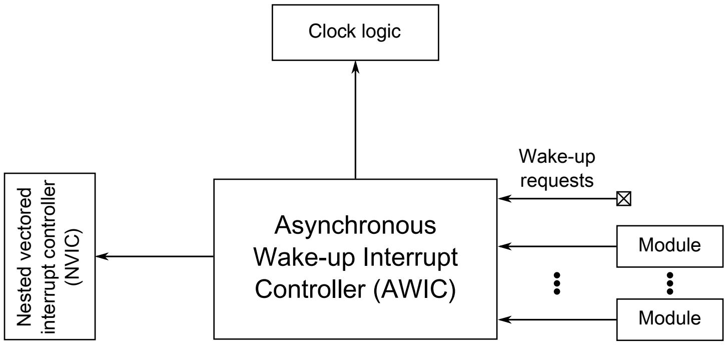

Clock logic
Wake-up requests
Nested vectored interrupt controller (NVIC)
Asynchronous Wake-up Interrupt Controller (AWIC)
Module
Module

Figure 3-3. Asynchronous wake-up interrupt controller configuration

Table 3-7. Reference links to related information

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Nested vectored interrupt controller (NVIC)	NVIC
Wake-up requests		AWIC wake-up sources

3.3.3.1 Wake-up sources

The device uses the following internal and external inputs to the AWIC module.

Table 3-8. AWIC stop wake-up sources

Wake-up source	Description
Available system resets	$\overline{\text{RESET}}$ pin when LPO is its clock source
Low-voltage detect	Mode Controller
Low-voltage warning	Mode Controller
DCDC	PSWITCH pin edge detect
Pin interrupts	Port control module - Any enabled pin interrupt is capable of waking the system
ADC	The ADC is functional when using internal clock source
CMP0	Interrupt in normal or trigger mode
Radio	Bluetooth LL \802.15.4 \ Radio interrupts
I^{2}Cx	Address match wakeup
UART	Any interrupt provided clock remains enabled
RTC	Alarm or seconds interrupt
TSI	Any interrupt
NMI	NMI pin
TPMx	Any interrupt provided clock remains enabled

Table continues on the next page...

Table 3-8. AWIC stop wake-up sources (continued)

Wake-up source	Description
LPTMR	Any interrupt provided clock remains enabled

3.4 System modules

3.4.1 SIM configuration

This section summarizes how the module has been configured in the chip.

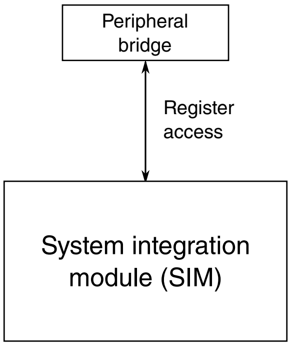


Figure 3-4. SIM configuration

Table 3-9. Reference links to related information

Topic	Related module	Reference
Full description	SIM	SIM
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management

3.4.2 System mode controller (SMC) configuration

This section summarizes how the module has been configured in the chip.

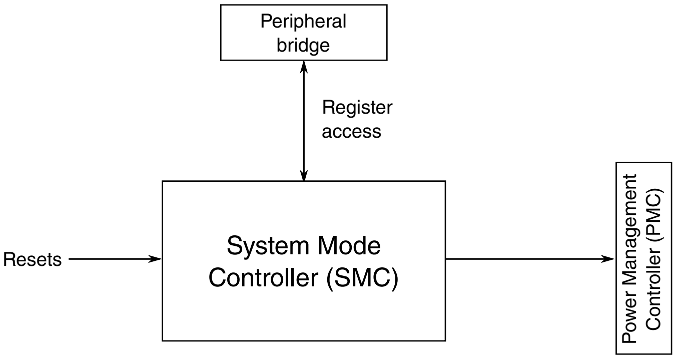

Peripheral bridge
Register access
Resets
System Mode Controller (SMC)
Power Management Controller (PMC)

Figure 3-5. System mode controller configuration

Table 3-10. Reference links to related information

Topic	Related module	Reference
Full description	System mode controller (SMC)	SMC
System memory map	—	System memory map
Power management	—	Power management
—	Power management controller (PMC)	PMC
	Low-leakage wakeup unit (LLWU)	LLWU
—	Reset control module (RCM)	Reset

3.4.3 PMC configuration

This section summarizes how the module has been configured in the chip.

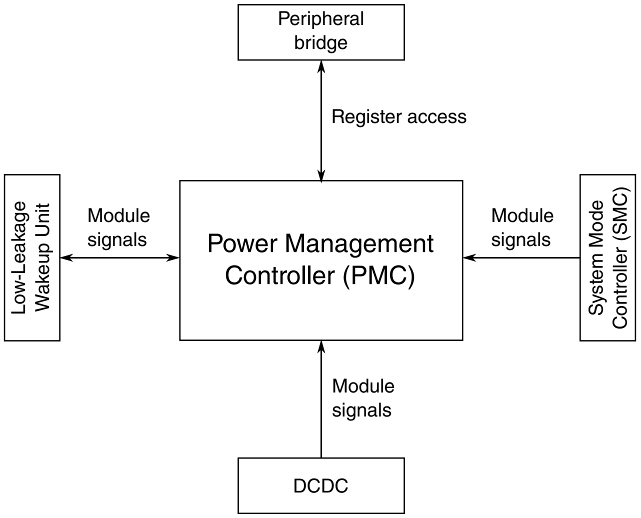


Figure 3-6. PMC configuration

Table 3-11. Reference links to related information

Topic	Related module	Reference
Full description	PMC	PMC
System memory map	—	System memory map
Power management	—	Power management
Full description	System mode controller (SMC)	System Mode Controller
	Low-leakage wakeup unit (LLWU)	LLWU
—	Reset control module (RCM)	Reset

3.4.4 DCDC configuration

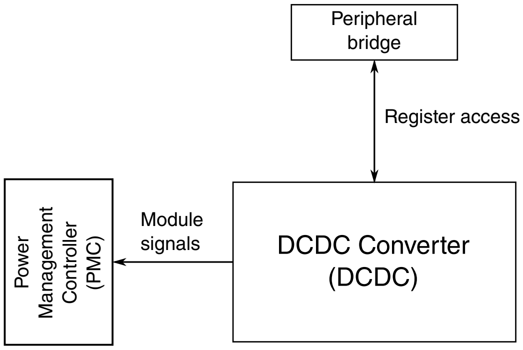


Figure 3-7. DCDC configuration

Table 3-12. Reference links to related information

Topic	Related module	Reference
Full description	DCDC	DCDC
System memory map	—	System memory map
Power management	—	Power management

3.4.5 Low-Leakage Wake-up Unit (LLWU) Configuration

This section summarizes how the module has been configured in the chip.

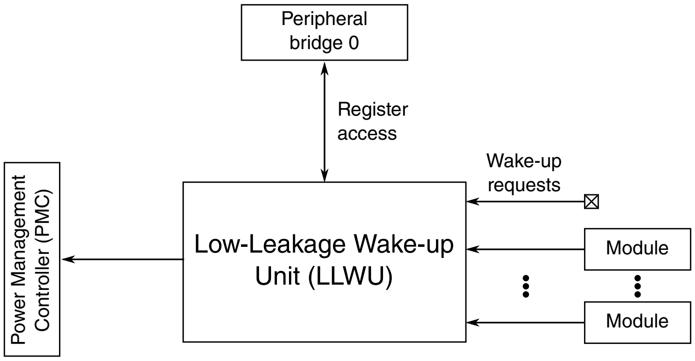

Peripheral bridge 0
Register access
Wake-up requests
☒
Power Management Controller (PMC)
Low-Leakage Wake-up Unit (LLWU)
Module
Module

Figure 3-8. Low-Leakage Wake-up Unit configuration

Table 3-13. Reference links to related information

Topic	Related module	Reference
Full description	LLWU	LLWU
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management chapter

Table continues on the next page...

Table 3-13. Reference links to related information (continued)

Topic	Related module	Reference
	Power Management Controller (PMC)	Power Management Controller (PMC)
	System Mode Controller (SMC)	System Mode Controller
Wake-up requests		LLWU wake-up sources

3.4.5.1 LLWU interrupt

NOTE

Do not mask the LLWU interrupt when in LLS mode. Masking the interrupt prevents the device from exiting stop mode when a wakeup is detected.

3.4.5.2 Wake-up Sources

The device uses the following internal peripheral and external pin inputs as wakeup sources to the LLWU module. LLWU_Px are external pin inputs, and LLWU_M0IF-M7IF are connections to the internal peripheral interrupt flags.

NOTE

In addition to the LLWU wakeup sources, the device also wakes from low power modes when NMI or RESET pins are enabled and the respective pin is asserted.

Table 3-14. LLWU Wakeup Sources

LLWU Inputs	Module source or pin name
LLWU_P0	PTC16[1]
LLWU_P1	PTC17[1]
LLWU_P2	PTC18[1]
LLWU_P3	PTC19[1]
LLWU_P4	PTA16[1]
LLWU_P5	PTA17[1]
LLWU_P6	PTA18[1]
LLWU_P7	PTA19[1]
LLWU_P8	PTB0[1]
LLWU_P9	PTC0[1]
LLWU_P10	PTC2[1]

Table continues on the next page...

Table 3-14. LLWU Wakeup Sources (continued)

LLWU Inputs	Module source or pin name
LLWU_P11	PTC3[1]
LLWU_P12	PTC4[1]
LLWU_P13	PTC5[1]
LLWU_P14	PTC6[1]
LLWU_M0IF	LPTMR0
LLWU_M1IF	CMP0
LLWU_M2IF	BTLL sysclk_req[2]
LLWU_M3IF	DCDC[3]
LLWU_M4IF	TSI
LLWU_M5IF	RTC Alarm
LLWU_M6IF	Reserved
LLWU_M7IF	RTC Seconds

1. When the DCDC is operating in buck or boost mode, a pin wake up event will also set the ISF (interrupt status flag) of the associated pin when exiting from LLS mode. The application should ignore this flag and simply clear it in the interrupt service routine used to service the LLS wake up event.
2. This wakeup interrupt signal is generated by the RSIM based on the rising edge of sysclk_req signal from the BTLL
3. This is for the PSWITCH interrupt

3.4.6 MCM configuration

This section summarizes how the module has been configured in the chip.

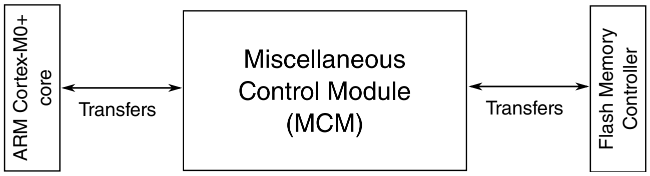


Figure 3-9. MCM configuration

Table 3-15. Reference links to related information

Topic	Related module	Reference
Full description	Miscellaneous control module (MCM)	MCM
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Private peripheral bus (PPB)	ARM Cortex-M0+ core	ARM Cortex-M0+ core
Transfer	Flash memory controller	Flash memory controller

3.4.7 Crossbar-light switch configuration

This section summarizes how the module has been configured in the chip.

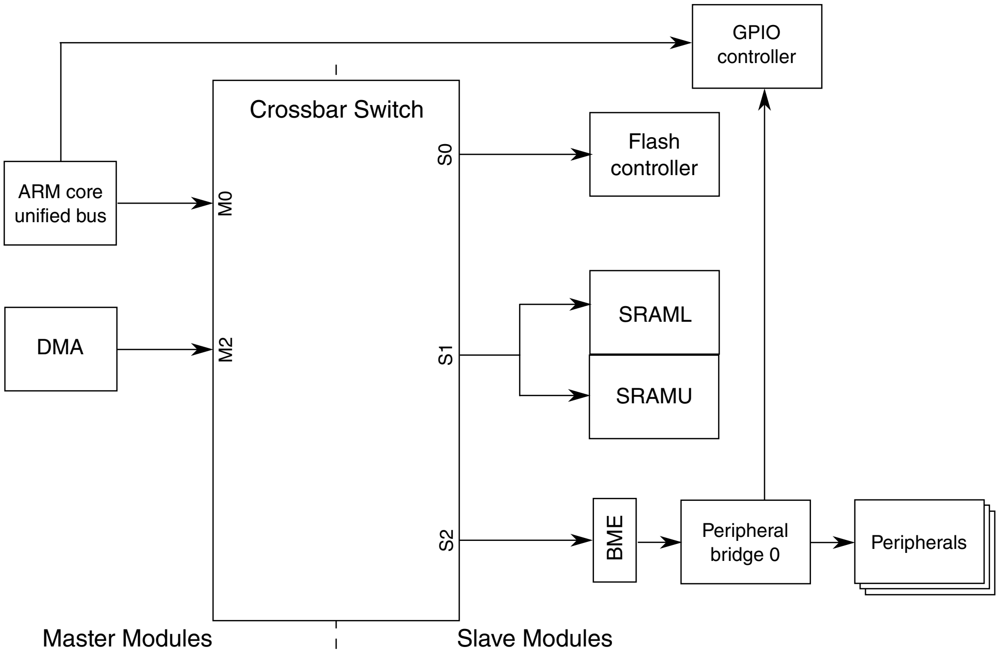


Figure 3-10. Crossbar-light switch integration

Table 3-16. Reference links to related information

Topic	Related module	Reference
Full description	Crossbar switch	Crossbar switch
System memory map	-	System memory map
Clocking	-	Clock distribution
Crossbar switch master	ARM Cortex-M0+ core	ARM Cortex-M0+ core
Crossbar switch master	DMA controller	DMA controller
Crossbar switch slave	Flash memory controller	Flash memory controller
Crossbar switch slave	SRAM controller	SRAM configuration
Crossbar switch slave	Peripheral bridge	Peripheral bridge
2-ported peripheral	GPIO controller	GPIO controller

3.4.7.1 Crossbar-Light Switch Master Assignments

The masters connected to the crossbar switch are assigned as follows:

Master module	Master port number
ARM core unified bus	0
DMA	2

3.4.7.2 Crossbar Switch Slave Assignments

This device contains 3 slaves connected to the crossbar switch.

The slave assignment is as follows:

Slave module	Slave port number
Flash memory controller	0
SRAM controller	1
Peripheral bridge 0	2

3.4.8 Peripheral bridge configuration

This section summarizes how the module has been configured in the chip.

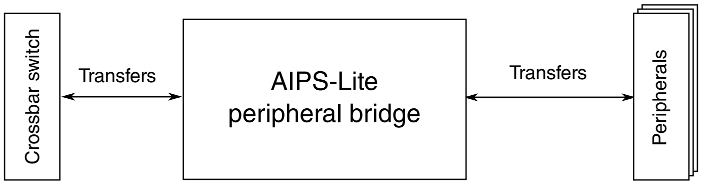


Figure 3-11. Peripheral bridge configuration

Table 3-17. Reference links to related information

Topic	Related module	Reference
Full description	Peripheral bridge (AIPS-Lite)	Peripheral bridge (AIPS-Lite)
System memory map	—	System memory map
Clocking	—	Clock distribution
Crossbar switch	Crossbar switch	Crossbar switch

3.4.8.1 Number of peripheral bridges

This device contains one peripheral bridge.

3.4.8.2 Memory maps

The peripheral bridges are used to access the registers of most of the modules on this device. See AIPS0 Memory Map for the memory slot assignment for each module.

3.4.9 DMA request multiplexer configuration

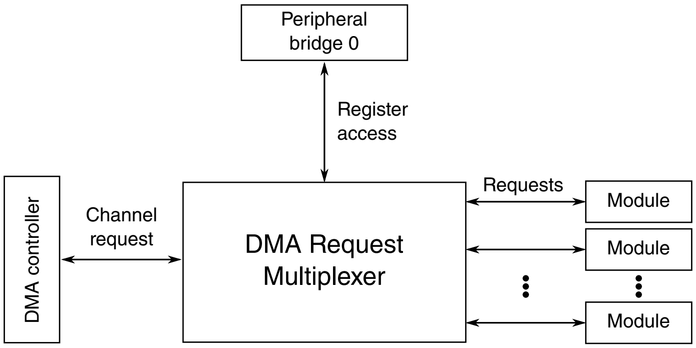


Figure 3-12. DMA request multiplexer configuration

Table 3-18. Reference links to related information

Topic	Related module	Reference
Full description	DMA request multiplexer	DMA Mux
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Channel request	DMA controller	DMA Controller
Requests		DMA request sources

3.4.9.1 DMA MUX Request Sources

This device includes a DMA request mux that allows up to 63 DMA request signals to be mapped to any of the 4 DMA channels. Because of the mux there is no hard correlation between any of the DMA request sources and a specific DMA channel. Some of the modules support Asynchronous DMA operation as indicated by the last column in the following DMA source assignment table.

Table 3-19. DMA request sources - MUX 0

Source number	Source module	Source description	Async DMA capable
0	—	Channel disabled[1]	
1	Reserved	Not used	
2	UART0(LPUART)	Receive	Yes
3	UART0(LPUART)	Transmit	Yes
4	Reserved	—	
5	Reserved	—	
6	Reserved	—	
7	Reserved	—	
8	Reserved	—	
9	Reserved	—	
10	Reserved	—	
11	Reserved	-	
12	Reserved	—	
13	Reserved	—	
14	Reserved	—	
15	Reserved	—	
16	SPI0	Receive	
17	SPI0	Transmit	
18	SPI1	Receive	
19	SPI1	Transmit	
20	AESA	Input FIFO	
21	AESA	Output FIFO	
22	I^2C0	—	
23	I^2C1	—	
24	TPM0	Channel 0	Yes
25	TPM0	Channel 1	Yes
26	TPM0	Channel 2	Yes
27	TPM0	Channel 3	Yes
28	Reserved	—	
29	Reserved	—	
30	Reserved	—	
31	Reserved	—	
32	TPM1	Channel 0	Yes
33	TPM1	Channel 1	Yes
34	TPM2	Channel 0	Yes
35	TPM2	Channel 1	Yes
36	Reserved	—	
37	Reserved	—	
38	Reserved	—	

Table continues on the next page...

Table 3-19. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description	Async DMA capable
39	Reserved	—	
40	ADC0	—	Yes
41	Reserved	—	
42	CMP0	—	Yes
43	Reserved	—	
44	Reserved	—	
45	DAC0	—	
46	Reserved	—	
47	CMT	—	
48	Reserved	—	
49	Port control module	Port A	Yes
50	Port control module	Port B	Yes
51	Port control module	Port C	Yes
52	Reserved	—	
53	Reserved	—	
54	TPM0	Overflow	Yes
55	TPM1	Overflow	Yes
56	TPM2	Overflow	Yes
57	TSI	—	Yes
58	Reserved	—	
59	Reserved	—	
60	DMA MUX	Always enabled	
61	DMA MUX	Always enabled	
62	DMA MUX	Always enabled	
63	DMA MUX	Always enabled	

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

3.4.9.2 DMA transfers via PIT trigger

The PIT module can trigger a DMA transfer on the first two DMA channels. The assignments are detailed at PIT/DMA Periodic Trigger Assignments .

3.4.10 DMA Controller Configuration

This section summarizes how the module has been configured in the chip.

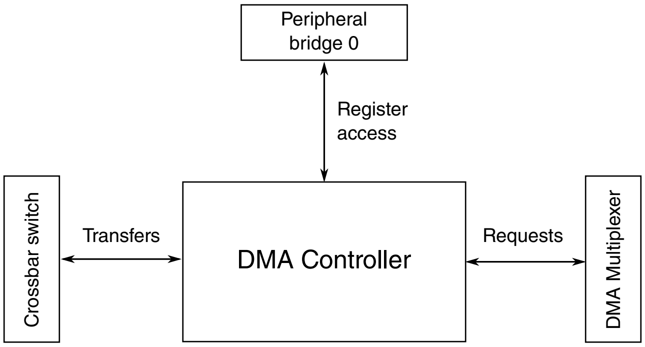


Figure 3-13. DMA Controller configuration

Table 3-20. Reference links to related information

Topic	Related module	Reference
Full description	DMA controller	DMA controller
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Crossbar switch	Crossbar switch	Crossbar switch
Requests		DMA request sources

3.4.11 Computer operating properly (COP) watchdog configuration

This section summarizes how the module has been configured in the chip.

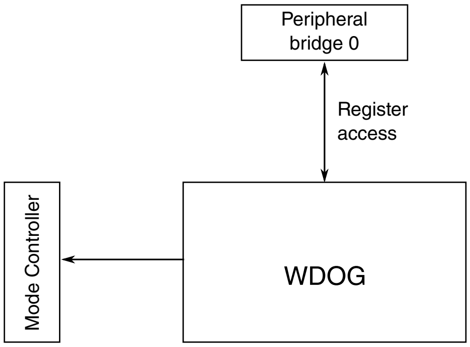


Figure 3-14. COP watchdog configuration

Table 3-21. Reference links to related information

Topic	Related module	Reference
Clocking	—	Clock distribution
Power management	—	Power management

Table continues on the next page...

Table 3-21. Reference links to related information (continued)

Topic	Related module	Reference
Programming model	System integration module (SIM)	SIM

3.4.11.1 COP clocks

The multiple clock inputs for the COP are:

- 1 kHz (LPO) clock
- bus clock
- MCGIRCLK
- OSCERCLK

3.4.11.2 COP watchdog operation

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), the application software must reset the COP counter periodically. If the application program gets lost and fails to reset the COP counter before it times out, a system reset is generated to force the system back to a known starting point.

After any reset, the COP watchdog is enabled. If the COP watchdog is not used in an application, it can be disabled by clearing SIM_COPC[COPT].

The COP counter is reset by writing 0x55 and 0xAA (in that order) to the address of the SIM's Service COP (SRVCOP) register during the selected timeout period. Writes do not affect the data in the SRVCOP register. As soon as the write sequence is complete, the COP timeout period is restarted. If the program fails to perform this restart during the timeout period, the microcontroller resets. Also, if any value other than 0x55 or 0xAA is written to the SRVCOP register, the microcontroller immediately resets.

SIM_COPC[COPCLKS] and SIM_COPCTRL[COPCLKSEL] select the timeout duration and clock source used for the COP timer. The clock source options are either the bus clock, MCGIRCLK, OSCERLK, or the internal 1 kHz (LPO) clock source. With each clock source, the associated timeouts are controlled by SIM_COPC[COPT] and SIM_COPC[COPCLKS]. The following table summarizes the control functions of

SIM_COPCTRL[COPCLKS] and SIM_COPC[COPCLKSEL] and SIM_COPC[COPT] fields. The COP watchdog defaults to operation from the 1 kHz clock source and the longest timeout is 2^{10} cycles.

Table 3-22. COP configuration options

Control bits			Clock source	COP window opens (SIM_COPC[COPW]=1)	COP overflow count
SIM_COPC[COPCLKSEL]	SIM_COPC[COPCLKS]	SIM_COPC[COPT]			
N/A	N/A	00	N/A	N/A	COP is disabled.
00	0	01	1 kHz	N/A	2^{5} cycles (32ms)
	1			6,144 cycles	2^{13} cycles (8192ms)
00	0	10	1 kHz	N/A	2^{8} cycles (256ms)
	1			49,152 cycles	2^{16} cycles (65536ms)
00	0	11	1 kHz	N/A	2^{10} cycles (1024 ms)
	1			196,608 cycles	2^{18} cycles (262144ms)
01	0	01	MCGIRCLK	N/A	2^{5} cycles
	1			6,144 cycles	2^{13} cycles
01	0	10	MCGIRCLK	N/A	2^{8} cycles
	1			49,152 cycles	2^{16} cycles
01	0	11	MCGIRCLK	N/A	2^{10} cycles
	1			196,608 cycles	2^{18} cycles
10	0	01	OSCERCLK	N/A	2^{5} cycles
	1			6,144 cycles	2^{13} cycles
10	0	10	OSCERCLK	N/A	2^{8} cycles
	1			49,152 cycles	2^{16} cycles
10	0	11	OSCERCLK	N/A	2^{10} cycles
	1			196,608 cycles	2^{18} cycles
11	0	01	bus	N/A	2^{5} cycles
	1			6,144 cycles	2^{13} cycles
11	0	10	bus	N/A	2^{8} cycles
	1			49,152 cycles	2^{16} cycles
11	0	11	bus	N/A	2^{10} cycles
	1			196,608 cycles	2^{18} cycles

After the long timeout (COPCLKS = 1) is selected, windowed COP operation is available by setting SIM_COPC[COPW]. In this mode, writes to SIM_SRVCOP to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write immediately resets the chip. When the short timeout (COPCLKS =0) is selected, windowed COP operation is not available.

The COP counter is initialized by the first writes to SIM_COPC and after any system reset. Subsequent writes to SIM_COPC have no effect on COP operation. Even if an application uses the reset default settings of SIM_COPC[COPT], SIM_COPC[COPCLKS], SIM_COPC[COPCLKSEL], and SIM_COPC[COPW] fields, the user should write to the write-once SIM_COPC register during reset initialization to lock in the settings. This approach prevents accidental changes if the application program becomes lost.

The write to SIM_SRVCOP that services (clears) the COP counter should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails.

If the selected clock is not the 1 kHz clock source, the COP counter does not increment while the microcontroller is in Debug mode or while the system is in Stop (including VLPS or LLS) mode. The COP counter resumes when the microcontroller exits Debug or Stop mode.

The COP counter is re-initialized to 0 upon entry to either Debug mode or Stop (including VLPS or LLS) mode. The counter begins from 0 upon exit from Debug mode or Stop mode.

The COP counter can also be configured to continue incrementing during Debug mode or Stop (including VLPS) mode if either COPDBGEN or COPSTPEN are set respectively. When the selected clock is the bus clock and COPSTEN bit is set, the COP counter cannot increment during Stop modes, however the COP counter is not reset to 0.

Regardless of the clock selected, the COP is disabled when the chip enters a VLLSx mode. Upon a reset that wakes the chip from the VLLSx mode, the COP is reinitialized and enabled as for any reset.

3.4.11.3 Clock Gating

This family of devices includes clock gating control for each peripheral, that is, the clock to each peripheral can explicitly be gated on or off, using clock-gate control bits in the SIM module.

3.5 Clock modules

3.5.1 MCG configuration

This section summarizes how the module has been configured in the chip.

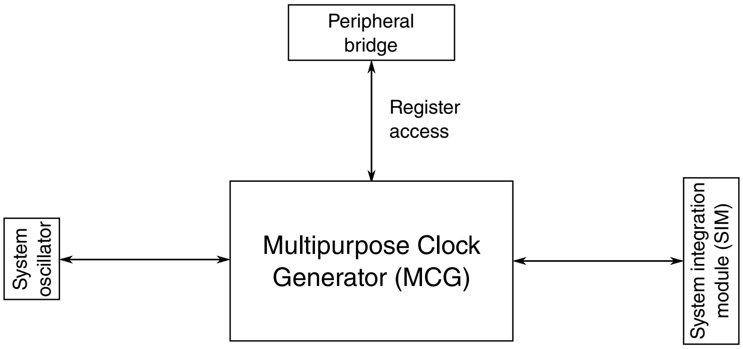

Peripheral bridge
Register access
System oscillator
Multipurpose Clock Generator (MCG)
System integration module (SIM)

Figure 3-15. MCG configuration

Table 3-23. Reference links to related information

Topic	Related module	Reference
Full description	MCG	MCG
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

3.5.1.1 MCG Instantiation Information

- OSC0 refers to the reference (32 MHz) oscillator
- MCG_C2[RANGE0] should be set to 1x for clock monitor function to work correctly with the RF oscillator
- MCG_C2[HGO0] is not used
- MCG_C2[EREFS0] is not used
- MCG_S[OSCINIT0] is tied to 0 in in KW40Z. Software can use the RSIM's CONTROL[RF_OSC_READY] bit to check on the status of the RF oscillator
- MCG_C7[OSCSEL]: The 32kHz oscillator should be enabled via the RTC before attempting to program the MCG to use the 32kHz oscillator

3.5.1.2 MCG FLL modes

The MCGFLLCLK frequency is limited to 48 MHz at maximum in this device. The digitally-controller oscillator (DCO) is limited to the two lowest range settings, that is, MCG_C4[DRST_DRS] must be set to either 0b00 or 0b01.

3.5.2 32kHz OSC Configuration

This section summarizes how the module has been configured in the chip.

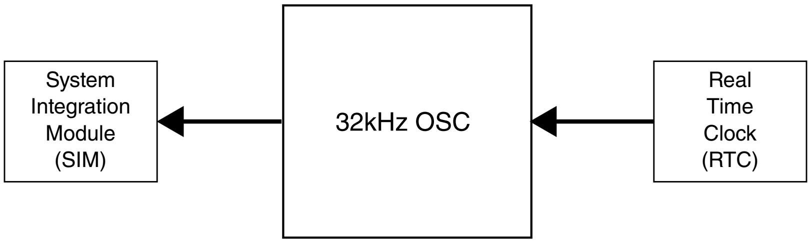


Figure 3-16. 32kHz OSC configuration

Table 3-24. Reference links to related information

Topic	Related module	Reference
Full description	32kHz Oscillator	OSC
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

3.5.2.1 32kHz OSC Instantiation Information

The 32kHz oscillator provides the clock source for the RTC. It supports 32kHz crystal with very low power consumption. Internal programmable capacitors are controlled by RTC module

3.5.3 Reference Oscillator Configuration

This section summarizes how the module has been configured in the chip.

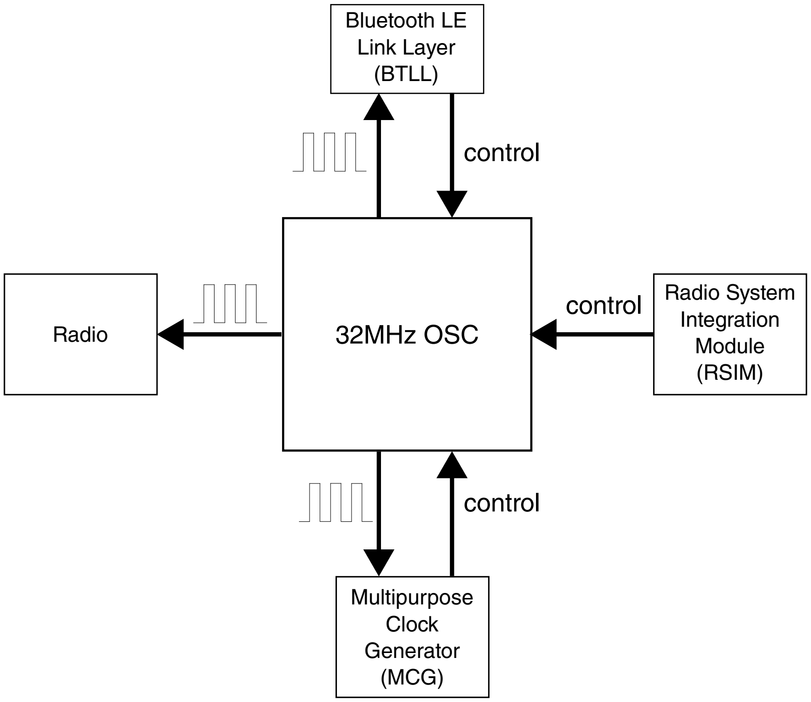


Figure 3-17. Reference OSC configuration

Table 3-25. Reference links to related information

Topic	Related module	Reference
Full description	Reference Oscillator	REFOSC
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

3.5.3.1 Reference Oscillator Instantiation Information

The reference oscillator is the master clock for the radio and the MCU core. There are multiple enable/disable sources for the oscillator, they are Bluetooth LE Link Layer (BTLL), Radio System Integration Module (RSIM) and Multipurpose Clock Generator (MCG).

The reference oscillator can be enabled manually for Run/Wait modes, and also optionally for Stop mode. It is done by setting the appropriate bits in the Radio System Integration Module (RSIM).

The MCG outputs a signal which enables the reference oscillator whenever OSCERCLK is selected for MCGOUTCLK or used as the reference for the FLL.

Note that use of the reference oscillator in VLPx modes is only possible when the DCDC is configured in continous mode.

3.6 Memories and memory interfaces

3.6.1 Flash Memory Configuration

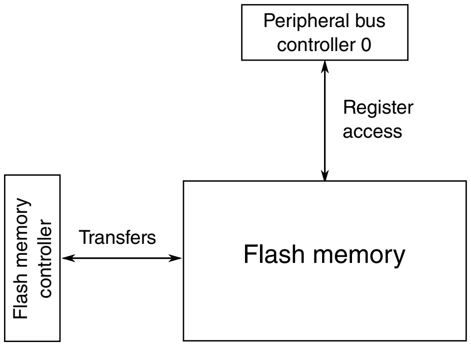


Figure 3-18. Flash memory configuration

Table 3-26. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory	Flash memory
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory controller	Flash memory controller
Register access	Peripheral bridge	Peripheral bridge

3.6.1.1 Flash Memory Sizes

The device contains 2 program flash blocks (one 128Kbytes, one 32Kbytes) consisting of 1 KB sectors.

3.6.1.2 Flash Memory Map

The flash memory and the flash registers are located at different base addresses as shown in the following figure. The base address for each is specified in System memory map.

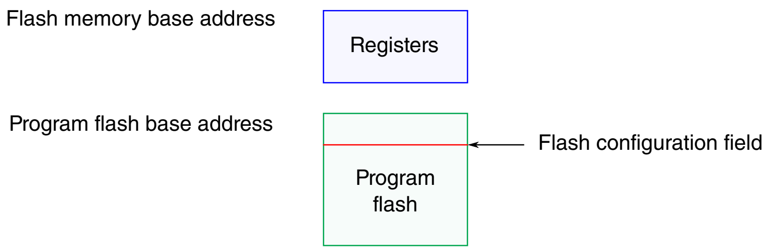

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program flash

Figure 3-19. Flash memory map

The on-chip Flash is implemented in a portion of the allocated Flash range to form a contiguous block in the memory map beginning at address 0x0000_0000. See Flash Memory Sizes for details of supported ranges.

Accesses to the Flash memory ranges outside the amount of Flash on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master.

3.6.1.3 Flash Security

How flash security is implemented on this device is described in Chip Security.

3.6.1.4 Flash Modes

The flash memory chapter defines two modes of operation - NVM normal and NVM special modes. On this device, The flash memory only operates in NVM normal mode. All references to NVM special mode should be ignored.

3.6.1.5 Erase All Flash Contents

In addition to software, the entire flash memory may be erased external to the flash memory via the SW-DP debug port by setting MDM-AP CONTROL[0]. MDM-AP STATUS[0] is set to indicate the mass erase command has been accepted. MDM-AP STATUS[0] is cleared when the mass erase completes.

3.6.1.6 FTFA_FOPT Register

The flash memory's FTFA_FOPT register allows the user to customize the operation of the MCU at boot time. See FOPT boot options for details of its definition.

3.6.2 Flash Memory Controller Configuration

This section summarizes how the module has been configured in the chip.

See MCM_PLACR register description for details on the reset configuration of the FMC.

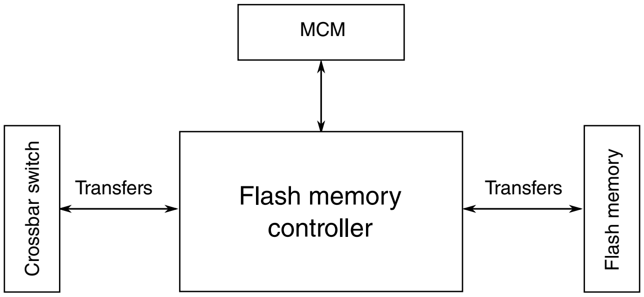


Figure 3-20. Flash memory controller configuration

Table 3-27. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory controller	Flash memory controller
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory	Flash memory
Transfers	Crossbar switch	Crossbar Switch
Register access	MCM	MCM

3.6.3 SRAM Configuration

This section summarizes how the module has been configured in the chip.

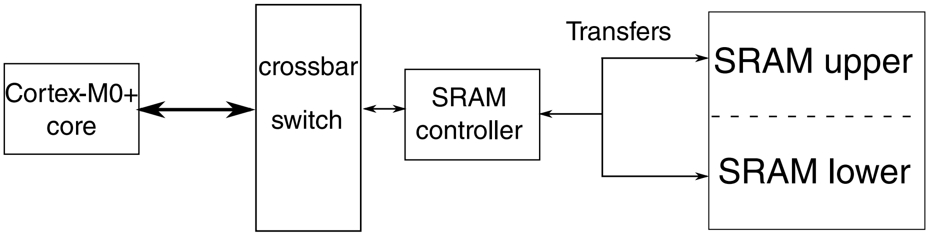


Figure 3-21. SRAM configuration

Table 3-28. Reference links to related information

Topic	Related module	Reference
Full description	SRAM	SRAM
System memory map		System memory map
Clocking		Clock Distribution
ARM Cortex-M0+ core		ARM Cortex-M0+ core

3.6.3.1 SRAM Sizes

The device contains 20 KB of SRAM which can be accessed by bus masters through the cross-bar switch.

All but 4 Kbytes of the SRAM will be power gated in LLS2 and VLLS2 modes.

3.6.3.2 SRAM Ranges

The device contains 20 KB of SRAM, split into two ranges, 1/5 (4Kbytes) is allocated SRAM_L (RAM1) and 4/5 (16Kbytes) is allocated to SRAM_U (RAM2). The first 4Kbytes of SRAM_U remains powered in LLS2 and VLLS2 modes. The ranges are as follows:

- SRAM_L: 0x1FFF_F000 – 0x1FFF_FFFF (4Kbytes). Powered off in LLS2 and VLLS2
- SRAM_U: 0x2000_0000 - 0x2000_3FFF (16Kbytes)
  - 0x2000_0000 - 0x2000_0FFF (4Kbytes). Remains powered in LLS2 and VLLS2
  - 0x2000_1000 - 0x2000_3FFF (12Kbytes). Powered off in LLS2 and VLLS2

This is illustrated in the following figure.

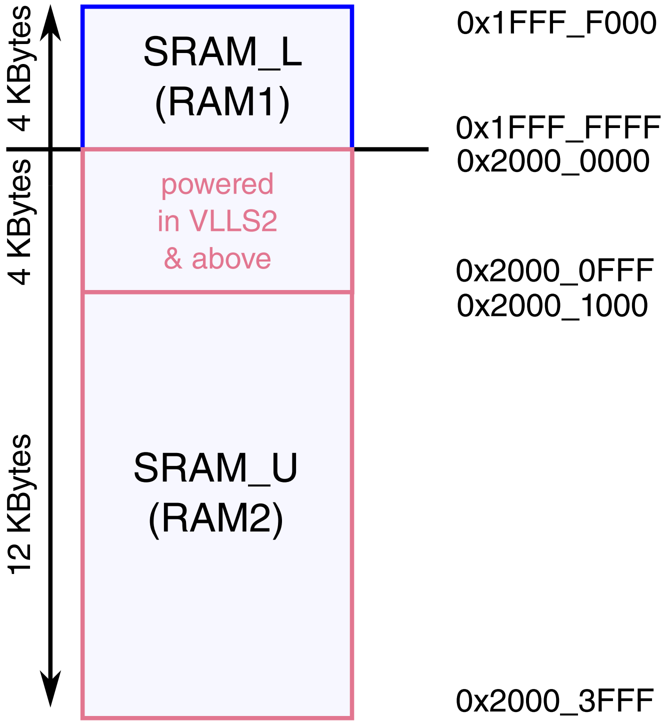


Figure 3-22. SRAM blocks memory map

3.6.3.3 SRAM retention in low power modes

In VLLS1 and VLLS0, no SRAM is retained. In LLS2 and VLLS2 modes, only 4Kbytes of SRAM (from 0x2000_0000 to 0x2000_0FFF) is retained. In all other low power modes the contents of the SRAM are retained.

3.6.4 System Register File Configuration

This section summarizes how the module has been configured in the chip.

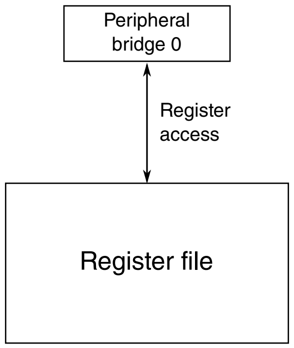


Figure 3-23. System Register file configuration

Table 3-29. Reference links to related information

Topic	Related module	Reference
Full description	Register file	Register file
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

3.6.4.1 System Register file

This device includes a 32-byte register file that is powered in all power modes.

Also, it retains contents during low-voltage detect (LVD) events and is only reset during a power-on reset.

3.7 Analog

3.7.1 Analog reference options

Several analog blocks have selectable reference voltages as shown in the table below. These options allow analog peripherals to share or have separate analog references. Care should be taken when selecting analog references to avoid cross talk noise.

NOTE

In the 32pin package, VREFH and VDDA are connected to the same package pin. Also in the 32pin package, VREFL and VSSA are tied to the ground flag. In the 48pin package VREFH and VDDA are separate package pins, while VREFL is connected to the VSSA package pin.

Table 3-30. Analog reference options

Module	Reference option	Comment/ Reference selection
16-bit SAR ADC	00 (V_{REF}) - VREFH/L 01 (V_{ALT}) - VDDA/VSSA	Selected by ADCx_SC2[REFSEL] bits
12-bit DAC	0 (DACREF_1) - VREFH 1 (DACREF_2) - VDDA	Selected by DACx_C0[DACRFS] bit
CMP with 6-bit DAC	0 (Vin1) - VREFH 1 (Vin2) - VDD	Selected by CMPx_DACCR[VRSEL] bit

3.7.2 16-bit SAR ADC configuration

This section summarizes how the module has been configured in the chip.

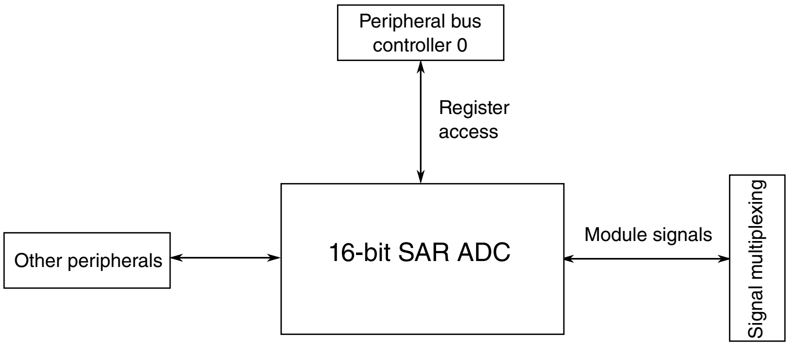


Figure 3-24. 16-bit SAR ADC configuration

Table 3-31. Reference links to related information

Topic	Related module	Reference
Full description	16-bit SAR ADC	16-bit SAR ADC
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.2.1 ADC Instantiation Information

This device contains one 16-bit successive approximation ADC.

The ADC supports both software and hardware triggers. The hardware trigger sources are listed in the Module-to-Module section. The ADC will have ADCSC1A and ADCSC1B status and control registers and the corresponding result registers.

3.7.2.2 DMA Support on ADC

Applications may require continuous sampling of the ADC that may have considerable load on the CPU. The ADC supports DMA request functionality for higher performance when the ADC is sampled at a very high rate. The ADC can trigger the DMA (via DMA req) on conversion completion.

3.7.2.3 ADC0 Connections/Channel Assignment

The ADC channel assignments are shown below. Note that not all channels are supported in all packages.

3.7.2.3.1 ADC0 Channel Assignment

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0	ADC0_DP0
00001	DAD1	Reserved	ADC0_SE1
00010	DAD2	Reserved	ADC0_SE2
00011	DAD3	Reserved	ADC0_SE3
00100	AD4	Reserved	ADC0_SE4/ 12-bit DAC0 Output
00101	AD5	Reserved	Reserved
00110	AD6	Reserved	Reserved
00111	AD7	Reserved	Reserved
01000	AD8	Reserved	Reserved
01001	AD9	Reserved	Reserved
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	Reserved
01101	AD13	Reserved	Reserved
01110	AD14	Reserved	Reserved
01111	AD15	Reserved	Reserved
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	Reserved

Table continues on the next page...

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
10010	AD18	Reserved	Reserved
10011	AD19	Reserved	Reserved
10100	AD20	Reserved	Reserved
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	Battery voltage[1]
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff)	Bandgap (S.E)[2]
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

[1] Battery voltage option is the internal connection to the DCDC's scaled battery voltage output. It is not main battery voltage supply.

[2] This is the PMC bandgap 1V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V_{BG}) specification.

3.7.2.4 ADC analog supply and reference connections

This device contains VREFH pin on the 48-pin package. In the 32-pin package, VREFH is tied to VDDA internally. The VREFL signal is internally connected to VSS for both packages.

This device includes dedicated VDDA. The VSSA signal internally connected to VSS.

V_{ALTH} and V_{ALTL} are internally connected to VDDA and VSSA respectively.

3.7.2.5 Alternate clock

For this device, the alternate clock is connected to the external reference clock (OSCERCLK).

3.7.3 CMP Configuration

This section summarizes how the module has been configured in the chip.

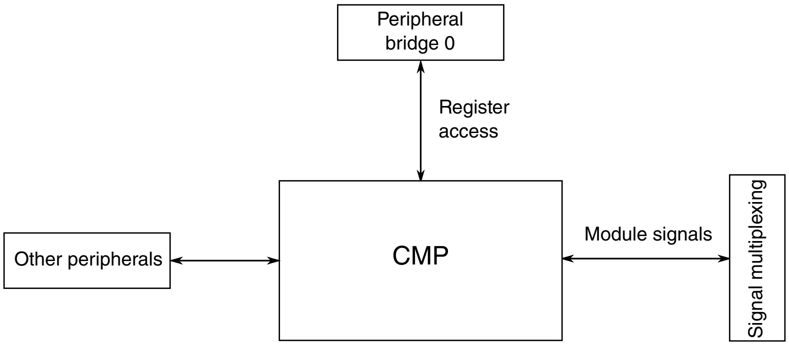


Figure 3-25. CMP configuration

Table 3-32. Reference links to related information

Topic	Related module	Reference
Full description	Comparator (CMP)	Comparator
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.3.1 CMP Instantiation Information

The device includes one high speed comparator with two 8-input multiplexors for both the inverting and non-inverting inputs of the comparator. Each CMP input channel connects to both muxes. See the channel assignment table for a summary of CMP input connections for this device.

The CMP also includes one 6-bit DAC with a 64-tap resistor ladder network, which provides a selectable voltage reference for applications where voltage reference is needed for internal connection to the CMP.

The CMP can be optionally on in all modes except VLLS0.

The CMP has several module to module interconnects in order to facilitate ADC triggering, timer triggering and UART IR interfaces. For complete details on the CMP module interconnects please refer to the Module-to-Module section.

The CMP does not support window compare function and CMP_CR1[WE] must always be written to 0. The sample function has limited functionality since the SAMPLE input to the block is not connected to a valid input. Usage of sample operation is limited to a divided version of the bus clock (CMP_CR1[SE] = 0).

Due to the pin number limitation, the CMP pass through mode is not supported by this device, so the CMPx_MUXCR[PSTM] must be left as 0.

3.7.3.2 CMP input connections

The following table shows the CMP input channel assignments.

Table 3-33. CMP input connections

Input Channel	Assignment
IN0	CMP0_IN0(pin)
IN1	CMP0_IN1(pin)
IN2	12-bit DAC0 reference / CMP0_IN2(pin)
IN3	CMP0_IN3(pin)
IN4	CMP0_IN4(pin)
IN5	CMP0_IN5(pin)
IN6	Bandgap
IN7	6-bit DAC0 reference

3.7.3.3 CMP external references

The 6-bit DAC sub-block supports selection of two references. For this device, the references are connected as follows:

- VREFH - V_{in1} input. When using VREFH, any ADC conversion using this same reference at the same time is negatively impacted.
- VDD - V_{in2} input

3.7.3.4 CMP trigger mode

The CMP and 6-bit DAC sub-block supports trigger mode operation when the CMP_CR1[TRIGM] is set. When trigger mode is enabled, the trigger event will initiate a compare sequence that must first enable the CMP and DAC prior to performing a CMP operation and capturing the output. In this device, control for this two staged sequencing is provided from the LPTMR. The LPTMR triggering output is always enabled when the LPTMR is enabled. The first signal is supplied to enable the CMP and DAC and is asserted at the same time as the TCF flag is set. The delay to the second signal that triggers the CMP to capture the result of the compare operation is dependent on the LPTMR configuration. In Time Counter mode with prescaler enabled, the delay is 1/2 Prescaler output period. In Time Counter mode with prescaler bypassed, the delay is 1/2 Prescaler clock period.

The delay between the first signal from LPTMR and the second signal from LPTMR must be greater than the Analog comparator initialization delay as defined in the device datasheet.

3.7.4 12-bit DAC configuration

This section summarizes how the module has been configured in the chip.

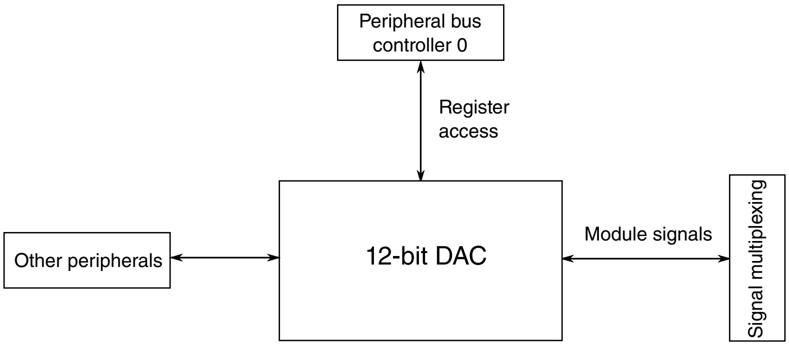


Figure 3-26. 12-bit DAC configuration

Table 3-34. Reference links to related information

Topic	Related module	Reference
Full description	12-bit DAC	12-bit DAC
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	-Signal multiplexing -

3.7.4.1 12-bit DAC Instantiation Information

This device contains one 12-bit digital-to-analog converter (DAC) with programmable reference generator output. The DAC includes a two word FIFO for DMA support.

3.7.4.2 12-bit DAC Output

The output of the DAC can be placed on an external pin or selected as an input to the analog comparator or ADC.

3.7.4.3 12-bit DAC Analog Supply Connections

This device includes a dedicated VDDA pin. A dedicated VSSA pin is not supported in the 32pin package.

3.7.4.4 12-bit DAC Reference

For this device VREFH and VDDA are selectable as the DAC reference. VREFH is connected to the DACREF_1 input and VDDA is connected to the DACREF_2 input. Use DACx_C0[DACRFS] control bit to select between these two options.

NOTE

> In the 32-pin package, the VREFH and VDDA share the same package pin.

Be aware that if the DAC and ADC use the same reference simultaneously, some degradation of ADC accuracy is to be expected due to DAC switching.

See also Analog References .

3.8 Radio

3.8.1 Radio module configuration

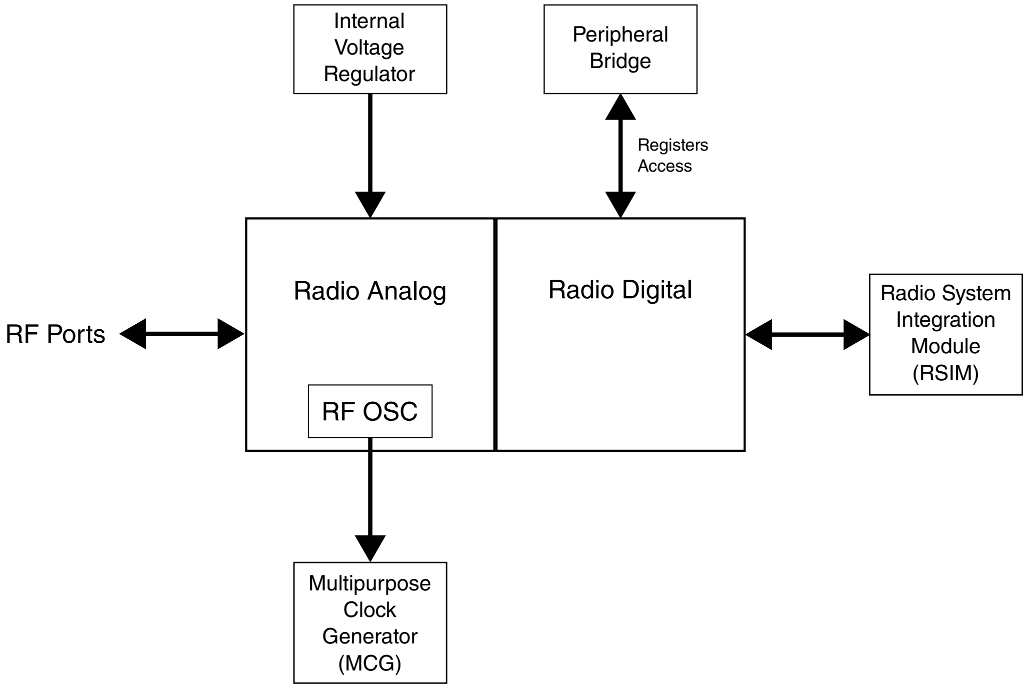

Internal Voltage Regulator
Peripheral Bridge
Registers Access
Radio Analog
Radio Digital
RF Ports
RF OSC
Radio System Integration Module (RSIM)
Multipurpose Clock Generator (MCG)

Figure 3-27. Radio configuration

Table 3-35. Reference links to related information

Topic	Related module	Reference
Full description	Radio	2.4 GHz Radio
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

3.8.1.1 Radio Overview

The radio block consists of radio analog and radio digital sections. The radio analog is comprised of the RF transmitter, receiver and the supporting analog functions. The RF reference oscillator is also included in the radio analog. It is the reference oscillator for KW40Z, and can be configured using the MCG to be the master clock for both the radio and the MCU core. The Radio System Integration Module (RSIM) provides system control for the radio block. The MCU core configures the radio through radio registers within the radio digital section. Software uses the radio to communicate by interfacing with the RF protocol link layers.

3.9 Timers

3.9.1 Timer/TPM Configuration

This section describes the configuration of the TPM timers in this device.

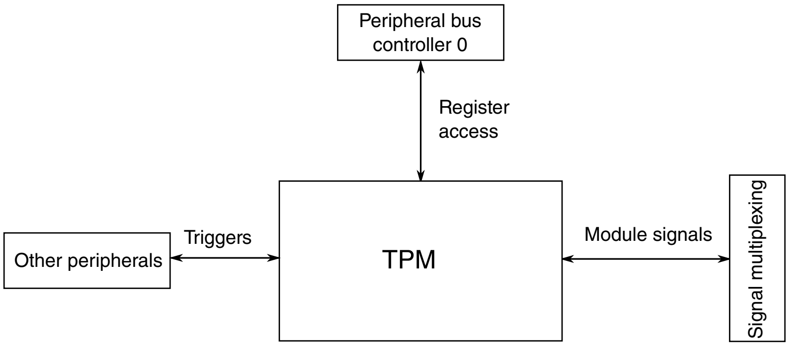


Figure 3-28. TPM configuration

Table 3-36. Reference links to related information

Topic	Related module	Reference
Full description	Timer/PWM Module	Timer/PWM Module
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.9.1.1 TPM Instantiation Information

This device contains three Low Power TPM modules (TPM). TPM0 is configured as shown in the table below with 4 channels and without quadrature decode function. TPM1 and TPM2 are configured for 2 channels each and include the quadrature decode function, to support the wireless mouse use case. All TPMs can be functional in Stop/VLPS mode; the clock source is either external or internal in Stop/VLPS mode.

Table 3-37. TPM configuration

Instance Name	Channels	Features
TPM0	4	• Basic TPM • NO quadrature decoder • Functional in Stop/VLPS mode

Table continues on the next page...

Table 3-37. TPM configuration (continued)

TPM1	2	• Basic TPM • Quadrature Decoder and filtering • Functional in Stop/VLPS
TPM2	2	• Basic TPM • Quadrature Decoder and filtering • Functional in Stop/VLPS

3.9.1.2 Clock Options

The TPM block is clocked from a clock that can be selected from OSCERCLK, MCGIRCLK, or MCGFLLCLK. The selected source is controlled by SIM_SOPT2[TPMSRC]. This is discussed in TPM Clocking.

Each TPM also supports an external clock mode (TPM_SC[CMOD]=1x) in which the counter increments after a synchronized (to the selected TPM clock source) rising edge detect of an external clock input. The available external clock (either TPM_CLKIN0 or TPM_CLKIN1) is selected by SIM_SOPT4[TPMxCLKSEL] control register. To guarantee valid operation the selected external clock must be less than half the frequency of the selected TPM clock source.

3.9.1.3 Trigger Options

Each TPM has a selectable trigger input source controlled by the TPMx_CONF[TRGSEL] field to use for starting the counter and/or reloading the counter. The options available are shown in the following table.

Table 3-38. TPM trigger options

TPMx_CONF[TRGSEL]	Selected source
0000	External trigger pin input (EXTRG_IN)
0001	CMP0 output
0010	Reserved
0011	Reserved
0100	PIT trigger 0
0101	PIT trigger 1
0110	Reserved
0111	Reserved
1000	TPM0 overflow
1001	TPM1 overflow
1010	TPM2 overflow

Table continues on the next page...

Table 3-38. TPM trigger options (continued)

TPMx_CONF[TRGSEL]	Selected source
1011	Reserved
1100	RTC alarm
1101	RTC seconds
1110	LPTMR trigger
1111	Radio TSM

These TPM trigger inputs are also described in Module to Module Interconnects.Each TPM also outputs channel and overflow triggers. The connections of these are described in the Module to Module interconnects section as well.

3.9.1.4 Global timebase

Each TPM has a global timebase feature controlled by TPMx_CONF[GTBEEN]. TPM1 is configured as the global time when this option is enabled.

3.9.1.5 Interrupts

The TPM has have multiple sources of interrupt. However, these sources are OR'd together to generate a single interrupt request per TPM module to the interrupt controller. When an interrupt occurs, read the TPM status registers (SC and STATUS) to determine the exact interrupt source.

3.9.2 PIT Configuration

This section summarizes how the module has been configured in the chip.

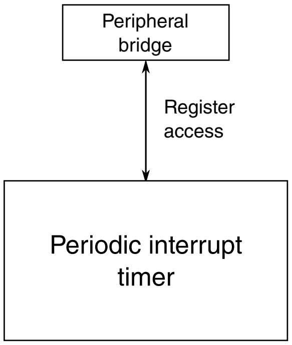


Figure 3-29. PIT configuration

Table 3-39. Reference links to related information

Topic	Related module	Reference
Full description	PIT	PIT
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

3.9.2.1 PIT/DMA Periodic Trigger Assignments

The PIT generates periodic trigger events to the DMA channel mux as shown in the table below.

Table 3-40. PIT channel assignments for periodic DMA triggering

PIT Channel	DMA Channel Number
PIT Channel 0	DMA Channel 0
PIT Channel 1	DMA Channel 1

3.9.2.2 PIT/ADC Triggers

PIT triggers are selected as ADCx trigger sources using the SOPT7[ADCxTRGSEL] bits in the SIM module. For more details, refer to SIM chapter.

3.9.2.3 PIT/FTM Triggers

PIT triggers are selected as FTMx trigger sources using the FTMx_CONF[TRGSEL] bits in the FTM module. For more details, refer to TPM chapter.

3.9.2.4 PIT/DAC Triggers

PIT Channel 0 is configured as the DAC hardware trigger source. For more details, refer to DAC chapter.

3.9.3 Low-power timer configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

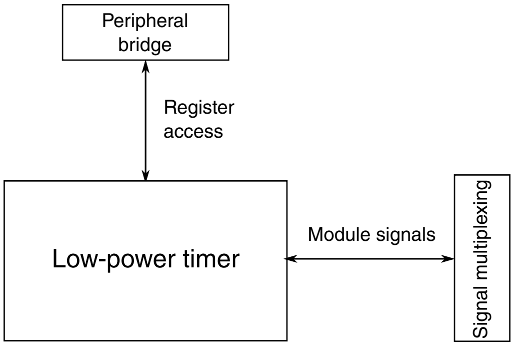


Figure 3-30. LPT configuration

Table 3-41. Reference links to related information

Topic	Related module	Reference
Full description	Low-power timer	Low-power timer
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.9.3.1 LPTMR Instantiation Information

The low-power timer (LPTMR) allows operation during all power modes. The LPTMR can operate as a real-time interrupt or pulse accumulator. It includes a 2^N prescaler (real-time interrupt mode) or glitch filter (pulse accumulator mode).

The LPTMR can be clocked from the internal reference clock, the internal 1 kHz LPO, OSCERCLK, or an external 32.768 kHz crystal.

An interrupt is generated (and the counter may reset) when the counter equals the value in the 16-bit compare register.

3.9.3.2 LPTMR pulse counter input options

The LPTMR_CSR[TPS] bitfield configures the input source used in pulse counter mode. The following table shows the chip-specific input assignments for this bitfield.

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0 output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	Reserved

3.9.3.3 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from one of four sources determined by the LPTMR0_PSR[PCS] bitfield. The following table shows the chip-specific clock assignments for this bitfield.

NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	MCGIRCLK — internal reference clock
01	1	LPO — 1 kHz clock
10	2	ERCLK32K
11	3	OSCERCLK — external reference clock

3.9.4 RTC configuration

This section summarizes how the module has been configured in the chip.

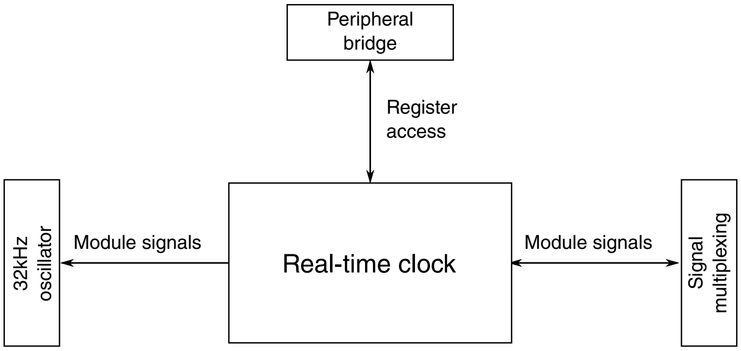


Figure 3-31. RTC configuration

Table 3-42. Reference links to related information

Topic	Related module	Reference
Full description	RTC	RTC
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

3.9.4.1 RTC Instantiation Information

RTC prescaler is clocked by ERCLK32K.

RTC is reset on POR Only.

RTC_CR[OSCE] is used to enable the 32kHz oscillator, and the RTC_CR register's SC2P, SC4P, SC8P and SC16P bit-fields are used to configure the 32kHz oscillator.

RTC_CR[WPE] and RTC_CR[WPS] are not used since the RTC wakeup pin output is not connected in this device.

Before using the 32kHz oscillator as the external reference source for the MCG, the RTC_CR[OSCE] bit should be set.

If an external square wave clock is being used to clock the RTC, the RTC_CLKIN path must be used.

3.9.4.2 RTC_CLKOUT options

RTC_CLKOUT pin is driven with the RTC 1Hz output.

3.10 Communication interfaces

3.10.1 SPI configuration

This section summarizes how the module has been configured in the chip.

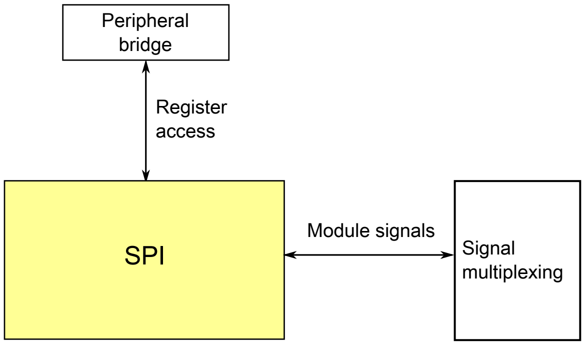


Figure 3-32. SPI configuration

Table 3-43. Reference links to related information

Topic	Related module	Reference
Full description	SPI	SPI
System memory map		System memory map
Clocking		Clock Distribution
Signal Multiplexing	Port control	Signal Multiplexing

3.10.1.1 SPI Instantiation Information

This device contains two DSPI modules which can be used as either SPI slave or SPI master, depending on the use case.

The configuration for the DSPI modules are shown in the table below.

Table 3-44. DSPI configuration

Parameter	SPI0	SPI1
CTAR Registers	2	2
TX FIFO Depth	4	4
RX FIFO Depth	4	4

The DSPI module must be in Run or Wait modes to operate as either a master or slave.

The reset value of TFFF bit in SPI_SR register will be 0 without any operation after reset. If MDIS bit in DSPI_MCR register is written with 1 after reset and the DSPI clock is enabled. TFFF bit will be 1.

3.10.2 I2C Configuration

This section summarizes how the module has been configured in the chip.

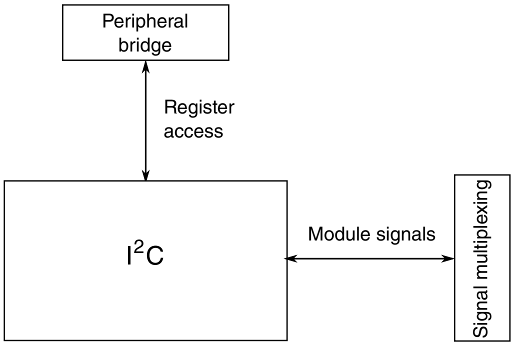


Figure 3-33. I2C configuration

Table 3-45. Reference links to related information

Topic	Related module	Reference
Full description	I²C	I²C
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.10.2.1 I2C Instantiation Information

This device has two I2C modules. I2C0 is clocked by the bus clock and I2C1 is clocked by the system clock.

When the package pins associated with I2C have their mux select configured for I2C operation, the pins (SCL and SDA) are driven in a pseudo open drain configuration.

The digital glitch filter implemented in the I2C0 module, controlled by the I2C0_FLT[FLT] registers, is clocked from the bus clock and thus has filter granularity in bus clock cycle counts.

The digital glitch filter implemented in the I2C1 module, controlled by the I2C1_FLT[FLT] registers, is clocked from the system clock and thus has filter granularity in system clock cycle counts.

The pull up voltage on a pseudo open drain pin should not be higher than VDD.

3.10.3 UART Configuration

This section summarizes how the module has been configured in the chip.

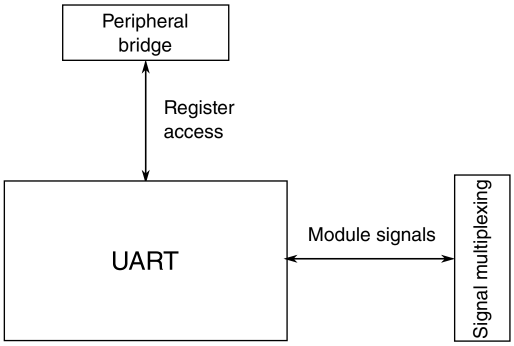


Figure 3-34. UART configuration

Table 3-46. Reference links to related information

Topic	Related module	Reference
Full description	UART0	UART
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

3.10.3.1 UART0 overview

The UART0 module supports basic UART with DMA interface function, x4 to x32 oversampling of baud-rate, and hardware flow control.

The UART0 module RX and TX FIFOs are 8 entries each.

The module can remain functional in VLPS mode provided the clock it is using remains enabled.

3.11 Human-machine interfaces (HMI)

3.11.1 GPIO Configuration

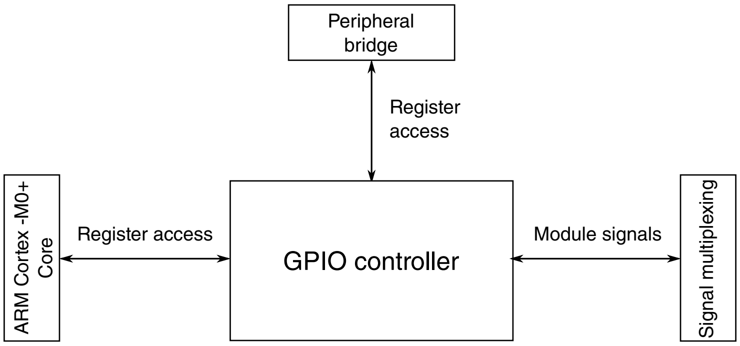


Figure 3-35. GPIO configuration

Table 3-47. Reference links to related information

Topic	Related module	Reference
Full description	GPIO	GPIO
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Crossbar switch	Crossbar switch	Crossbar switch
Signal Multiplexing	Port control	Signal Multiplexing

3.11.1.1 GPIO Instantiation Information

The device will include 6 pins with high current drive capability. These pins are PTC0, PTC1, PTC2, PTC3, PTB0 and PTB1. High drive can be controlled for these pin using the PORTx_PCRn[DSE] field.

3.11.1.1.1 Pull Devices and Directions

The pull devices are enabled out of POR only on RESET_B, NMI_b and respective SWD signals. Other PORT pins can be enabled by writing to PORTx_PCRn[PE] field.

All the PORT pins have controllable pull direction using the PORTx_PCRn[PS] field. All the pins default to pullup except for SWD_CLK, when enabled.

3.11.1.2 Port control and interrupt summary

The following table provides more information regarding the Port Control and Interrupt configurations .

Table 3-48. Ports summary

Feature	Port A	Port B	Port C
Pull select control	Yes	Yes	Yes
Pull select at reset	PTA1=Pull down, Others=Pull up	Pull up	Pull up
Pull enable control	Yes	Yes	Yes
Pull enable at reset	PTA0/PTA1/PTA2=Enabled; Others=Disabled	Disabled	Disabled
Slew rate enable control	Yes	Yes	Yes
Slew rate enable at reset	PTA16/PTA17/PTA18/ PTA19=Disabled; Others=Enabled	PTB3 = Disabled; Others=Enabled	PTC7/PTC16/PTC17/PTC18/ PTC17=Disabled; Others=Enabled
Passive filter enable control	No	PTB18 (NMI_b) only	No
Passive filter enable at reset	Disabled	PTB18=Enabled; Others=Disabled	Disabled
Open drain enable control[1]	No	No	No
Open drain enable at reset	Disabled	Disabled	Disabled
Drive strength enable control	No	PTB0/PTB1 only	PTC0/PTC1/PTC2/PTC3 only
Drive strength enable at reset	Disabled	Disabled	Disabled
Pin mux control	Yes	Yes	Yes
Pin mux at reset	PTA0/PTA1/PTA2=ALT7; Others=ALT0	PTB18=ALT7; Others=ALT0	ALT0
Lock bit	No	No	No
Interrupt and DMA request	Yes	Yes	Yes
Digital glitch filter	No	No	No

[1] UART signals can be configured for open-drain using SIM_SOPT5 register. I2C signals are automatically enabled for open drain when selected.

3.11.1.3 GPIO accessibility in the memory map

The GPIO is multi-ported and can be accessed directly by the core with zero wait states at base address 0xF800_0000. It can also be accessed by the core and DMA masters through the cross bar/AIPS interface at 0x400F_F000 and at an aliased slot (15) at address 0x4000_F000. All BME operations to the GPIO space can be accomplished referencing the aliased slot (15) at address 0x4000_F000. Only some of the BME operations can be accomplished referencing GPIO at address 0x400F_F000.

3.11.2 TSI Configuration

This section summarizes how the module has been configured in the chip.

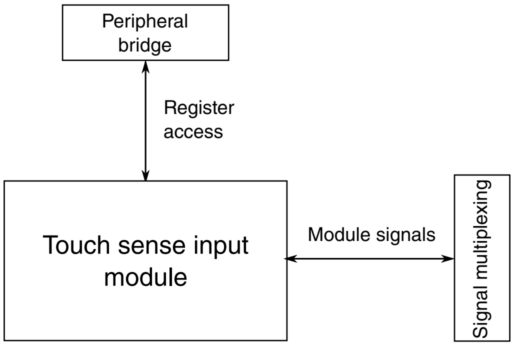


Figure 3-36. TSI configuration

Table 3-49. Reference links to related information

Topic	Related module	Reference
Full description	TSI	TSI
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.11.2.1 TSI Instantiation Information

This device includes one 16channel TSI module. All 16 channels are supported in the 48LGA package. In Stop, VLPS, LLSx and VLLSx modes any one channel can be enabled to be the wakeup source.

TSI hardware trigger is from the LPTMR. For complete details on the LPTMR module interconnects refer to the Module-to-Module section.

3.11.2.2 TSI Interrupts

The TSI has multiple sources of interrupt requests. However, these sources are OR'd together to generate a single interrupt request. When a TSI interrupt occurs, read the TSI status register to determine the exact interrupt source.

Chapter 4
Memory Map

This section describes the memory and peripheral locations within the memory space of this device.

4.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in a 4G bytes memory space. This chapter describes the memory and peripheral locations within that memory space.

4.2 System memory map

The following table shows the high-level device memory map.

Table 4-1. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF[1]	Program flash and read-only data (Includes exception vectors in first 196 bytes)	All masters
0x0800_0000–0x1FFF_EFFF	Reserved	—
0x1FFF_F000-0x1FFF_FFFF	SRAM_L: Lower SRAM	All masters
0x2000_0000-0x2000_3FFF	SRAM_U: Upper SRAM[2]	All masters
0x2000_4000–0x3FFF_FFFF	Reserved	–
0x4000_0000–0x4007_FFFF	AIPS Peripherals	Cortex-M0+ core & DMA
0x4008_0000–0x400F_EFFF	Reserved	–
0x400F_F000–0x400F_FFFF	General purpose input/output (GPIO)	Cortex-M0+ core & DMA
0x4010_0000–0x43FF_FFFF	Reserved	–
0x4400_0000–0x5FFF_FFFF	Bit Manipulation Engine (BME) access to AIPS Peripherals for slots 0-127[3]	Cortex-M0+ core

Table continues on the next page...

Table 4-1. System memory map (continued)

System 32-bit Address Range	Destination Slave	Access
0x6000_0000–0xDFFF_FFFF	Reserved	–
0xE000_0000–0xE00F_FFFF	Private Peripherals	Cortex-M0+ core
0xE010_0000–0xEFFF_FFFF	Reserved	–
0xF000_0000–0xF000_0FFF	Micro Trace Buffer (MTB) registers	Cortex-M0+ core
0xF000_1000–0xF000_1FFF	MTB Data Watchpoint and Trace (MTBDWT) registers	Cortex-M0+ core
0xF000_2000–0xF000_2FFF	ROM table	Cortex-M0+ core
0xF000_3000–0xF000_3FFF	Miscellaneous Control Module (MCM)	Cortex-M0+ core
0xF000_4000–0xF7FF_FFFF	Reserved	–
0xF800_0000–0xFFFF_FFFF	IOPORT: GPIO (single cycle)	Cortex-M0+ core

1. The program flash always begins at 0x0000_0000 but the end of implemented flash varies depending on the amount of flash implemented for a particular device.
2. Refer to SRAM Ranges in the Peripherals chapter for more information on the split of the SRAM into Lower and Upper regions
3. Includes BME operations to GPIO at slot 15 (based at 0x4000_F000)

4.3 Flash Memory Map

The flash memory and the flash registers are located at different base addresses as shown in the following figure. The base address for each is specified in System memory map.

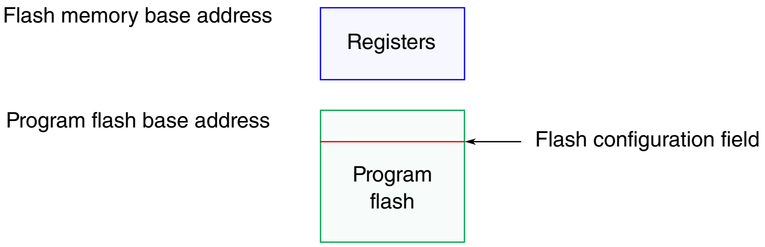

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program
flash

Figure 4-1. Flash memory map

The on-chip Flash is implemented in a portion of the allocated Flash range to form a contiguous block in the memory map beginning at address 0x0000_0000. See Flash Memory Sizes for details of supported ranges.

Accesses to the Flash memory ranges outside the amount of Flash on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master.

4.3.1 Alternate Non-Volatile IRC User Trim Description

The following non-volatile locations (4 bytes) are reserved for custom IRC user trim supported by some development tools. An alternate IRC trim to the factory loaded trim can be stored at this location. To override the factory trim, user software must load new values into the MCG trim registers.

Non-Volatile Byte Address	Alternate IRC Trim Value
0x0000_03FC	Reserved
0x0000_03FD	Reserved
0x0000_03FE (bit 0)	SCFTRIM
0x0000_03FE (bit 4:1)	FCTRIM
0x0000_03FE (bit 6)	FCFTRIM
0x0000_03FF	SCTRIM

4.4 SRAM memory map

The on-chip RAM is split between SRAM_L and SRAM_U. The RAM is also implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. See SRAM Ranges for details.

Accesses to the SRAM_L and SRAM_U memory ranges outside the amount of RAM on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master.

4.5 Bit Manipulation Engine

The Bit Manipulation Engine (BME) provides hardware support for atomic read-modify-write memory operations to the peripheral address space. By combining the basic load and store instruction support in the Cortex-M instruction set architecture with the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers. See the Bit Manipulation Engine Block Guide (BME) for a detailed description of BME functionality.

4.6 Peripheral bridge (AIPS-Lite) memory map

The peripheral memory map is accessible via one slave port on the crossbar in the 0x4000_0000–0x400F_FFFF region. The device implements one peripheral bridge that defines a 1024 KB address space.

The three regions associated with this space are:

- A 128 KB region, partitioned as 32 spaces, each 4 KB in size and reserved for on-platform peripheral devices. The AIPS controller generates unique module enables for all 32 spaces.
- A 384 KB region, partitioned as 96 spaces, each 4 KB in size and reserved for off-platform modules. The AIPS controller generates unique module enables for all 96 spaces.
- The last slot is a 4 KB region beginning at 0x400F_F000 for accessing the GPIO module. The GPIO slot (slot 128) is an alias of slot 15. This block is also directly interfaced to the core and provides direct access without incurring wait states associated with accesses via the AIPS controller.

Modules that are disabled via their clock gate control bits in the SIM registers disable the associated AIPS slots. Access to any address within an unimplemented or disabled peripheral bridge slot results in a transfer error termination.

For programming model accesses via the peripheral bridges, there is generally only a small range within the 4 KB slots that is implemented. Accessing an address that is not implemented in the peripheral results in a transfer error termination.

4.6.1 Read-after-write sequence and required serialization of memory operations

In some situations, a write to a peripheral must be completed fully before a subsequent action can occur. Examples of such situations include:

- Exiting an interrupt service routine (ISR)
- Changing a mode
- Configuring a function

In these situations, the application software must perform a read-after-write sequence to guarantee the required serialization of the memory operations:

1. Write the peripheral register.
2. Read the written peripheral register to verify the write.
3. Continue with subsequent operations.

4.6.2 Peripheral Bridge (AIPS-Lite) Memory Map

Table 4-2. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module
0x4000_0000	0	—
0x4000_1000	1	—
0x4000_2000	2	—
0x4000_3000	3	—
0x4000_4000	4	—
0x4000_5000	5	—
0x4000_6000	6	—
0x4000_7000	7	—
0x4000_8000	8	DMA controller
0x4000_9000	9	—
0x4000_A000	10	—
0x4000_B000	11	—
0x4000_C000	12	—
0x4000_D000	13	—
0x4000_E000	14	—
0x4000_F000	15	GPIO controller (aliased to 0x400F_F000)
0x4001_0000	16	—
0x4001_1000	17	—
0x4001_2000	18	—
0x4001_3000	19	—
0x4001_4000	20	—
0x4001_5000	21	—
0x4001_6000	22	—
0x4001_7000	23	—
0x4001_8000	24	—
0x4001_9000	25	—
0x4001_A000	26	—
0x4001_B000	27	—
0x4001_C000	28	—
0x4001_D000	29	—
0x4001_E000	30	—
0x4001_F000	31	—
0x4002_0000	32	Flash memory
0x4002_1000	33	DMA channel mutiplexer 0
0x4002_2000	34	—
0x4002_3000	35	—
0x4002_4000	36	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	True Random Number Generator
0x4002_A000	42	—
0x4002_B000	43	—
0x4002_C000	44	SPI0
0x4002_D000	45	SPI1
0x4002_E000	46	—
0x4002_F000	47	—
0x4003_0000	48	—
0x4003_1000	49	—
0x4003_2000	50	—
0x4003_3000	51	—
0x4003_4000	52	—
0x4003_5000	53	—
0x4003_6000	54	—
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	Timer/PWM (TPM) 0
0x4003_9000	57	Timer/PWM (TPM) 1
0x4003_A000	58	Timer/PWM (TPM) 2
0x4003_B000	59	Analog-to-digital converter (ADC) 0
0x4003_C000	60	—
0x4003_D000	61	Real-time clock (RTC)
0x4003_E000	62	—
0x4003_F000	63	DAC0
0x4004_0000	64	Low-power timer (LPTMR)
0x4004_1000	65	System register file
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	Touch Sense Input (TSI)
0x4004_6000	70	—
0x4004_7000	71	SIM low-power logic
0x4004_8000	72	System integration module (SIM)
0x4004_9000	73	Port A multiplexing control
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4004_C000	76	
0x4004_D000	77	
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	—
0x4005_3000	83	—
0x4005_4000	84	UART 0
0x4005_5000	85	—
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	LP Trusted Cryptography(LTC)
0x4005_9000	89	Radio System Integration Module (RSIM)
0x4005_A000	90	DCDC
0x4005_B000	91	Bluetooth Low Energy (BLE) Link Layer (BTLL)
0x4005_C000	92	PHY_DIG
0x4005_D000	93	ZigBee/802.15.4
0x4005_E000	94	—
0x4005_F000	95	—
0x4006_0000	96	—
0x4006_1000	97	—
0x4006_2000	98	Carrier modulator timer (CMT)
0x4006_3000	99	—
0x4006_4000	100	Multi-purpose Clock Generator (MCG)
0x4006_5000	101	—
0x4006_6000	102	I^{2}C 0
0x4006_7000	103	I^{2}C 1
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	—
0x4006_B000	107	—
0x4006_C000	108	—
0x4006_D000	109	—
0x4006_E000	110	—
0x4006_F000	111	—
0x4007_0000	112	—
0x4007_1000	113	—
0x4007_2000	114	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4007_3000	115	Analog comparator (CMP) / 6-bit digital-to-analog converter (DAC)
0x4007_4000	116	—
0x4007_5000	117	—
0x4007_6000	118	—
0x4007_7000	119	—
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	Low-leakage wakeup unit (LLWU)
0x4007_D000	125	Power management controller (PMC)
0x4007_E000	126	System Mode controller (SMC)
0x4007_F000	127	Reset Control Module (RCM)
0x400F_F000	128	GPIO controller

4.6.3 Modules Restricted Access in User Mode

In user mode, for RCM, SIM (slot 71 and 72), SMC, LLWU, and PMC, reads are allowed, but writes are blocked and generate bus errors.

In user mode, for MCG, writes are blocked.

By default, the SRTC blocks write access in user mode, but this restriction can be removed by programming the RTC_CR register's SUP bitfield.

4.7 Private Peripheral Bus (PPB) memory map

The PPB is part of the defined ARM bus architecture and provides access to select processor-local modules. These resources are only accessible from the core; other system masters do not have access to them.

Table 4-3. PPB memory map

System 32-bit Address Range	Resource	Additional Range Detail	Resource
0xE000_0000–0xE000_DFFF	Reserved		
0xE000_E000–0xE000_EFFF	System Control Space (SCS)	0xE000_E000–0xE000_E00F	Reserved

Table continues on the next page...

Table 4-3. PPB memory map (continued)

System 32-bit Address Range	Resource	Additional Range Detail	Resource
		0xE000_E010–0xE000_E0FF	SysTick
		0xE000_E100–0xE000_ECFF	NVIC
		0xE000_ED00–0xE000_ED8F	System Control Block
		0xE000_ED90–0xE000_EDEF	Reserved
		0xE000_EDF0–0xE000_EEFF	Debug
		0xE000_EF00–0xE000_EFFF	Reserved
0xE000_F000–0xE00F_EFFF	Reserved		
0xE00F_F000–0xE00F_FFFF	Core ROM Space (CRS)		

Chapter 5
Clock Distribution

5.1 Introduction

This chapter presents the clock architecture for the device, the overview of the clocks and includes a terminology section.

The Cortex M0+ resides within a synchronous core platform, where the processor and bus masters, Flash and peripheral clocks can be configured independently. The clock distribution figure shows how clocks from the MCG, Reference Oscillator and 32kHz Oscillator modules are distributed to the microcontroller’s other function units. Some modules in the microcontroller have selectable clock input.

5.2 Programming model

The selection and multiplexing of system clock sources is controlled and programmed via the MCG module. The setting of clock dividers and module clock gating for the system are programmed via the SIM module. Reference those sections for detailed register and bit descriptions.

5.3 High-Level device clocking diagram

The device includes the following clock sources:

- RF Reference oscillator. This supports a 32MHz crystal. The clock is used by the radio analog and digital. It can be used also by the MCU core and as a clock source for some peripherals
- 32kHz RTC oscillator. This is used as the clock for the RTC and the deepsleep clock for the Bluetooth LE . The 32kHz IRC clock is not accurate enough to use for the deepsleep clock, and consumes significantly more power as well.

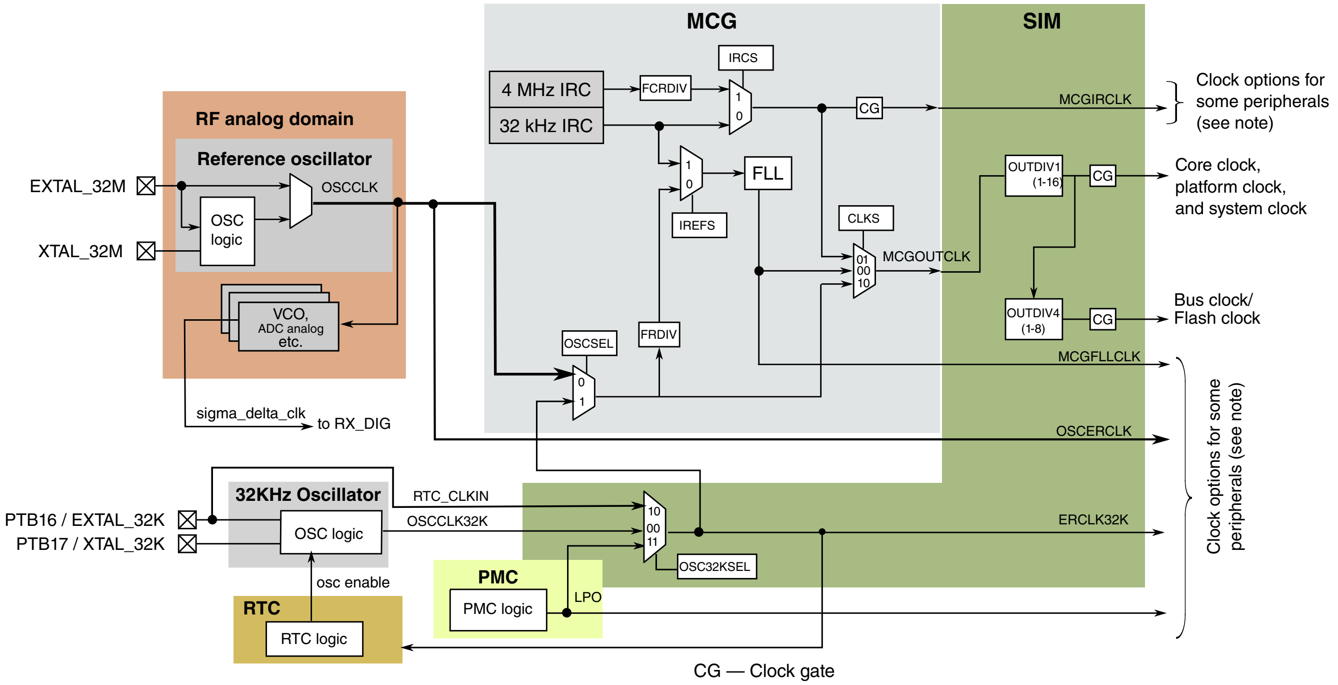
- 32kHz IRC. This is used as the reference for the FLL at reset, and is a clock option for some peripherals
- 4MHz IRC. This is a clock option for the MCU primarily intended to support VLPx mode, and is also a clock option for some peripherals.

See Reference Oscillator Instantiation Information for more information on control of the reference oscillator.

The 32kHz oscillator will be controlled through the RTC

The following MCG, and SIM module registers control the multiplexers, dividers, and clock gates shown in the below figure:

	MCG	SIM
Multiplexers	MCG_C*x*	SIM_SOPT1, SIM_SOPT2
Dividers	MCG_C*x*	SIM_CLKDIV*x*
Clock gates	MCG_C1	SIM_SCGC*x*



CG — Clock gate

Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram

5.4 Clock definitions

The following table describes the clocks in the previous block diagram.

Clock name	Description
Core clock	MCGOUTCLK divided by OUTDIV1, clocks the ARM Cortex-M0+ core
Platform clock	MCGOUTCLK divided by OUTDIV1, clocks the crossbar switch and NVIC
System clock	MCGOUTCLK divided by OUTDIV1, clocks the bus masters directly
Bus clock	System clock divided by OUTDIV4, clocks the bus slaves and peripherals.
Flash clock	Flash memory clock. On this device it is the same as Bus clock.
MCGIRCLK	MCG output of the internal reference clock
MCGOUTCLK	MCG output of either IRC, MCGFLLCLK or MCG's external reference clock that sources the core, system, bus, and flash clock.
MCGFLLCLK	MCG output of the FLL. MCGFLLCLK may clock some modules.
OSCCLK	Output of the internal Reference oscillator or sourced directly from EXTAL. Used as MCG external reference clock.
OSCERCLK	Reference oscillator output sourced from OSCCLK that clocks some on-chip modules
OSC32KCLK	Output of the internal 32KHz oscillator. A 32.768kHz crystal or clock is normally used to support RTC features. If RTC is not needed, a 32kHz crystal or clock is an option which is supported by BTLL
ERCLK32K	Clock source for some modules. Can be selected as either OSC32KCLK (default), RTC_CLKIN or LPO. This clock is used to provide the 32kHz clock to the LE link layer; when using BLE, LPO should not be selected.
LPO	PMC 1kHz output

5.4.1 Device clock summary

The following table provides more information regarding the on-chip clocks.

Table 5-1. Clock Summary

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
MCGOUTCLK	Up to 48 MHz	Up to 4 MHz	MCG	In all stop modes except for partial stop modes
MCGFLLCLK	Up to 48 MHz	N/A	MCG	MCG clock controls do not enable, and in all stop modes
Core clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all wait and stop modes

Table continues on the next page...

Table 5-1. Clock Summary (continued)

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
Platform clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes
System clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes Compute Operation
Bus clock	Up to 24 MHz	Up to 1 MHz in BLPE[1] Up to 800 kHz in BLPI[2]	MCGOUTCLK clock divider	In all stop modes except for partial STOP2 mode, and Compute Operation
SWD Clock	Up to 24 MHz	Up to 1 MHz	SWD_CLK pin	In all stop modes
Flash clock	Up to 24 MHz	Up to 1 MHz in BLPE Up to 800 kHz in BLPI	MCGOUTCLK clock divider	In all stop modes except for partial STOP2 mode
Internal reference (MCGIRCLK)	30-40 kHz	4 MHz Fast IRC only	MCG	MCG_C1[IRCLKEN] cleared, Stop/VLPS mode and MCG_C1[IREFSTEN] cleared, or LLS/VLLS mode
Reference oscillator (OSCERCLK)	32 MHz	DCDC configured for continuous mode: 32 MHz Otherwise: N/A	Reference Oscillator	See Reference Oscillator Enable Sources
32K External reference (OSC32KCLK)	32 KHz	32 kHz	32KHz Oscillator	RTC's RTC_CR[OSCE] cleared
External reference 32kHz (ERCLK32K)	30-40 kHz	30-40 kHz	32kHz Oscillator, RTC_CLKIN or LPO	Selected clock source disabled
LPO	1 kHz	1 kHz	PMC	Available in all power modes except VLLS0
UART clock	Up to 48 MHz	Up to 4 MHz	MCGIRCLK, MCGFLLCLK, or OSCERCLK	SIM_SOPT2[UART0SRC]=00 or selected clock source disabled.

1. BLPE: MCG mode where MCGOUT is derived from an external oscillator. For KW40Z, use of BLPE in VLPR mode is only feasible when the DCDC is configured for continuous mode
2. BLPI: MCG mode where MCGOUT is derived from the internal reference.

5.5 Internal clocking requirements

The clock dividers are programmed via the SIM module's CLKDIV registers. The following requirements must be met when configuring the clocks for this device:

1. The core, platform, and system clock are programmable from a divide-by-1 through divide-by-16 setting. The core, platform, and system clock frequencies must be 48 MHz or slower.
2. The bus clock and flash clock frequency is divided from the system clock and is programmable from a divide-by-1 through divide-by-8 setting. The bus clock and flash clock must be programmed to 24 MHz or slower.

5.5.1 Clock divider values after reset

Out of reset, the MCG selects the FLL, using the 32KHz internal reference, as the MCGOUTCLK output. This clock is approximately 20MHz.

Two bits in the flash memory's FTFA_FOPT register controls the reset value of the core clock, system clock, bus clock, and flash clock dividers (in the SIM's CLKDIV1 register) as shown below:

FTFA_FOPT [4,0]	Core/system clock	Bus/Flash clock	Description
00	0x7 (divide by 8), ~2.5MHz	0x1 (divide by 2), ~1.25MHz	Low power boot
01	0x3 (divide by 4), ~5MHz	0x1 (divide by 2), ~2.5MHz	Low power boot
10	0x1 (divide by 2), ~10MHz	0x1 (divide by 2), ~5MHz	Low power boot
11	0x0 (divide by 1), ~20MHz	0x1 (divide by 2), ~10MHz	Fast clock boot

This gives the user flexibility in selecting between a lower frequency, low-power boot option vs. higher frequency, higher power during and after reset.

The flash erased state defaults to fast clocking mode, since these bits reside in flash, which is logic 1 in the flash erased state. To enable a lower power boot option, program the appropriate bits in FTFA_FOPT. During the reset sequence, if either of the control bits is cleared, the system is in a slower clock configuration. Upon any system reset, the clock dividers return to this configurable reset state.

5.5.2 VLPR mode clocking

For KW4x device, the VLPx modes are provided primarily as an option for the MCU subsystem to consume less power when the radio is not being used. It is possible to use VLPx modes when the radio is active, but only if the DCDC is configured for continuous mode, in which case biasing will not be enabled. The radio requires use of the 32MHz oscillator and its clock cannot be used in VLPx modes unless the DCDC is configured for continuous mode. The MCG BLPI mode therefore needs to be used in most VLPx use cases.

Some additional restrictions on VLPR mode are provided below.

The clock dividers cannot be changed while in VLPR mode. They must be programmed prior to entering VLPR mode to guarantee operation. Max frequency limitations for VLPR mode are as follows :

- the core/system clocks are less than or equal to 4 MHz, and
- the bus and flash clocks are
  - less than or equal to 800 kHz if using BLPI
  - less than or equal to 1 MHz if using BLPE. As this requires the use of the 32MHz oscillator, this is only possible when the DCDC is configured for continuous mode

5.6 Clock Gating

The clock to each module can be individually gated on and off using the SIM module's SCGC*x* registers. Most of these bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing a module, set the corresponding bit in SCGC*x* register to enable the clock. Before turning off the clock, make sure to disable the module.

Any bus access to a peripheral that has its clock disabled generates an error termination.

Refer to Introduction for more information on the SIM.

5.7 Module clocks

The following table summarizes the clocks associated with each module.

Table 5-2. Module clocks

Module	Bus interface clock	Internal clocks	I/O interface clocks
Core modules			
ARM Cortex-M0+ core	Platform clock	Core clock	—
NVIC	Platform clock	—	—
DAP	Platform clock	—	SWD_CLK
System modules			
DMA	System clock	—	—
DMA Mux	Bus clock	—	—
Port control	Bus clock	—	—
Crossbar Switch	Platform clock	—	—

Table continues on the next page...

Table 5-2. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
Peripheral bridges	System clock	Bus clock	—
LLWU, PMC, SIM, RCM	Bus clock	LPO	—
Mode controller	Bus clock	—	—
MCM	Platform clock	—	—
COP	Bus clock	COP clock	—
Clocks			
MCG	Bus clock	MCGOUTCLK, MCGFLLCLK, MCGIRCLK, OSCERCLK	—
Reference Oscillator	—	OSCERCLK	—
32KHz Oscillator	—[1]	OSC32KCLK	—
Memory and memory interfaces			
Flash Controller	Platform clock		—
Flash memory	Flash clock	—	—
Analog			
ADC	Bus clock	OSCERCLK	—
CMP	Bus clock	—	—
DAC	Bus clock	—	—
Timers			
TPM0/1/2	Bus clock	TPM clock	TPM_CLKINx
PIT	Bus clock	—	—
CMT	Bus clock	—	—
LPTMR	Bus clock	LPTMR clock	—
RTC	Bus clock	ERCLK32K	—
Communication interfaces			
SPI0/1	Bus clock	—	SPI0_SCK, SPI1_SCK
I^2C0	Bus clock	—	I2C0_SCL
I^2C1	System clock	—	I2C1_SCL
UART	Bus clock	UART clock	—
Human-machine interfaces			
GPIO	Platform clock	—	—
TSI	Bus clock	—	—
Security			
AESA	System clock	—	—
TRNG	Bus clock	—	—
Radio			
RF/Analog	—	OSC CLK	—

[1] In KW4x, the 32 kHz osc will be controlled via the RTC

5.7.1 PMC 1-kHz LPO clock

The Power Management Controller (PMC) generates a 1-kHz clock that is enabled in all modes of operation, including all low power modes except VLLS0. This 1-kHz source is commonly referred to as LPO clock or 1-kHz LPO clock.

5.7.2 COP clocking

The COP may be clocked from four clock sources as shown in the following figure.

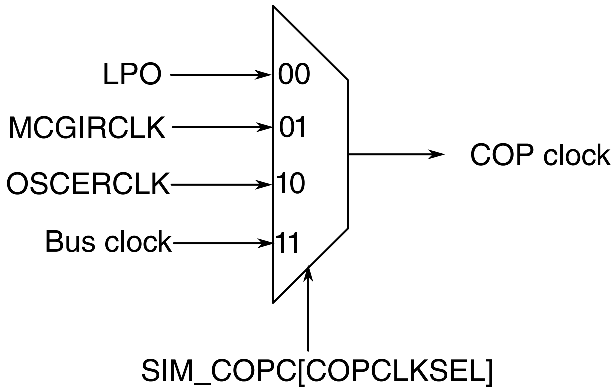


Figure 5-2. COP clock generation

5.7.3 RTC clocking

The RTC module can be clocked as shown in the following figure.

NOTE

The chosen clock must remain enabled if the RTC is to continue operating in all required low-power modes.

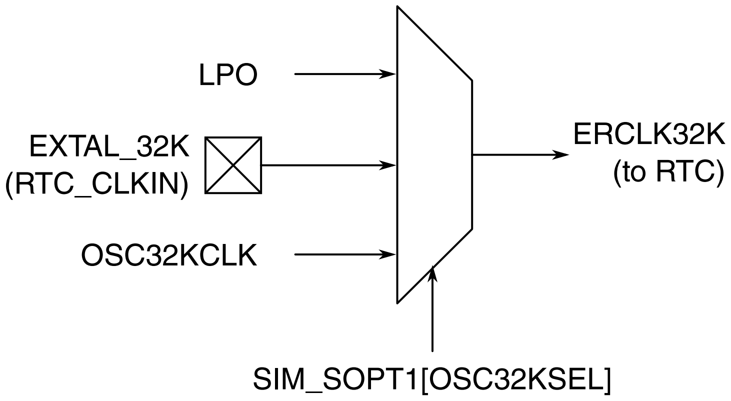


Figure 5-3. RTC clock generation

5.7.4 LPTMR clocking

The prescaler and glitch filters in each of the LPTMR*x* modules can be clocked as shown in the following figure.

NOTE

The chosen clock must remain enabled if the LPTMR*x* is to continue operating in all required low-power modes.

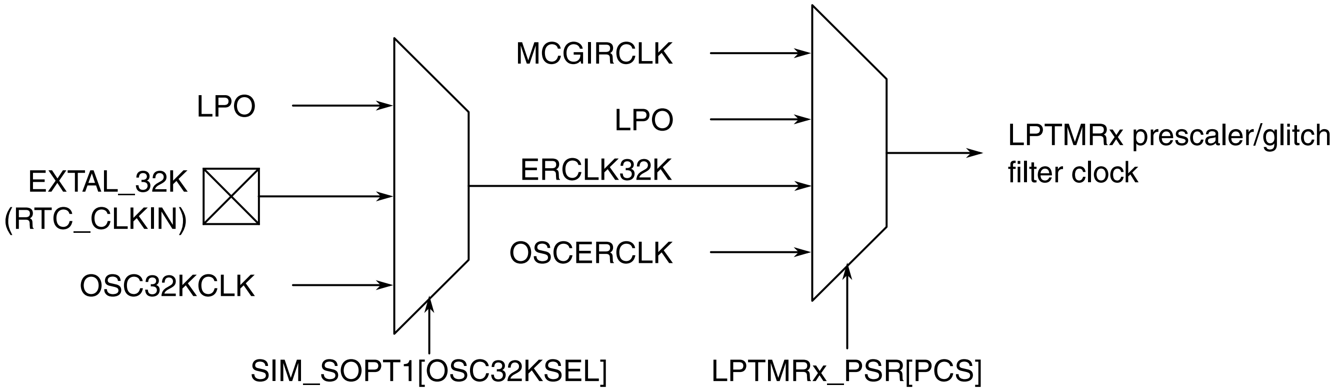


Figure 5-4. LPTMRx prescaler/glitch filter clock generation

5.7.5 TPM clocking

The clock used by the TPM modules can be selected as shown in the following figure.

NOTE

The chosen clock must remain enabled if the TPM is to continue operating in all required low-power modes.

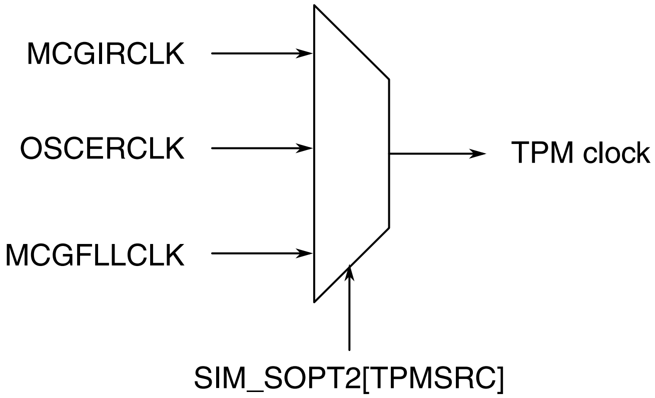


Figure 5-5. TPM clock generation

5.7.6 UART clocking

The UART module has a selectable clock as shown in the following figure.

NOTE

The chosen clock must remain enabled if the UART is to continue operating in all required low-power modes.

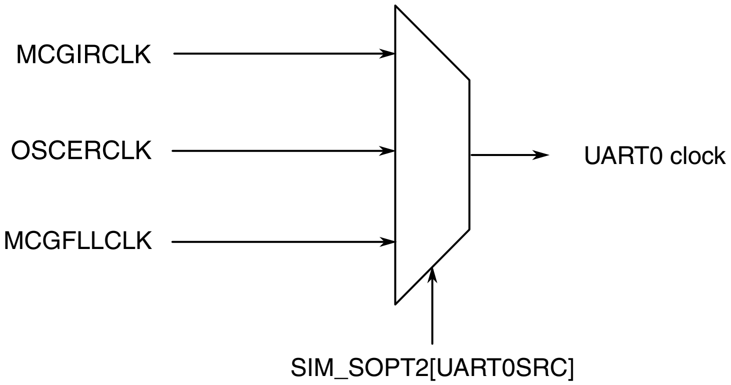


Figure 5-6. UART clock generation

Chapter 6
Reset and Boot

This section describes the basic reset and boot mechanisms and sources for this device.

6.1 Introduction

The reset sources supported in this MCU are listed in the table found here.

Table 6-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low-voltage detect (LVD) • Computer operating properly (COP) watchdog reset • Low leakage wakeup (LLWU) reset • Multipurpose clock generator loss of clock (LOC) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • MDM DAP system reset
Debug reset	• Debug reset

Each of the system reset sources has an associated bit in the System Reset Status (SRS) registers. See the Reset Control Module for register details.

The MCU can exit and reset in functional mode where the CPU is executing code (default) or the CPU is in a debug halted state. There are several boot options that can be configured. See Boot information for more details.

6.2 Reset

The information found here discusses basic reset mechanisms and sources.

Some modules that cause resets can be configured to cause interrupts instead. Consult the individual peripheral chapters for more information.

6.2.1 Power-on reset (POR)

When power is initially applied to the PMC or when the supply voltage drops below the power-on reset re-arm voltage level (V_{POR}), the PMC's POR circuit causes a POR reset condition.

As the supply voltage rises, the LVD circuit holds the MCU in reset until the supply has risen above the LVD low threshold (V_{LVDL}). The POR and LVD fields in the Reset Status Register are set following a POR.

6.2.2 System reset sources

Resetting the MCU provides a way to start processing from a known set of initial conditions. System reset begins with the on-chip PMC/DCDC regulators in full regulation and system clocking generation from an internal reference. When the processor exits reset, it performs the following:

- Reads the start SP (SP_main) from vector-table offset 0
- Reads the start PC from vector-table offset 4
- LR is set to 0xFFFF_FFFF

The on-chip peripheral modules are disabled and the non-analog I/O pins are initially configured as disabled. The pins with analog functions assigned to them default to their analog function after reset.

During and following a reset, the SWD pins have their associated input pins configured as:

- SWD_CLK in pull-down (PD)
- SWD_DIO in pull-up (PU)

6.2.2.1 External pin reset (RESET_b)

This pin is open drain and has an internal pullup device. Asserting RESET_b wakes the device from any mode.

The RESET_b pin can be disabled by programming RESET_PIN_CFG option bit to 0. When this option is selected, there could be a short period of contention during a POR ramp where the device drives the pin-out low prior to establishing the setting of this option and releasing the reset function on the pin. When the RESET pin is disabled and configured as a GPIO output, it operates as a pseudo open drain output.

6.2.2.1.1 $\overline{\text{RESET}}$ pin filter

The $\overline{\text{RESET}}$ pin filter supports filtering from both the 1 kHz LPO clock and the bus clock. RCM_RPFC[RSTFLTSS], RCM_RPFC[RSTFLTSRW], and RCM_RPFW[RSTFLTSEL] control this functionality; see the RCM chapter. The filters are asynchronously reset by Chip POR. The reset value for each filter assumes the $\overline{\text{RESET}}$ pin is negated.

For all stop modes where LPO clock is still active (Stop, VLPS, LLS, VLLS3, and VLLS1), the only filtering option is the LPO-based digital filter. The filtering logic either switches to bypass operation or has continued filtering operation depending on the filtering mode selected. When entering VLLS0, the $\overline{\text{RESET}}$ pin filter is disabled and bypassed.

The LPO filter has a fixed filter value of 3. Due to a synchronizer on the input data, there is also some associated latency (2 cycles). As a result, 5 cycles are required to complete a transition from low to high or high to low.

6.2.2.2 Low-voltage detect (LVD)

The chip includes a system for managing low-voltage conditions to protect memory contents and control MCU system states during supply voltage variations. The system consists of a power-on reset (POR) circuit and an LVD circuit with a user-selectable trip voltage. The LVD system is always enabled in Normal Run, Wait, or Stop mode. The LVD system is disabled when entering VLPx, LLS, or VLLSx modes.

The LVD can be configured to generate a reset upon detection of a low-voltage condition by setting PMC_LVDSC1[LVDRE] to 1. The low-voltage detection threshold is determined by PMC_LVDSC1[LVDV]. After an LVD reset has occurred, the LVD system holds the MCU in reset until the supply voltage has risen above the low voltage detection threshold. RCM_SRS0[LVD] is set following either an LVD reset or POR.

When using the DCDC, the default configuration of the LVD and LVW levels will ensure proper device operation. If the DCDC is programmed to output a voltage higher than 1.8V on VDD_1p8OUT pin, the low-voltage warning (LVW) level can be changed if desired, but the LVD level should be left at its default value. It is also recommended that the PMC_LVDSC1[LVDRE] bit should remain set to allow LVD to generate a reset on a low-voltage condition.

6.2.2.3 Computer operating properly (COP) watchdog timer

The computer operating properly (COP) watchdog timer (WDOG) monitors the operation of the system by expecting periodic communication from the software. This communication is generally known as servicing (or refreshing) the COP watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset. The COP reset causes RCM_SRS0[WDOG] to set.

6.2.2.4 Low leakage wakeup (LLWU)

The LLWU module provides the means for a number of external pins and a number of internal peripherals to wake the MCU from low leakage power modes. The LLWU module is functional only in low leakage power modes. In VLLSx modes, all enabled inputs to the LLWU can generate a system reset.

After a system reset, the LLWU retains the flags indicating the input source of the last wakeup until the user clears them.

NOTE

Some flags are cleared in the LLWU and some flags are required to be cleared in the peripheral module. Refer to the individual peripheral chapters for more information.

6.2.2.5 Multipurpose clock generator loss-of-clock (LOC)

The MCG module supports external reference clocks.

If MCG_C6[CME] is set, the clock monitor associated with the RF reference oscillator is enabled. If the external reference falls below f_{loc_low} or f_{loc_high}, as controlled by MCG_C2[RANGE], the MCU resets. MCG_SC[LOCS0] and [RCM_SRS0[LOC] are set to indicate this reset source.

If MCG_C8[CME1] is set, the clock monitor associated with the RTC oscillator is enabled. If the external reference falls below f_{loc_low}, the MCU resets. MCG_C8[LOCS1] and RCM_SRS0[LOC] are set to indicate this reset source.

NOTE

> To prevent unexpected loss of clock reset events, all clock monitors must be disabled before entering any low-power modes, including VLPR and VLPW.

6.2.2.6 Stop mode acknowledge error (SACKERR)

This reset is generated if the core attempts to enter Stop mode or Compute Operation, but not all modules acknowledge Stop mode within 1025 cycles of the 1 kHz LPO clock.

A module might not acknowledge the entry to Stop mode if an error condition occurs. The error can be caused by a failure of an external clock input to a module.

6.2.2.7 Software reset (SW)

The SYSRESETREQ field in the NVIC Application Interrupt and Reset Control register can be set to force a software reset on the device. (See ARM's NVIC documentation for the full description of the register fields, especially the VECTKEY field requirements.) Setting SYSRESETREQ generates a software reset request. This reset forces a system reset of all major components except for the debug module. A software reset causes RCM_SRS1[SW] to set.

6.2.2.8 Lockup reset (LOCKUP)

The LOCKUP gives immediate indication of seriously errant kernel software. This is the result of the core being locked because of an unrecoverable exception following the activation of the processor’s built in system state protection hardware.

The LOCKUP condition causes a system reset and also causes RCM_SRS1[LOCKUP] to set.

6.2.2.9 MDM-AP system reset request

Set the System Reset Request field in the MDM-AP control register to initiate a system reset. This is the primary method for resets via the SWD interface. The system reset is held until this field is cleared.

Set the Core Hold Reset field in the MDM-AP control register to hold the core in reset as the rest of the chip comes out of system reset.

6.2.3 MCU resets

A variety of resets are generated by the MCU to reset different modules.

6.2.3.1 POR Only

The POR Only reset asserts on the POR reset source only. It resets the PMC and RTC.

The POR Only reset also causes all other reset types to occur.

6.2.3.2 Chip POR not VLLS

The Chip POR not VLLS reset asserts on POR and LVD reset sources. It resets parts of the SMC and SIM. It also resets the LPTMR.

The Chip POR not VLLS reset also causes these resets to occur: Chip POR, Chip Reset not VLLS, and Chip Reset (including Early Chip Reset).

6.2.3.3 Chip POR

The Chip POR asserts on POR, LVD, and VLLS Wakeup reset sources. It resets the Reset Pin Filter registers and parts of the SIM and MCG.

The Chip POR also causes the Chip Reset (including Early Chip Reset) to occur.

6.2.3.4 Chip Reset not VLLS

The Chip Reset not VLLS reset asserts on all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin. It resets parts of the SMC, LLWU, and other modules that remain powered during VLLS mode.

The Chip Reset not VLLS reset also causes the Chip Reset (including Early Chip Reset) to occur.

6.2.3.5 Chip Reset not VLLS3/2

The Chip Reset not VLLS3/2 reset asserts on all reset sources except a VLLS3 or VLLS2 Wakeup that does not occur via the $\overline{\text{RESET}}$ pin. It resets the radio digital logic which remains in state-retention during VLLS3 and VLLS2 modes.

6.2.3.6 Early Chip Reset

The Early Chip Reset asserts on all reset sources. It resets only the flash memory module. It negates before flash memory initialization begins ("earlier" than when the Chip Reset negates).

6.2.3.7 Chip Reset

Chip Reset asserts on all reset sources and only negates after flash initialization has completed and the $\overline{\text{RESET}}$ pin has also negated. It resets the remaining modules (the modules not reset by other reset types).

6.2.4 RESET_b pin

For all reset sources except a VLLS Wakeup that does not occur via the RESET_b pin, the RESET_b pin is driven low by the MCU for at least 128 bus clock cycles and until flash initialization has completed.

After flash initialization has completed, the RESET_b pin is released, and the internal Chip Reset negates after the RESET_b pin is pulled high. Keeping the RESET_b pin asserted externally delays the negation of the internal Chip Reset.

The RESET_b pin can be disabled by programming FTFA_FOPT[RESET_PIN_CFG] option bit to 0 (See Table 6-2). When this option is selected, there could be a short period of contention during a POR ramp where the device drives the pin low prior to establishing the setting of this option and releasing the reset function on the pin. When the RESET pin is disabled and configured as a GPIO output, it operates as a pseudo open drain output.

6.3 Boot

The information found here describes the boot sequence, including sources and options.

Some configuration information such as clock trim values stored in factory programmed flash locations is autoloaded.

6.3.1 Boot sources

The CM0+ core adds support for a programmable Vector Table Offset Register (VTOR) to relocate the exception vector table. This device supports booting from internal flash and RAM.

This device supports booting from internal flash with the reset vectors located at addresses 0x0 (initial SP_main), 0x4 (initial PC), and RAM with relocating the exception vector table to RAM.

6.3.2 FOPT boot options

The Flash Option (FOPT) register in the Flash Memory module (FTFA_FOPT) allows the user to customize the operation of the MCU at boot time. The register contains read-only bits that are loaded from the NVM's option byte in the flash configuration field. The default setting for all values in the FTFA_FOPT register is logic 1 since it is copied from the option byte residing in flash, which has all bits as logic 1 in the flash erased state. To configure for alternate settings, program the appropriate bits in the NVM option byte. The new settings will take effect on subsequent POR, VLLSx recoveries, and any system reset. For more details on programming the option byte, see the flash memory chapter.

The MCU uses the bits of FTFA_FOPT to configure the device at reset as shown in the following table.

NOTE

> Reserved bits in the option byte should be left in their default erased state of logic 1 to avoid FOPT[7:0] = 0x00 which is not valid, and is treated as FOPT[7:0] =0xFF.

Table 6-2. Flash Option Register (FTFA_FOPT) bit definitions

Bit Num	Field	Value	Definition
7-6	Reserved	Reserved for future expansion.	
5	FAST_INIT	Selects initialization speed on POR, VLLSx, and any system reset .	
		0	Slower initialization: The flash initialization will be slower with the benefit of reduced average current during this time. The duration of the recovery will be controlled by the clock divider selection determined by the LPBOOT setting.
		1	Fast Initialization: The flash has faster recoveries at the expense of higher current during these times.

Table continues on the next page...

Table 6-2. Flash Option Register (FTFA_FOPT) bit definitions (continued)

Bit Num	Field	Value	Definition
3	RESET_PIN_CFG	Enables/disables control for the RESET pin.	
		0	$\overline{\text{RESET}}$ pin is disabled following a POR and cannot be enabled as reset function. When this option is selected, there could be a short period of contention during a POR ramp where the device drives the pinout low prior to establishing the setting of this option and releasing the reset function on the pin. This bit is preserved through system resets and low-power modes. When $\overline{\text{RESET}}$ pin function is disabled, it cannot be used as a source for low-power mode wake-up. **NOTE:** When the reset pin has been disabled and security has been enabled by means of the FSEC register, a mass erase can be performed only by setting both the Mass Erase and System Reset Request fields in the MDM-AP register.
		1	$\overline{\text{RESET}}$ pin is dedicated. The port is configured with pullup enabled, open drain, passive filter enabled.
2	NMI_DIS	Enables/disables control for the NMI function.	
		0	NMI interrupts are always blocked. The associated pin continues to default to $\overline{\text{NMI}}$ pin controls with internal pullup enabled. When $\overline{\text{NMI}}$ pin function is disabled, it cannot be used as a source for low-power mode wake-up.
		1	$\overline{\text{NMI}}$ pin/interrupts reset default to enabled.
1	Reserved	Reserved for future expansion.	
4,0	LPBOOT	Controls the reset value of OUTDIV1 value in SIM_CLKDIV1 register. Larger divide value selections produce lower average power consumption during POR, VLLSx recoveries and reset sequencing and after reset exit. The recovery times are also extended if the FAST_INIT option is not selected.	
		00	Core and system clock divider (OUTDIV1) is 0x7 (divide by 8).
		01	Core and system clock divider (OUTDIV1) is 0x3 (divide by 4).
		10	Core and system clock divider (OUTDIV1) is 0x1 (divide by 2).
		11	Core and system clock divider (OUTDIV1) is 0x0 (divide by 1).

6.3.3 Boot sequence

At power up, the on-chip regulator holds the system in a POR state until the input supply exceeds the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating voltage as determined by the LVD. The Reset Controller logic then controls a sequence to exit reset.

1. A system reset is held on internal logic, the $\overline{\text{RESET}}$ pin is driven out low, and the MCGis enabled in its default clocking mode.
2. Required clocks are enabled (system clock, flash clock, and any bus clocks that do not have clock gate control reset to disabled).

3. The system reset on internal logic continues to be held, but the Flash module is released from reset and begins initialization operation while the Reset Control logic continues to drive the $\overline{\text{RESET}}$ pin out low.
4. Early in reset sequencing, the NVM option byte is read and stored to the FOPT register of the Flash Memory module (FTFA_FOPT). If the bits associated with FTFA_FOPT[LPBOOT] are programmed for an alternate clock divider reset value, the system/core clock is switched to a slower clock speed. If FTFA_FOPT[FAST_INIT] is programmed clear, the flash initialization switches to slower clock resulting longer recovery times.
5. When flash Initialization completes, the $\overline{\text{RESET}}$ pin is released. If $\overline{\text{RESET}}$ continues to be asserted (an indication of a slow rise time on the $\overline{\text{RESET}}$ pin or external drive in low), the system continues to be held in reset. Once the $\overline{\text{RESET}}$ pin is detected high, the core clock is enabled and the system is released from reset.
6. When the system exits reset, the processor sets up the stack, program counter (PC), and link register (LR). The processor reads the start SP (SP_main) from vector-table offset 0. The core reads the start PC from vector-table offset 4. LR is set to 0xFFFF_FFFF. The next sequence of events depends on the $\overline{\text{NMI}}$ input and FTFA_FOPT[NMI_DIS] (See Table 6-2) :
   - If the $\overline{\text{NMI}}$input is high or the NMI function is disabled in FTFA_FOPT, the CPU begins execution at the PC location.
   - If the $\overline{\text{NMI}}$ input is low and the NMI function is enabled in FTFA_FOPT, this results in an NMI interrupt. The processor executes an Exception Entry and reads the NMI interrupt handler address from vector-table offset 8. The CPU begins execution at the NMI interrupt handler.

Subsequent system resets follow this same reset flow.

Chapter 7
Power Management

This section describes the various chip power modes and the functionality of the individual modules in these modes.

7.1 Introduction

This chapter describes the chip power distribution as well as the various chip power modes and functionality of the individual modules in these modes.

The power architecture for this device is based on the PMC and a DCDC converter which can operate in a buck, boost or bypass configuration.

The DCDC is described in the DCDC.

7.2 Clocking Modes

This sections describes the various clocking modes supported on this device.

7.2.1 Partial Stop

Partial Stop is a clocking option that can be taken instead of entering STOP mode and is configured in the SMC Stop Control Register (SMC_STOPCTRL). The Stop mode is only partially entered, which leaves some additional functionality alive at the expense of higher power consumption. Partial Stop can be entered from either Run mode or VLP Run mode.

When configured for PSTOP2, only the core and system clocks are gated and the bus clock remains active. The bus masters and bus slaves clocked by the system clock enter Stop mode, but the bus slaves clocked by bus clock remain in Run (or VLP Run) mode. The clock generators in the MCG and the on-chip regulator in the PMC also remain in

Run (or VLP Run) mode. Exit from PSTOP2 can be initiated by a reset, an asynchronous interrupt from a bus master or bus slave clocked by the system clock, or a synchronous interrupt from a bus slave clocked by the bus clock. If configured, a DMA request (using the asynchronous DMA wakeup) can also be used to exit Partial Stop for the duration of a DMA transfer before the device is transitioned back into PSTOP2.

When configured for PSTOP1, both the system clock and bus clock are gated. All bus masters and bus slaves enter Stop mode, but the clock generators in the MCG and the on-chip regulator in the PMC remain in Run (or VLP Run) mode. Exit from PSTOP1 can be initiated by a reset or an asynchronous interrupt from a bus master or bus slave. If configured, an asynchronous DMA request can also be used to exit Partial Stop for the duration of a DMA transfer before the device is transitioned back into PSTOP1.

PSTOP1 is functionally similar to STOP mode, but offers faster wakeup at the expense of higher power consumption. Another benefit is that it keeps all of the MCG clocks enabled, which can be useful for some of the asynchronous peripherals that can remain functional in Stop modes.

7.2.2 DMA Wakeup

The DMA can be configured to wakeup the device on a DMA request whenever it is placed in stop mode. The wakeup is configured per DMA channel and is supported in Compute Operation, PSTOP, STOP and VLPS low power modes.

When a DMA wakeup is detected in PSTOP, STOP or VLPS then the device will initiate a normal exit from the low power mode. This can include restoring the on-chip regulator and internal power switches, enabling the clock generators in the MCG, enabling the system and bus clocks (but not the core clock) and negating the stop mode signal to the bus masters and bus slaves. The only difference is that the CPU will remain in the low power mode with the CPU clock disabled.

During Compute Operation, a DMA wakeup will initiate a normal exit from Compute Operation. This includes enabling the clocks and negating the stop mode signal to the bus masters and bus slaves. The core clock always remains enabled during Compute Operation.

Since the DMA wakeup will enable the clocks and negate the stop mode signals to all bus masters and slaves, software needs to ensure that bus masters and slaves that are not involved with the DMA wakeup and transfer remain in a known state. That can be accomplished by disabling the modules before entry into the low power mode or by setting the Doze enable bit in selected modules.

Once the DMA request that initiated the wakeup negates and the DMA completes the current transfer, the device will transition back into the original low power mode. This includes requesting all non-CPU bus masters to enter Stop mode and then requesting bus slaves to enter Stop mode. In STOP and VLPS modes the MCG and PMC would then also enter their appropriate modes.

NOTE

If the requested DMA transfer cannot cause the DMA request to negate then the device will remain in a higher power state until the low power mode is fully exited.

An enabled DMA wakeup can cause an aborted entry into the low power mode, if the DMA request asserts during the stop mode entry sequence (or reentry if the request asserts during a DMA wakeup) and can cause the SMC to assert its Stop Abort flag. Once the DMA wakeup completes, entry into the low power mode will restart.

An interrupt that occurs during a DMA wakeup will cause an immediate exit from the low power mode (this is optional for Compute Operation) without impacting the DMA transfer.

A DMA wakeup can be generated by either a synchronous DMA request (supported in PSTOP2 or during the stop mode entry sequence) or an asynchronous DMA request (supported in all other low power modes). Not all peripherals can generate an asynchronous DMA request in stop modes, although in general if a peripheral can generate synchronous DMA requests and also supports asynchronous interrupts in stop modes, then it can generate an asynchronous DMA request.

7.2.3 Compute Operation

Compute Operation is an execution or compute-only mode of operation that keeps the CPU enabled with full access to the SRAM and Flash read port, but places all other bus masters and bus slaves into their stop mode. Compute Operation can be enabled in either Run mode or VLP Run mode.

NOTE

Do not enter any stop mode without first exiting Compute Operation.

Because Compute Operation reuses the stop mode logic (including the staged entry with bus masters disabled before bus slaves), any bus master or bus slave that can remain functional in stop mode also remains functional in Compute Operation, including generation of asynchronous interrupts and DMA requests. When enabling Compute Operation in Run mode, module functionality for bus masters and slaves is the equivalent of STOP mode. When enabling Compute Operation in VLP Run mode, module functionality for bus masters and slaves is the equivalent of VLPS mode. The MCG, PMC, SRAM and Flash read port are not affected by Compute Operation, although the Flash register interface is disabled.

During Compute Operation, the AIPS peripheral space is disabled and attempted accesses generate bus errors. The private peripheral space remains accessible during Compute Operation, including the MCM, NVIC, IOPORT and SysTick. Although access to the GPIO registers via the IOPORT is supported, the GPIO port data input registers do not return valid data since clocks are disabled to the Port Control and Interrupt modules. By writing to the GPIO port data output registers, it is possible to control those GPIO ports that are configured as output pins.

Compute Operation is controlled by the CPO register in the MCM, which is only accessible to the CPU. Setting or clearing the CPOREQ bit in the MCM initiates entry or exit into Compute Operation. Compute Operation can also be configured to exit automatically on detection of an interrupt, which is required in order to service most interrupts. Only the core system interrupts (exceptions, including NMI and SysTick) and any edge sensitive interrupts can be serviced without exiting Compute Operation.

When entering Compute Operation, the CPOACK status bit indicates when entry has completed. When exiting Compute Operation in Run mode, the CPOACK status bit negates immediately. When exiting Compute Operation in VLP Run mode, the exit is delayed to allow the PMC to handle the change in power consumption. This delay means the CPOACK bit is polled to determine when the AIPS peripheral space can be accessed without generating a bus error.

The DMA wakeup is also supported during Compute Operation and causes the CPOACK status bit to clear and the AIPS peripheral space to be accessible for the duration of the DMA wakeup. At the completion of the DMA wakeup, the device transitions back into Compute Operation.

7.2.4 Peripheral Doze

Several peripherals support a peripheral Doze mode, where a register bit can be used to disable the peripheral for the duration of a low power mode. The Flash can also be placed in a low power state during Peripheral Doze via a register bit in the SIM.

Peripheral Doze is defined to include all of the modes of operation listed below.

- The CPU is in wait mode.

- The CPU is in stop mode, including the entry sequence and for the duration of a DMA wakeup.
- The CPU is in Compute Operation, including the entry sequence and for the duration of a DMA wakeup.

Peripheral Doze can therefore be used to disable selected bus masters or slaves for the duration of WAIT or VLPW mode. It can also be used to disable selected bus slaves immediately on entry into any stop mode (or Compute Operation), instead of waiting for the bus masters to acknowledge the entry as part of the stop entry sequence. Finally, it can be used to disable selected bus masters or slaves that should remain inactive during a DMA wakeup.

If the Flash is not being accessed during WAIT and PSTOP modes, then the Flash Doze mode can be used to reduce power consumption, at the expense of a slightly longer wakeup when executing code and vectors from Flash. It can also be used to reduce power consumption during Compute Operation when executing code and vectors from SRAM.

7.2.5 Clock Gating

To conserve power, the clocks to most modules can be turned off using the SCGCx registers in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module. Prior to initializing a module, set the corresponding bit in the SCGCx register to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution and SIM chapters.

7.3 Power modes

The power management controller (PMC) provides multiple power options to allow the user to optimize power consumption for the level of functionality needed.

Depending on the stop requirements of the user application, a variety of stop modes are available that provide state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. The following table compares the various power modes available.

For each run mode there is a corresponding wait and stop mode. Wait modes are similar to ARM sleep modes. Stop modes (VLPS, STOP) are similar to ARM sleep deep mode. The very low power run (VLPR) operating mode can drastically reduce runtime power when the maximum bus frequency is not required to handle the application needs.

The three primary modes of operation are run, wait and stop. The WFI instruction invokes both wait and stop modes for the chip. The primary modes are augmented in a number of ways to provide lower power based on application needs.

Table 7-1. Power modes (At 25 deg C)

Power mode	Description	CPU recovery method	Radio
Normal Run (all peripherals clock off)	Allows maximum performance of chip.	—	Radio can be active
Normal Wait - via WFI	Allows peripherals to function, while allowing CPU to go to sleep reducing power.	Interrupt	
Normal Stop - via WFI	Places chip in static state. Lowest power mode that retains all registers while maintaining LVD protection.	Interrupt	
PStop2 (Partial Stop 2)	Core and system clocks are gated. Bus clock remains active. Masters and slaves clocked by bus clock reamin in Run or VLPRun mode. The clock generators in MCG and the on-chip regulator in the PMC also remain in Run or VLPRun mode.	Interrupt	
PStop1 (Partial Stop 1)	Core, system clocks and bus clock are gated. All bus masters and slaves enter Stop mode. The clock generators in MCG and the on-chip regulator in the PMC also remain in Run or VLPRun mode.	Interrupt	
VLPR (Very Low Power Run) (all peripherals off)	Reduced frequency (1MHz) Flash access mode, regulator in low power mode, LVD off. Internal oscillator can provide low power 4 MHz source for core. (Values @2MHz core/ 1MHz bus and flash, module off, execution from flash). Biasing is disabled when DCDC is configured for continuous mode in VLPR/W	—	Radio operation is possible only when DCDC is configured for continuous mode.[1] However, there may be insufficient MIPS with a 4MHz MCU to support much in the way of radio operation.
VLPW (Very Low Power Wait) - via WFI (all peripherals off)	Similar to VLPR, with CPU in sleep to further reduce power. (Values @4MHz core/ 1MHz bus, module off) Biasing is disabled when DCDC is configured for continous mode in VLPR/W	Interrupt	
VLPS (Very Low Power Stop) via WFI	Places MCU in static state with LVD operation off. Lowest power mode with ADC and all pin interrupts functional. LPTMR, RTC, CMP, TSI can be operational. Biasing is disabled when DCDC is configured for continuous mode in VLPS	Interrupt	
LLS3 (Low Leakage Stop)	State retention power mode. LLWU, LPTMR, RTC, CMP, TSI can be operational. All of the radio Sea of Gates(SOG) logic is in state retention	Wakeup Interrupt	Radio SOG is in state retention in LLSx. The BTLL DSM[2] logic can be active using the 32kHz clock
LLS2 (Low Leakage Stop)	State retention power mode. LLWU, LPTMR, RTC, CMP, TSI can be operational. Only 4KBytes of RAM retained.All of the radio SOG logic is in state retention	Wakeup Interrupt	
VLLS3 (Very Low Leakage Stop3)	Full SRAM retention. LLWU, LPTMR, RTC, CMP, TSI can be operational. All of the radio SOG logic is in state retention	Wakeup Reset	Radio SOG is in state retention in VLLS3/2. The BTLL DSM logic can be active using the 32kHz clock

Table continues on the next page...

Table 7-1. Power modes (At 25 deg C) (continued)

Power mode	Description	CPU recovery method	Radio
VLLS2 (Very Low Leakage Stop2)	Partial SRAM retention. 4KBytes of RAM retained. LLWU, LPTMR, RTC, CMP, TSI can be operational.All of the radio SOG logic is in state retention	Wakeup Reset	
VLLS1 (Very Low Leakage Stop1) with RTC + 32kHz OSC	All SRAM powered off. The 32-byte system register file remains powered for customer-critical data. LLWU, LPTMR, RTC, CMP can be operational. Radio logic is power gated.	Wakeup Reset	Radio operation not supported. The Radio SOG is power-gated in VLLS1/0. Radio state is lost at VLLS1 and lower power states
VLLS1 (Very Low Leakage Stop1) with LPTMR + LPO	All SRAM powered off. The 32-byte system register file remains powered for customer-critical data. LLWU, LPTMR, RTC, CMP, TSI can be operational.	Wakeup Reset	
VLLS0 (Very Low Leakage Stop0) with Brown-out Detection	VLLS0 is not supported with DCDC The 32-byte system register file remains powered for customer-critical data. Disable all analog modules in PMC and retains I/O state and DGO state. LPO disabled, POR brown-out detection enabled, Pin interrupt only. Radio logic is power gated.	Wakeup Reset	Radio operation not supported. The Radio digital is power-gated in VLLS1/0
VLLS0 (Very Low Leakage Stop0)	VLLS0 is not supported with DCDC buck/boost configuration but is supported with bypass configuration The 32-byte system register file remains powered for customer-critical data. Disable all analog modules in PMC and retains I/O state and DGO state. LPO disabled, POR brown-out detection disabled, Pin interrupt only. Radio logic is power gated.	Wakeup Reset	

1. Biasing is disabled, but the Flash is in a low power mode for VLPx, so this configuration can realize some power savings over use of Run/Wait/Stop
2. DSM refers to BTLL's deepsleep mode. DSM does not refer to the ARM sleep deep mode.

7.4 Entering and exiting power modes

The WFI instruction invokes wait and stop modes for the chip. The processor exits the low-power mode via an interrupt. For LLS and VLLS modes, the wakeup sources are limited to LLWU generated wakeups, NMI pin, or $\overline{\text{RESET}}$ pin assertions. When the NMI pin or $\overline{\text{RESET}}$ pin have been disabled through associated FOPT settings, then these pins are ignored as wakeup sources. The wake-up flow from VLLSx is always through reset.

NOTE

The WFE instruction can have the side effect of entering a low-power mode, but that is not its intended usage. See ARM documentation for more on the WFE instruction.

On VLLSx recoveries, the I/O pins continue to be held in a static state after code execution begins, allowing software to reconfigure the system before unlocking the I/O. RAM is retained in VLLS3 and VLLS2 only.

7.5 Module Operation in Low Power Modes

The following table illustrates the functionality of each module while the chip is in each of the low power modes. The standard behavior is shown with some exceptions for Compute Operation (CPO) and Partial Stop2 (PSTOP2).

(Debug modules are discussed separately; see Debug in low-power modes.) Number ratings (such as 4 MHz and 1 Mbps) represent the maximum frequencies or maximum data rates per mode. Also, these terms are used:

- FF = Full functionality. In VLPR and VLPW the system frequency is limited, but if a module does not have a limitation in its functionality, it is still listed as FF.
- Async operation = Fully functional with alternate clock source, provided the selected clock source remains enabled
- static = Module register states and associated memories are retained.
- powered = Memory is powered to retain contents.
- low power = Memory is powered to retain contents in a lower power state
- OFF = Modules are powered off; module is in reset state upon wakeup. For clocks, OFF means disabled.
- wakeup = Modules can serve as a wakeup source for the chip.

Table 7-2. Module operation in low power modes

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
Core modules						
NVIC	FF	FF	static	static	static	OFF
System modules						
Mode Controller	FF	FF	FF	FF	FF	FF
LLWU[1]	static	static	static	static	FF	FF[2]
PMC Regulator	DCDC configured for continuous mode : ON Otherwise: low power	DCDC configured for continuous mode : ON Otherwise: low power	ON	DCDC configured for continuous mode : ON Otherwise: low power	low power	low power in VLLS3, OFF in VLLS0/1
LVD	disabled	disabled	ON	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/3, optionally

Table continues on the next page...

Table 7-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
						disabled in VLLS0[3]
DMA	FF Async operation in CPO	FF	Async operation	Async operation	static	OFF
Watchdog	FF static in CPO	FF	static FF in PSTOP2	static	static	OFF
Clocks						
1kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/3, OFF in VLLS0
Reference oscillator (OSC)[4]	Should be OFF, except when DCDC configured for continuousmode (see footnote)	Should be OFF, except when DCDC configured for continuousmode (see footnote)	ON as needed	Should be OFF, except when DCDC configured for continuousmode (see footnote)	OFF	OFF
32kHz oscillator (OSC32K)	optional	optional	optional	optional	optional	OFF in VLLS0[5]
MCG	4 MHz IRC	4 MHz IRC	static - MCGIRCLK optional	static - MCGIRCLK optional	static - no clock output	OFF
Core clock	4 MHz max	OFF	OFF	OFF	OFF	OFF
Platform clock	4 MHz max	4 MHz max	OFF	OFF	OFF	OFF
System clock	4 MHz max OFF in CPO	4 MHz max	OFF	OFF	OFF	OFF
Bus clock	1 MHz max OFF in CPO	1 MHz max	OFF 24 MHz max in PSTOP2 from RUN 1 MHz max in PSTOP2 from VLPR	OFF	OFF	OFF
Memory and memory interfaces						
Flash	1 MHz max access - no program No register access in CPO	low power	low power	low power	OFF	OFF
SRAM_U and SRAM_L	low power	low power	low power	low power	low power in LLS3, partially OFF in LLS2	low power in VLLS3, partially OFF in VLLS2, all OFF in VLLS0/1
System Register File	powered	powered	powered	powered	powered	powered
Communication interfaces						

Table continues on the next page...

Table 7-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
UART0 (LPUART)	1 Mbps Async operation in CPO	1 Mbps	Async operation FF in PSTOP2	Async operation	static	OFF
DSPI1/0	1Mbit	1Mbit	static	static	static	OFF
I^{2}C0	50 kbps static, address match wakeup in CPO	50 kbps	static, address match wakeup FF in PSTOP2	static, address match wakeup	static	OFF
I^{2}C1	100 kbps static, address match wakeup in CPO	100 kbps	static, address match wakeup	static, address match wakeup	static	OFF
CMT	FF static in CPO	FF	static	static	static	OFF
Timers						
TPMx	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	static	OFF
PIT	FF static in CPO	FF	static	static	static	OFF
LPTMR	FF	FF	Async operation FF in PSTOP2	Async operation	Async operation	Async operation[6]
RTC	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	Async operation	Async operation[7]
General Purpose Analog						
16-bit ADC	FF ADC internal clock only in CPO	FF	ADC internal clock only FF in PSTOP2	ADC internal clock only	static	OFF
CMP[8]	FF HS or LS compare in CPO	FF	HS or LS compare FF in PSTOP2	HS or LS compare	LS compare	LS compare in VLLS1/2/3, OFF in VLLS0
6-bit DAC (in CMP)	FF static in CPO	FF	static FF in PSTOP2	static	static	static, OFF in VLLS0
12-bit DAC	FF static in CPO	FF	static FF in PSTOP2	static	static	static
Human-machine interfaces						
GPIO	FF IOPORT write only in CPO	FF	static output, wakeup input FF in PSTOP2	static output, wakeup input	static, pins latched	OFF, pins latched
TSI[9]	FF	Async operation	Async operation	Async operation	Async operation	Async operation

Table continues on the next page...

Table 7-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
	Async operation in CPO					
Security						
AESA	FF static in CPO	FF	static	static	static	OFF
TRNG	FF static in CPO	FF	static	static	static	OFF
Radio						
Bluetooth LE Link Layer (BTLL)	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM) (see footnote)	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM) (see footnote)	FF	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM) (see footnote)	static (state retention) DSM timer can be active	VLLS3/2: static (state retention) DSM timer can be active VLLS1/0: OFF
ZigBee link layer	DCDC configured for continuous mode : FF Otherwise: static	DCDC configured for continuous mode : FF Otherwise: static	FF	DCDC configured for continuous mode : FF Otherwise: static	static (state retention)	VLLS3/2: static (state retention) VLLS1/0: OFF
PHY_DIG	PHY_DIG operation follows the same behavior as described in the rows above for BTLL and ZigBee.					
Reference OSC	Refer to the "Reference OSC" entry in the "Clocks" section of this table					
Other Radio Analog	DCDC configured for continuous mode : Can be enabled by Radio TSM Otherwise: should be OFF	DCDC configured for continuous mode : Can be enabled by Radio TSM Otherwise: should be OFF	Can be enabled by Radio TSM	DCDC configured for continuous mode : Can be enabled by Radio TSM Otherwise: should be OFF	OFF	OFF

1. Using the LLWU module, the external pins available for this chip do not require the associated peripheral function to be enabled. It only requires the function controlling the pin (GPIO or peripheral) to be configured as an input to allow a transition to occur to the LLWU.
2. Since LPO clock source is disabled, filters will be bypassed during VLLS0.
3. The STOPCTRL[PORPO] bit in the SMC module controls this option.
4. The reference oscillator is not usable in VLPx modes except when when the DCDC is configured to use its normal (continuous) power mode in VLPx, in which case biasing is disabled
5. The 32KHz oscillator can be bypassed, in which case the 32kHz EXTAL_32K clock can be used in VLLS0
6. LPO clock source is not available in VLLS0. In VLLS0 it must be configured for bypass (external clock) operation. Pulse counting is available in all modes.
7. In VLLS0 the only clock option for the RTC is the 32kHz bypass clock (EXTAL32K)
8. CMP in stop or VLPS supports high speed or low speed external pin to pin or external pin to DAC compares. CMP in LLS or VLLSx only supports low speed external pin to pin or external pin to DAC compares. Windowed, sampled & filtered modes of operation are not available while in stop, VLPS, LLS, or VLLSx modes.
9. TSI wakeup from all low power modes is limited to a single selectable pin.

Chapter 8
Security

This section provides an overview of flash security and details the effects of security on non-flash modules.

8.1 Introduction

This device implements security based on the mode selected from the flash module.

The following sections provide an overview of flash security and details the effects of security on non-flash modules.

8.2 Flash security

The flash module provides security information to the MCU based on the state held by FTFA_FSEC[SEC]. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes FTFA_FSEC using data read from the security byte of the flash configuration field.

NOTE

> The security features apply only to external accesses: debug. CPU accesses to the flash are not affected by the status of FTFA_FSEC.

In the unsecured state, all flash commands are available on the programming interfaces either from the debug port (SWD) or user code execution. When the flash is secured (FTFA_FSEC[SEC] = 00, 01, or 11), the programmer interfaces are only allowed to launch mass erase operations. Additionally, in this mode, the debug port has no access to memory locations.

8.3 Security interactions with other modules

The flash security settings are used by the system to determine what resources are available. The following sections describe the interactions between modules and the flash security settings or the impact that the flash security has on non-flash modules.

8.3.1 Security interactions with Debug

When flash security is active, the SWD port cannot access the memory resources of the MCU.

Although most debug functions are disabled, the debugger can write to the Flash Mass Erase in Progress field of the MDM-AP Control register to trigger a mass erase (Erase All Blocks) command. A mass erase via the debugger is allowed even when some memory locations are protected.

When mass erase is disabled, mass erase via the debugger is blocked.

8.3.2 Firmware Distribution Protection

KW4x will implement a Flash protection scheme designed to enable the distribution of Firmware to third parties from the factory. The protection mechanism allows firmware in the flash to be marked as execute-only on a per-segment basis.

Refer to chapter FTFA for Flash protection information

8.4 Security Peripherals

KW4x includes an AES Accelerator (AESA) and a True Random Generator (TRNG) to support wireless security protocols.

Chapter 9
Debug

This section describes the debug architecture and components of this device.

9.1 Introduction

Debug of this device is based on the ARM CoreSight™ architecture and is configured to provide the maximum flexibility as allowed by the restrictions of the pinout and other available resources.

It provides register and memory accessibility from the external debugger interface, basic run/halt control plus 2 breakpoints and 2 watchpoints.

Only one debug interface is supported:

- Serial Wire Debug (SWD)

9.2 Debug port pin descriptions

The debug port pins default after POR to their SWD functionality.

Table 9-1. Serial wire debug pin description

Pin name	Type	Description
SWD_CLK	Input	Serial Wire Clock This pin is the clock for debug logic when in the Serial Wire Debug mode. This pin is pulled down internally.
SWD_DIO	Input / Output	Serial Wire Debug Data Input/Output The SWD_DIO pin is used by an external debug tool for communication and device control. This pin is pulled up internally.

9.3 SWD status and control registers

Through the ARM Debug Access Port (DAP), the debugger has access to the status and control elements, implemented as registers on the DAP bus as shown in the figure found here.

These registers provide additional control and status for low power mode recovery and typical run-control scenarios. The status register bits also provide a means for the debugger to get updated status of the core without having to initiate a bus transaction across the crossbar switch, thus remaining less intrusive during a debug session.

It is important to note that these DAP control and status registers are not memory mapped within the system memory map and are only accessible via the Debug Access Port using SWD. The MDM-AP is accessible as Debug Access Port 1 with the available registers shown in this table.

Table 9-2. MDM-AP register summary

Address	Register	Description
0x0100_0000	Status	See MDM-AP Status Register
0x0100_0004	Control	See MDM-AP Control Register
0x0100_00FC	IDR	Read-only identification register that always reads as 0x001C_0020

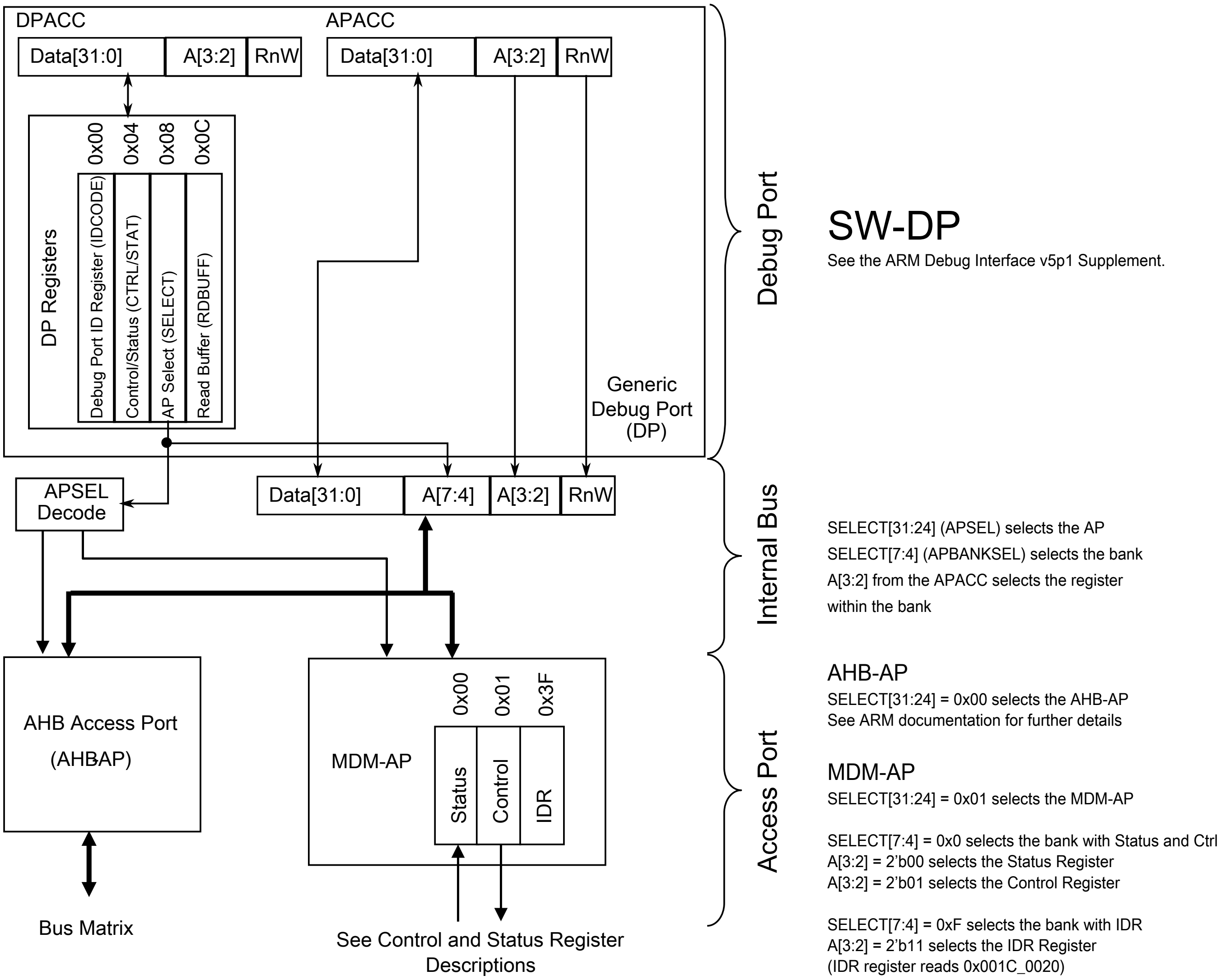

DPACC
Data[31:0]
A[3:2]
RnW
APACC
Data[31:0]
A[3:2]
RnW
DP Registers
0x00
0x04
0x08
0x0C
Debug Port ID Register (IDCODE)
Control/Status (CTRL/STAT)
AP Select (SELECT)
Read Buffer (RDBUFF)
Generic
Debug Port
(DP)
Debug Port
SW-DP
See the ARM Debug Interface v5p1 Supplement.
APSEL
Decode
Data[31:0]
A[7:4]
A[3:2]
RnW
Internal Bus
SELECT[31:24] (APSEL) selects the AP
SELECT[7:4] (APBANKSEL) selects the bank
A[3:2] from the APACC selects the register
within the bank
AHB Access Port
(AHBAP)
Bus Matrix
MDM-AP
0x00
0x01
0x3F
Status
Control
IDR
See Control and Status Register
Descriptions
Access Port
AHB-AP
SELECT[31:24] = 0x00 selects the AHB-AP
See ARM documentation for further details
MDM-AP
SELECT[31:24] = 0x01 selects the MDM-AP
SELECT[7:4] = 0x0 selects the bank with Status and Ctrl
A[3:2] = 2'b00 selects the Status Register
A[3:2] = 2'b01 selects the Control Register
SELECT[7:4] = 0xF selects the bank with IDR
A[3:2] = 2'b11 selects the IDR Register
(IDR register reads 0x001C_0020)

Figure 9-1. MDM AP addressing

9.3.1 MDM-AP Control Register

Table 9-3. MDM-AP Control register assignments

Bit	Name	Secure[1]	Description
0	Flash Mass Erase in Progress	Y	Set to cause mass erase. Cleared by power on reset after mass erase operation completes. Together with Flash Mass Erase Acknowledge bit (AP STATUS[0]), it represents the following states: • AP CTRL[0], AP STATUS [0] = 00, Idle, waiting for mass erase by writing AP CTRL[0] = 1 • AP CTRL[0], AP STATUS [0] = 01, Waiting for acknowledgement from flash

Table continues on the next page...

Table 9-3. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
			- AP CTRL[0], AP STATUS [0] = 10, Receive ack from flash, waiting for done from flash - AP CTRL[0], AP STATUS [0] = 11, Done from flash, waiting for next AP CTRL[0] = 1 When mass erase is disabled (via MEEN and SEC settings), the erase request does not occur and the Flash Mass Erase in Progress bit continues to assert until the next system reset. **NOTE:** When the reset pin has been disabled and security has been enabled by means of the FSEC register, a mass erase can be performed only by setting both the mass erase and system reset request bits in the MDM-AP register.
1	Debug Disable	N	Set to disable debug. Clear to allow debug operation. When set, it overrides the C_DEBUGEN bit within the DHCSR and force disables Debug logic.
2	Debug Request	N	Set to force the core to halt. If the core is in a Stop or Wait mode, this bit can be used to wake the core and transition to a halted state.
3	System Reset Request	Y	Set to force a system reset. The system remains held in reset until this bit is cleared.
4	Core Hold Reset	N	Configuration bit to control core operation at the end of system reset sequencing. 0 Normal operation: Release the core from reset along with the rest of the system at the end of system reset sequencing. 1 Suspend operation: Hold the core in reset at the end of reset sequencing. Once the system enters this suspended state, clearing this control bit immediately releases the core from reset and CPU operation begins.
5	VLLSx Debug Request (VLLDBGREQ)	N	Set to configure the system to be held in reset after the next recovery from a VLLSx mode. This bit is ignored on a VLLS wakeup via the Reset pin. During a VLLS wakeup via the Reset pin, the system can be held in reset by holding the reset pin asserted allowing the debugger to reinitialize the debug modules. This bit holds the system in reset when VLLSx modes are exited to allow the debugger time to re-initialize debug IP before the debug session continues. The Mode Controller captures this bit logic on entry to VLLSx modes. Upon exit from VLLSx modes, the Mode Controller will hold the system in reset until VLLDBGACK is asserted. VLLDBGREQ clears automatically due to the POR reset generated as part of the VLLSx recovery.
6	VLLSx Debug Acknowledge (VLLDBGACK)	N	Set to release a system being held in reset following a VLLSx recovery This bit is used by the debugger to release the system reset when it is being held on VLLSx mode exit. The debugger re-initializes all debug IP and then assert this control bit to allow the Mode Controller to release the system from reset and allow CPU operation to begin. VLLDBGACK is cleared by the debugger or can be left set because it clears automatically due to the POR reset generated as part of the next VLLSx recovery.

Table continues on the next page...

Table 9-3. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
7	LLS, VLLSx Status Acknowledge	N	Set this bit to acknowledge the DAP LLS and VLLS Status bits have been read. This acknowledge automatically clears the status bits. This bit is used by the debugger to clear the sticky LLS and VLLSx mode entry status bits. This bit is asserted and cleared by the debugger.
8 – 31	Reserved for future use	N	

[1] Command available in secure mode

9.3.2 MDM-AP Status Register

Table 9-4. MDM-AP Status register assignments

Bit	Name	Description
0	Flash Mass Erase Acknowledge	The Flash Mass Erase Acknowledge bit is cleared by power on reset. The bit is also cleared at launch of a mass erase command due to write of Flash Mass Erase in Progress bit in MDM AP Control Register. The Flash Mass Erase Acknowledge is set after Flash control logic has started the mass erase operation. When mass erase is disabled (via MEEN and SEC settings), an erase request due to setting of Flash Mass Erase in Progress bit is not acknowledged.
1	Flash Ready	Indicates Flash has been initialized and debugger can be configured even if system is continuing to be held in reset via the debugger.
2	System Security	Indicates the security state. When secure, the debugger does not have access to the system bus or any memory mapped peripherals. This bit indicates when the part is locked and no system bus access is possible.
3	System Reset	Indicates the system reset state. 0 System is in reset. 1 System is not in reset.
4	Reserved	
5	Mass Erase Enable	Indicates if the MCU can be mass erased or not 0 Mass erase is disabled. 1 Mass erase is enabled .
6	Backdoor Access Key Enable	Indicates if the MCU has the backdoor access key enabled. 0 Disabled 1 Enabled
7	LP Enabled	Decode of SMC_PMCTRL[STOPM] field to indicate that VLPS, LLS, or VLLSx are the selected power mode the next time the ARM Core enters Deep Sleep. 0 Low Power Stop Mode is not enabled. 1 Low Power Stop Mode is enabled.

Table continues on the next page...

Table 9-4. MDM-AP Status register assignments (continued)

Bit	Name	Description
		Usage intended for debug operation in which Run to VLPS is attempted. Per debug definition, the system actually enters the Stop state. A debugger should interpret deep sleep indication (with SLEEPDEEP and SLEEPING asserted), in conjunction with this bit asserted as the debugger-VLPS status indication.
8	Very Low Power Mode	Indicates current power mode is VLPx. This bit is not 'sticky' and should always represent whether VLPx is enabled or not. This bit is used to throttle SWD_CLK frequency up/down.
9	LLS Mode Exit	This bit indicates an exit from LLS mode has occurred. The debugger will lose communication while the system is in LLS (including access to this register). Once communication is reestablished, this bit indicates that the system had been in LLS. Since the debug modules held their state during LLS, they do not need to be reconfigured. This bit is set during the LLS recovery sequence. The LLS Mode Exit bit is held until the debugger has had a chance to recognize that LLS was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
10	VLLSx Modes Exit	This bit indicates an exit from VLLSx mode has occurred. The debugger will lose communication while the system is in VLLSx (including access to this register). Once communication is reestablished, this bit indicates that the system had been in VLLSx. Since the debug modules lose their state during VLLSx modes, they need to be reconfigured. This bit is set during the VLLSx recovery sequence. The VLLSx Mode Exit bit is held until the debugger has had a chance to recognize that a VLLS mode was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
11 – 15	Reserved for future use	Always read 0.
16	Core Halted	Indicates the core has entered Debug Halt mode
17	Core SLEEPDEEP	Indicates the core has entered a low-power mode SLEEPING==1 and SLEEPDEEP==0 indicates wait or VLPW mode. SLEEPING==1 and SLEEPDEEP==1 indicates stop or VLPS mode.
18	Core SLEEPING	
19 – 31	Reserved for future use	Always read 0.

9.4 Debug Resets

The debug system receives the following source of reset:

- System POR reset

Conversely the debug system is capable of generating system reset using the following mechanism:

- A system reset in the DAP control register which allows the debugger to hold the system in reset.

- SYSRESETREQ bit in the NVIC application interrupt and reset control register
- A system reset in the DAP control register which allows the debugger to hold the Core in reset.

9.5 Micro Trace Buffer (MTB)

The Micro Trace Buffer (MTB) provides a simple execution trace capability for the Cortex-M0+ processor.

When enabled, the MTB records changes in program flow reported by the Cortex-M0+ processor, via the execution trace interface, into a configurable region of the SRAM. Subsequently, an off-chip debugger may extract the trace information, which would allow reconstruction of an instruction flow trace. The MTB does not include any form of load/store data trace capability or tracing of any other information.

In addition to providing the trace capability, the MTB also operates as a simple AHB-Lite SRAM controller. The system bus masters, including the processor, have read/write access to all of the SRAM via the AHB-Lite interface, allowing the memory to be also used to store program and data information. The MTB simultaneously stores the trace information into an attached SRAM and allows bus masters to access the memory. The MTB ensures that trace information write accesses to the SRAM take priority over accesses from the AHB-Lite interface.

The MTB includes trace control registers for configuring and triggering the MTB functions. The MTB also supports triggering via TSTART and TSTOP control functions in the MTB DWT module.

9.6 Debug in low-power modes

In low-power modes, in which the debug modules are kept static or powered off, the debugger cannot gather any debug data for the duration of the low-power mode.

- In the case that the debugger is held static, the debug port returns to full functionality as soon as the low-power mode exits and the system returns to a state with active debug.
- In the case that the debugger logic is powered off, the debugger is reset on recovery and must be reconfigured once the low-power mode is exited.

Power mode entry logic monitors Debug Power Up and System Power Up signals from the debug port as indications that a debugger is active. These signals can be changed in RUN, VLPR, WAIT and VLPW. If the debug signal is active and the system attempts to enter Stop or VLPS, FCLK continues to run to support core register access. In these modes in which FCLK is left active the debug modules have access to core registers but not to system memory resources accessed via the crossbar.

With debug enabled, transitions from Run directly to VLPS result in the system entering Stop mode instead. Status bits within the MDM-AP Status register can be evaluated to determine this pseudo-VLPS state.

NOTE

With the debug enabled, transitions from Run --> VLPR --> VLPS are still possible.

In VLLS mode, all debug modules are powered off and reset at wakeup. In LLS mode, the debug modules retain their state but no debug activity is possible.

Going into a VLLSx mode causes all the debug controls and settings to be reset. To give time to the debugger to sync up with the HW, the MDM-AP Control register can be configured to hold the system in reset on recovery so that the debugger can regain control and reconfigure debug logic prior to the system exiting reset and resuming operation.

9.7 Debug and security

When flash security is enabled, the debug port capabilities are limited in order to prevent exploitation of secure data.

In the secure state, the debugger still has access to the status register and can determine the current security state of the device. In the case of a secure device, the debugger has the capability of only performing a mass erase operation.

Chapter 10
Port control and interrupt (PORT)

The port control and interrupt (PORT) module provides support for port control, and external interrupt functions.

10.1 Introduction

10.2 Overview

The Port Control and Interrupt (PORT) module provides support for port control, and external interrupt functions.

Most functions can be configured independently for each pin in the 32-bit port and affect the pin regardless of its pin muxing state.

There is one instance of the PORT module for each port. Not all pins within each port are implemented on a specific device.

10.2.1 Features

The PORT module has the following features:

- Pin interrupt on selected pins
  - Interrupt flag and enable registers for each pin
  - Support for edge sensitive (rising, falling, both) or level sensitive (low, high) configured per pin
  - Support for interrupt or DMA request configured per pin
  - Asynchronous wake-up in low-power modes
  - Pin interrupt is functional in all digital pin muxing modes
- Port control
  - Individual pull control fields with pullup, pulldown, and pull-disable support on selected pins

- Individual drive strength field supporting high and low drive strength on selected pins
- Individual slew rate field supporting fast and slow slew rates on selected pins
- Individual input passive filter field supporting enable and disable of the individual input passive filter on selected pins
- Individual mux control field supporting analog or pin disabled, GPIO, and up to six chip-specific digital functions
- Pad configuration fields are functional in all digital pin muxing modes.

10.2.2 Modes of operation

10.2.2.1 Run mode

In Run mode, the PORT operates normally.

10.2.2.2 Wait mode

In Wait mode, PORT continues to operate normally and may be configured to exit the Low-Power mode if an enabled interrupt is detected. DMA requests are still generated during the Wait mode, but do not cause an exit from the Low-Power mode.

10.2.2.3 Stop mode

In Stop mode, the PORT can be configured to exit the Low-Power mode via an asynchronous wake-up signal if an enabled interrupt is detected.

10.2.2.4 Debug mode

In Debug mode, PORT operates normally.

10.3 External signal description

The table found here describes the PORT external signal.

Table 10-1. Signal properties

Name	Function	I/O	Reset	Pull
PORTx[31:0]	External interrupt	I/O	0	-

NOTE

Not all pins within each port are implemented on each device.

10.4 Detailed signal description

The table found here contains the detailed signal description for the PORT interface.

Table 10-2. PORT interface—detailed signal description

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">PORTx[31:0]</td><td rowspan="3">I/O</td><td colspan="2">External interrupt.</td></tr>
<tr><td>State meaning</td><td>Asserted—pin is logic 1.
Negated—pin is logic 0.</td></tr>
<tr><td>Timing</td><td>Assertion—may occur at any time and can assert asynchronously to the system clock.
Negation—may occur at any time and can assert asynchronously to the system clock.</td></tr>
</table>

10.5 Memory map and register definition

Any read or write access to the PORT memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states.

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Pin Control Register n (PORTA_PCR0)	32	R/W	See section	10.5.1/191
4004_9004	Pin Control Register n (PORTA_PCR1)	32	R/W	See section	10.5.1/191
4004_9008	Pin Control Register n (PORTA_PCR2)	32	R/W	See section	10.5.1/191
4004_900C	Pin Control Register n (PORTA_PCR3)	32	R/W	See section	10.5.1/191
4004_9010	Pin Control Register n (PORTA_PCR4)	32	R/W	See section	10.5.1/191

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9014	Pin Control Register n (PORTA_PCR5)	32	R/W	See section	10.5.1/191
4004_9018	Pin Control Register n (PORTA_PCR6)	32	R/W	See section	10.5.1/191
4004_901C	Pin Control Register n (PORTA_PCR7)	32	R/W	See section	10.5.1/191
4004_9020	Pin Control Register n (PORTA_PCR8)	32	R/W	See section	10.5.1/191
4004_9024	Pin Control Register n (PORTA_PCR9)	32	R/W	See section	10.5.1/191
4004_9028	Pin Control Register n (PORTA_PCR10)	32	R/W	See section	10.5.1/191
4004_902C	Pin Control Register n (PORTA_PCR11)	32	R/W	See section	10.5.1/191
4004_9030	Pin Control Register n (PORTA_PCR12)	32	R/W	See section	10.5.1/191
4004_9034	Pin Control Register n (PORTA_PCR13)	32	R/W	See section	10.5.1/191
4004_9038	Pin Control Register n (PORTA_PCR14)	32	R/W	See section	10.5.1/191
4004_903C	Pin Control Register n (PORTA_PCR15)	32	R/W	See section	10.5.1/191
4004_9040	Pin Control Register n (PORTA_PCR16)	32	R/W	See section	10.5.1/191
4004_9044	Pin Control Register n (PORTA_PCR17)	32	R/W	See section	10.5.1/191
4004_9048	Pin Control Register n (PORTA_PCR18)	32	R/W	See section	10.5.1/191
4004_904C	Pin Control Register n (PORTA_PCR19)	32	R/W	See section	10.5.1/191
4004_9050	Pin Control Register n (PORTA_PCR20)	32	R/W	See section	10.5.1/191
4004_9054	Pin Control Register n (PORTA_PCR21)	32	R/W	See section	10.5.1/191
4004_9058	Pin Control Register n (PORTA_PCR22)	32	R/W	See section	10.5.1/191
4004_905C	Pin Control Register n (PORTA_PCR23)	32	R/W	See section	10.5.1/191
4004_9060	Pin Control Register n (PORTA_PCR24)	32	R/W	See section	10.5.1/191
4004_9064	Pin Control Register n (PORTA_PCR25)	32	R/W	See section	10.5.1/191
4004_9068	Pin Control Register n (PORTA_PCR26)	32	R/W	See section	10.5.1/191
4004_906C	Pin Control Register n (PORTA_PCR27)	32	R/W	See section	10.5.1/191
4004_9070	Pin Control Register n (PORTA_PCR28)	32	R/W	See section	10.5.1/191
4004_9074	Pin Control Register n (PORTA_PCR29)	32	R/W	See section	10.5.1/191
4004_9078	Pin Control Register n (PORTA_PCR30)	32	R/W	See section	10.5.1/191
4004_907C	Pin Control Register n (PORTA_PCR31)	32	R/W	See section	10.5.1/191
4004_9080	Global Pin Control Low Register (PORTA_GPCLR)	32	W (always reads 0)	0000_0000h	10.5.2/194
4004_9084	Global Pin Control High Register (PORTA_GPCHR)	32	W (always reads 0)	0000_0000h	10.5.3/194
4004_90A0	Interrupt Status Flag Register (PORTA_ISFR)	32	w1c	0000_0000h	10.5.4/195
4004_A000	Pin Control Register n (PORTB_PCR0)	32	R/W	See section	10.5.1/191
4004_A004	Pin Control Register n (PORTB_PCR1)	32	R/W	See section	10.5.1/191
4004_A008	Pin Control Register n (PORTB_PCR2)	32	R/W	See section	10.5.1/191
4004_A00C	Pin Control Register n (PORTB_PCR3)	32	R/W	See section	10.5.1/191
4004_A010	Pin Control Register n (PORTB_PCR4)	32	R/W	See section	10.5.1/191

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A014	Pin Control Register n (PORTB_PCR5)	32	R/W	See section	10.5.1/191
4004_A018	Pin Control Register n (PORTB_PCR6)	32	R/W	See section	10.5.1/191
4004_A01C	Pin Control Register n (PORTB_PCR7)	32	R/W	See section	10.5.1/191
4004_A020	Pin Control Register n (PORTB_PCR8)	32	R/W	See section	10.5.1/191
4004_A024	Pin Control Register n (PORTB_PCR9)	32	R/W	See section	10.5.1/191
4004_A028	Pin Control Register n (PORTB_PCR10)	32	R/W	See section	10.5.1/191
4004_A02C	Pin Control Register n (PORTB_PCR11)	32	R/W	See section	10.5.1/191
4004_A030	Pin Control Register n (PORTB_PCR12)	32	R/W	See section	10.5.1/191
4004_A034	Pin Control Register n (PORTB_PCR13)	32	R/W	See section	10.5.1/191
4004_A038	Pin Control Register n (PORTB_PCR14)	32	R/W	See section	10.5.1/191
4004_A03C	Pin Control Register n (PORTB_PCR15)	32	R/W	See section	10.5.1/191
4004_A040	Pin Control Register n (PORTB_PCR16)	32	R/W	See section	10.5.1/191
4004_A044	Pin Control Register n (PORTB_PCR17)	32	R/W	See section	10.5.1/191
4004_A048	Pin Control Register n (PORTB_PCR18)	32	R/W	See section	10.5.1/191
4004_A04C	Pin Control Register n (PORTB_PCR19)	32	R/W	See section	10.5.1/191
4004_A050	Pin Control Register n (PORTB_PCR20)	32	R/W	See section	10.5.1/191
4004_A054	Pin Control Register n (PORTB_PCR21)	32	R/W	See section	10.5.1/191
4004_A058	Pin Control Register n (PORTB_PCR22)	32	R/W	See section	10.5.1/191
4004_A05C	Pin Control Register n (PORTB_PCR23)	32	R/W	See section	10.5.1/191
4004_A060	Pin Control Register n (PORTB_PCR24)	32	R/W	See section	10.5.1/191
4004_A064	Pin Control Register n (PORTB_PCR25)	32	R/W	See section	10.5.1/191
4004_A068	Pin Control Register n (PORTB_PCR26)	32	R/W	See section	10.5.1/191
4004_A06C	Pin Control Register n (PORTB_PCR27)	32	R/W	See section	10.5.1/191
4004_A070	Pin Control Register n (PORTB_PCR28)	32	R/W	See section	10.5.1/191
4004_A074	Pin Control Register n (PORTB_PCR29)	32	R/W	See section	10.5.1/191
4004_A078	Pin Control Register n (PORTB_PCR30)	32	R/W	See section	10.5.1/191
4004_A07C	Pin Control Register n (PORTB_PCR31)	32	R/W	See section	10.5.1/191
4004_A080	Global Pin Control Low Register (PORTB_GPCLR)	32	W (always reads 0)	0000_0000h	10.5.2/194
4004_A084	Global Pin Control High Register (PORTB_GPCHR)	32	W (always reads 0)	0000_0000h	10.5.3/194
4004_A0A0	Interrupt Status Flag Register (PORTB_ISFR)	32	w1c	0000_0000h	10.5.4/195
4004_B000	Pin Control Register n (PORTC_PCR0)	32	R/W	See section	10.5.1/191
4004_B004	Pin Control Register n (PORTC_PCR1)	32	R/W	See section	10.5.1/191
4004_B008	Pin Control Register n (PORTC_PCR2)	32	R/W	See section	10.5.1/191
4004_B00C	Pin Control Register n (PORTC_PCR3)	32	R/W	See section	10.5.1/191
4004_B010	Pin Control Register n (PORTC_PCR4)	32	R/W	See section	10.5.1/191

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_B014	Pin Control Register n (PORTC_PCR5)	32	R/W	See section	10.5.1/191
4004_B018	Pin Control Register n (PORTC_PCR6)	32	R/W	See section	10.5.1/191
4004_B01C	Pin Control Register n (PORTC_PCR7)	32	R/W	See section	10.5.1/191
4004_B020	Pin Control Register n (PORTC_PCR8)	32	R/W	See section	10.5.1/191
4004_B024	Pin Control Register n (PORTC_PCR9)	32	R/W	See section	10.5.1/191
4004_B028	Pin Control Register n (PORTC_PCR10)	32	R/W	See section	10.5.1/191
4004_B02C	Pin Control Register n (PORTC_PCR11)	32	R/W	See section	10.5.1/191
4004_B030	Pin Control Register n (PORTC_PCR12)	32	R/W	See section	10.5.1/191
4004_B034	Pin Control Register n (PORTC_PCR13)	32	R/W	See section	10.5.1/191
4004_B038	Pin Control Register n (PORTC_PCR14)	32	R/W	See section	10.5.1/191
4004_B03C	Pin Control Register n (PORTC_PCR15)	32	R/W	See section	10.5.1/191
4004_B040	Pin Control Register n (PORTC_PCR16)	32	R/W	See section	10.5.1/191
4004_B044	Pin Control Register n (PORTC_PCR17)	32	R/W	See section	10.5.1/191
4004_B048	Pin Control Register n (PORTC_PCR18)	32	R/W	See section	10.5.1/191
4004_B04C	Pin Control Register n (PORTC_PCR19)	32	R/W	See section	10.5.1/191
4004_B050	Pin Control Register n (PORTC_PCR20)	32	R/W	See section	10.5.1/191
4004_B054	Pin Control Register n (PORTC_PCR21)	32	R/W	See section	10.5.1/191
4004_B058	Pin Control Register n (PORTC_PCR22)	32	R/W	See section	10.5.1/191
4004_B05C	Pin Control Register n (PORTC_PCR23)	32	R/W	See section	10.5.1/191
4004_B060	Pin Control Register n (PORTC_PCR24)	32	R/W	See section	10.5.1/191
4004_B064	Pin Control Register n (PORTC_PCR25)	32	R/W	See section	10.5.1/191
4004_B068	Pin Control Register n (PORTC_PCR26)	32	R/W	See section	10.5.1/191
4004_B06C	Pin Control Register n (PORTC_PCR27)	32	R/W	See section	10.5.1/191
4004_B070	Pin Control Register n (PORTC_PCR28)	32	R/W	See section	10.5.1/191
4004_B074	Pin Control Register n (PORTC_PCR29)	32	R/W	See section	10.5.1/191
4004_B078	Pin Control Register n (PORTC_PCR30)	32	R/W	See section	10.5.1/191
4004_B07C	Pin Control Register n (PORTC_PCR31)	32	R/W	See section	10.5.1/191
4004_B080	Global Pin Control Low Register (PORTC_GPCLR)	32	W (always reads 0)	0000_0000h	10.5.2/194
4004_B084	Global Pin Control High Register (PORTC_GPCHR)	32	W (always reads 0)	0000_0000h	10.5.3/194
4004_B0A0	Interrupt Status Flag Register (PORTC_ISFR)	32	w1c	0000_0000h	10.5.4/195

10.5.1 Pin Control Register n (PORTx_PCR*n*)

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset value of this device.

See the GPIO Configuration section for details on the available functions for each pin.

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	0				IRQC			
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					MUX			0	DSE	0	PFE	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	*	*	*	0	*	0	*	0	*	*	*

* Notes:

- MUX field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- DSE field: Varies by port. See the Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PFE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- SRE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PS field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.

PORTx_PCR*n* field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ISF	Interrupt Status Flag This field is read-only for pins that do not support interrupt generation. The pin interrupt configuration is valid in all digital pin muxing modes.

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
	0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 IRQC	Interrupt Configuration This field is read-only for pins that do not support interrupt generation. The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt/DMA request as follows: 0000 Interrupt Status Flag (ISF) is disabled. 0001 ISF flag and DMA request on rising edge. 0010 ISF flag and DMA request on falling edge. 0011 ISF flag and DMA request on either edge. 0100 Reserved. 0101 Reserved. 0110 Reserved. 0111 Reserved. 1000 ISF flag and Interrupt when logic 0. 1001 ISF flag and Interrupt on rising-edge. 1010 ISF flag and Interrupt on falling-edge. 1011 ISF flag and Interrupt on either edge. 1100 ISF flag and Interrupt when logic 1. 1101 Reserved. 1110 Reserved. 1111 Reserved.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 MUX	Pin Mux Control Not all pins support all pin muxing slots. Unimplemented pin muxing slots are reserved and may result in configuring the pin for a different pin muxing slot. The corresponding pin is configured in the following pin muxing slot as follows: 000 Pin disabled (Alternative 0) (analog). 001 Alternative 1 (GPIO). 010 Alternative 2 (chip-specific). 011 Alternative 3 (chip-specific). 100 Alternative 4 (chip-specific). 101 Alternative 5 (chip-specific). 110 Alternative 6 (chip-specific). 111 Alternative 7 (chip-specific).
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
6 DSE	Drive Strength Enable This field is read-only for pins that do not support a configurable drive strength. Drive strength configuration is valid in all digital pin muxing modes. 0 Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1 High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PFE	Passive Filter Enable This field is read-only for pins that do not support a configurable passive input filter. Passive filter configuration is valid in all digital pin muxing modes. 0 Passive input filter is disabled on the corresponding pin. 1 Passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input. Refer to the device data sheet for filter characteristics.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SRE	Slew Rate Enable This field is read-only for pins that do not support a configurable slew rate. Slew rate configuration is valid in all digital pin muxing modes. 0 Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1 Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable This field is read-only for pins that do not support a configurable pull resistor. Refer to the Chapter of Signal Multiplexing and Signal Descriptions for the pins that support a configurable pull resistor. Pull configuration is valid in all digital pin muxing modes. 0 Internal pullup or pulldown resistor is not enabled on the corresponding pin. 1 Internal pullup or pulldown resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select This bit is read only for pins that do not support a configurable pull resistor direction. Pull configuration is valid in all digital pin muxing modes. 0 Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1 Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

10.5.2 Global Pin Control Low Register (PORTx_GPCLR)

Only 32-bit writes are supported to this register.

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCLR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (15 through 0) bits [15:0] update with the value in GPWD. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

10.5.3 Global Pin Control High Register (PORTx_GPCHR)

Only 32-bit writes are supported to this register.

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCHR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (31 through 16) bits [15:0] update with the value in GPWD. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

10.5.4 Interrupt Status Flag Register (PORTx_ISFR)

The corresponding bit is read only for pins that do not support interrupt generation.

The pin interrupt configuration is valid in all digital pin muxing modes. The Interrupt Status Flag for each pin is also visible in the corresponding Pin Control Register, and each flag can be cleared in either location.

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_ISFR field descriptions

Field	Description
ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.

10.6 Functional description

10.6.1 Pin control

Each port pin has a corresponding Pin Control register, PORT_PCRn, associated with it.

The upper half of the Pin Control register configures the pin's capability to either interrupt the CPU or request a DMA transfer, on a rising/falling edge or both edges as well as a logic level occurring on the port pin. It also includes a flag to indicate that an interrupt has occurred. The LK bit (bit 15 of Pin Control Register PCR*n*) locks the lower 16 bits of each Pin Control register and blocks any writes to that register until the next system reset.

The lower half of the Pin Control register configures the following functions for each pin within the 32-bit port.

- Pullup or pulldown enable on selected pins
- Drive strength and slew rate configuration on selected pins
- Passive input filter enable on selected pins
- Pin Muxing mode

The functions apply across all digital pin muxing modes and individual peripherals do not override the configuration in the Pin Control register. For example, if an I^{2}C function is enabled on a pin, that does not override the pullup configuration for that pin.

When the Pin Muxing mode is configured for analog or is disabled, all the digital functions on that pin are disabled. This includes the pullup and pulldown enables, output buffer enable, input buffer enable, and passive filter enable.

The configuration of each Pin Control register is retained when the PORT module is disabled.

Whenever a pin is configured in any digital pin muxing mode, the input buffer for that pin is enabled allowing the pin state to be read via the corresponding GPIO Port Data Input Register (GPIO_PDIR) or allowing a pin interrupt or DMA request to be generated. If a pin is ever floating when its input buffer is enabled, then this can cause an increase in power consumption and must be avoided. A pin can be floating due to an input pin that is not connected or an output pin that has tri-stated (output buffer is disabled).

Enabling the internal pull resistor (or implementing an external pull resistor) will ensure a pin does not float when its input buffer is enabled; note that the internal pull resistor is automatically disabled whenever the output buffer is enabled allowing the Pull Enable bit to remain set. Configuring the Pin Muxing mode to disabled or analog will disable the pin's input buffer and results in the lowest power consumption.

10.6.2 Global pin control

The two global pin control registers allow a single register write to update the lower half of the pin control register on up to 16 pins, all with the same value.

The global pin control registers are designed to enable software to quickly configure multiple pins within the one port for the same peripheral function. However, the interrupt functions cannot be configured using the global pin control registers.

The global pin control registers are write-only registers, that always read as 0.

10.6.3 External interrupts

The external interrupt capability of the PORT module is available in all digital pin muxing modes provided the PORT module is enabled.

Each pin can be individually configured for any of the following external interrupt modes:

- Interrupt disabled, default out of reset
- Active high level sensitive interrupt
- Active low level sensitive interrupt
- Rising edge sensitive interrupt
- Falling edge sensitive interrupt
- Rising and falling edge sensitive interrupt
- Rising edge sensitive DMA request
- Falling edge sensitive DMA request
- Rising and falling edge sensitive DMA request

The interrupt status flag is set when the configured edge or level is detected on the pin . When not in Stop mode, the input is first synchronized to the bus clock to detect the configured level or edge transition.

The PORT module generates a single interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that port. The interrupt negates after the interrupt status flags for all enabled interrupts have been cleared by writing a logic 1 to the ISF flag in either the PORT_ISFR or PORT_PCRn registers.

The PORT module generates a single DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that port. The DMA request negates after the DMA transfer is completed, because that clears the interrupt status flags for all enabled DMA requests.

During Stop mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wake-up signal to exit the Low-Power mode.

Chapter 11
System Integration Module (SIM)

The System Integration Module (SIM) provides system control and chip configuration registers. This module contains several fields for selecting the clock source and dividers for various module clocks.

11.1 Introduction

The system integration module (SIM) provides system control and chip configuration registers.

11.1.1 Features

- System clocking configuration
  - System clock divide values
  - Architectural clock gating control
  - ERCLK32K clock selection
  - LPUART0 and TPM clock selection
- Flash and System RAM size configuration
- TPM external clock and input capture selection
- LPUART receive/transmit source selection/configuration

11.2 Memory map and register definition

The SIM module contains many bit fields for selecting the clock source and dividers for various module clocks.

NOTE

The SIM registers can be written only in supervisor mode. In user mode, write accesses are blocked and will result in a bus error.

NOTE

The SIM_SOPT1 register is located at a different base address than the other SIM registers.

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	0000_0000h	11.2.1/201
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_0000h	11.2.2/202
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	11.2.3/203
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	11.2.4/205
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	11.2.5/206
4004_8024	System Device Identification Register (SIM_SDID)	32	R	See section	11.2.6/208
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	F000_0030h	11.2.7/209
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0200_0182h	11.2.8/211
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	0000_0001h	11.2.9/214
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0100h	11.2.10/216
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	See section	11.2.11/217
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R/W	0F00_0000h	11.2.12/218
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	11.2.13/220
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	See section	11.2.14/221
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	See section	11.2.15/221
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	See section	11.2.16/222
4004_8100	COP Control Register (SIM_COPC)	32	R/W	0000_000Ch	11.2.17/222
4004_8104	Service COP (SIM_SRVCOP)	32	W	0000_0000h	11.2.18/224

11.2.1 System Options Register 1 (SIM_SOPT1)

NOTE

The SOPT1 register is only reset on POR or LVD.

Address: 4004_7000h base + 0h offset = 4004_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												OSC32KSEL		OSC32KOUT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT1 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 OSC32KSEL	32K Oscillator Clock Select Selects the 32 kHz clock source (ERCLK32K) for RTC and LPTMR. This field is reset only on POR/LVD. 00 32kHz oscillator (OSC32KCLK) 01 Reserved 10 RTC_CLKIN 11 LPO 1kHz
17–16 OSC32KOUT	32K oscillator clock output Outputs the ERCLK32K on the selected pin in all modes of operation (including LLS/VLLS and System Reset), overriding the existing pin mux configuration for that pin. This field is reset only on POR/LVD. 00 ERCLK32K is not output. 01 ERCLK32K is output on PTB3. 10 Reserved. 11 Reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.2 System Options Register 2 (SIM_SOPT2)

SOPT2 contains the controls for selecting many of the module clock source options on this device. See the Clock Distribution chapter for more information including clocking diagrams and definitions of device clocks.

Address: 4004_7000h base + 1004h offset = 4004_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		LPUART0SRC		TPMSRC		0		0				0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CLKOUTSEL			0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT2 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 LPUART0SRC	LPUART0 Clock Source Select Selects the clock source for the LPUART0 transmit and receive clock. 00 Clock disabled 01 MCGFLLCLK clock 10 OSCERCLK clock 11 MCGIRCLK clock
25–24 TPMSRC	TPM Clock Source Select Selects the clock source for the TPM counter clock 00 Clock disabled 01 MCGFLLCLK clock 10 OSCERCLK clock 11 MCGIRCLK clock
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 CLKOUTSEL	CLKOUT select Selects the clock to output on the CLKOUT pin. 000 OSCERCLK DIV2 001 OSCERCLK DIV4 010 Bus clock 011 LPO clock 1 kHz 100 MCGIRCLK 101 OSCERCLK DIV8 110 OSCERCLK 111 Reserved
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.3 System Options Register 4 (SIM_SOPT4)

Address: 4004_7000h base + 100Ch offset = 4004_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					TPM2CLKSEL	TPM1CLKSEL	TPM0CLKSEL	0			TPM2CH0SRC	0	TPM1CH0SRC	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT4 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 TPM2CLKSEL	TPM2 External Clock Pin Select Selects the external pin used to drive the clock to the TPM2 module.

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
	NOTE: The selected pin must also be configured for the TPM external clock function through the appropriate Pin Control Register in the Port Control module. 0 TPM2 external clock driven by TPM_CLKIN0 pin. 1 TPM2 external clock driven by TPM_CLKIN1 pin.
25 TPM1CLKSEL	TPM1 External Clock Pin Select Selects the external pin used to drive the clock to the TPM1 module. **NOTE:** The selected pin must also be configured for the TPM external clock function through the appropriate pin control register in the port control module. 0 TPM1 external clock driven by TPM_CLKIN0 pin. 1 TPM1 external clock driven by TPM_CLKIN1 pin.
24 TPM0CLKSEL	TPM0 External Clock Pin Select Selects the external pin used to drive the clock to the TPM0 module. **NOTE:** The selected pin must also be configured for the TPM external clock function through the appropriate pin control register in the port control module. 0 TPM0 external clock driven by TPM_CLKIN0 pin. 1 TPM0 external clock driven by TPM_CLKIN1 pin.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 TPM2CH0SRC	TPM2 Channel 0 Input Capture Source Select Selects the source for TPM2 channel 0 input capture. **NOTE:** When TPM2 is not in input capture mode, clear this field. 0 TPM2_CH0 signal 1 CMP0 output
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 TPM1CH0SRC	TPM1 Channel 0 Input Capture Source Select Selects the source for TPM1 channel 0 input capture. **NOTE:** When TPM1 is not in input capture mode, clear this field. 0 TPM1_CH0 signal 1 CMP0 output
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.4 System Options Register 5 (SIM_SOPT5)

Address: 4004_7000h base + 1010h offset = 4004_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	0	0	LPUART0ODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									0			0	LPUART0RXSRC	LPUART0TXSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT5 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 LPUART0ODE	LPUART0 Open Drain Enable 0 Open drain is disabled on LPUART0. 1 Open drain is enabled on LPUART0.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 LPUART0RXSRC	LPUART0 Receive Data Source Select Selects the source for the LPUART0 receive data. 0 LPUART_RX pin 1 CMP0 output
LPUART0TXSRC	LPUART0 Transmit Data Source Select

Table continues on the next page...

SIM_SOPT5 field descriptions (continued)

Field	Description
	Selects the source for the LPUART0 transmit data. 00 LPUART0_TX pin 01 LPUART0_TX pin modulated with TPM1 channel 0 output 10 LPUART0_TX pin modulated with TPM2 channel 0 output 11 Reserved

11.2.5 System Options Register 7 (SIM_SOPT7)

Address: 4004_7000h base + 1018h offset = 4004_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADC0ALTTRGEN	0		ADC0PRETRGSEL	ADC0TRGSEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT7 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADC0ALTTRGEN	ADC0 Alternate Trigger Enable Enables alternative conversion triggers for ADC0. 0 ADC ADHWT trigger comes from TPM1 channel 0 and channel1. Prior to the assertion of TPM1 channel 0, a pre-trigger pulse will be sent to ADHWTSA to initiate an ADC acquisition using ADCx_SC1A configuration and store ADC conversion in ADCx_RA Register. Prior to the assertion of TPM1 channel 1 a pre-trigger pulse will be sent to ADHWTSB to initiate an ADC acquisition using ADCx_SC1Bconfiguration and store ADC conversion in ADCx_RB Register. 1 ADC ADHWT trigger comes from a peripheral event selected by ADC0TRGSEL bits. ADC0PRETRGSEL bit will select the optional ADHWTSA or ADHWTSB select lines for choosing the ADCx_SC1x config and ADCx_Rx result regsiter to store the ADC conversion.

Table continues on the next page...

SIM_SOPT7 field descriptions (continued)

Field	Description
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ADC0PRETRGSEL	ADC0 Pretrigger Select Selects the ADC0 pre-trigger source when alternative triggers are enabled through ADC0ALTTRGEN. **NOTE:** The ADC0PRETRGSEL function is ignored if ADC0ALTTRGEN = 0. 0 Pre-trigger ADHDWTSA is selected, thus ADC0 will use ADC0_SC1A configuration for the next ADC conversion and store the result in ADC0_RA register. 1 Pre-trigger ADHDWTSB is selected, thus ADC0 will use ADC0_SC1B configuration for the next ADC conversion and store the result in ADC0_RB register.
ADC0TRGSEL	ADC0 Trigger Select Selects 1 of 16 peripherals to initiate an ADC conversion via the ADHWDT input, when ADC0ALTTRGEN =1, else is ignored by ADC0. 0000 External trigger pin input (EXTRG_IN) 0001 CMP0 output 0010 Reserved 0011 Reserved 0100 PIT trigger 0 0101 PIT trigger 1 0110 Reserved 0111 Reserved 1000 TPM0 overflow 1001 TPM1 overflow 1010 TPM2 overflow 1011 Reserved 1100 RTC alarm 1101 RTC seconds 1110 LPTMR0 trigger 1111 Radio TSM

11.2.6 System Device Identification Register (SIM_SDID)

Address: 4004_7000h base + 1024h offset = 4004_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FAMID				0		SUBFAMID		SERIESID				SRAMSIZE			
W																
Reset	*	*	*	*	0	0	*	*	0	1	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	REVID				DIEID					0			PINID			
W																
Reset	*	*	*	*	0	0	0	0	0	0	0	0	*	*	*	*

* Notes:

- FAMID field: Device specific value.
- SUBFAMID field: Device specific value.
- REVID field: Device specific value.
- PINID field: Device specific value.

SIM_SDID field descriptions

Field	Description
31–28 FAMID	Kinetis family ID Specifies the Kinetis family of the device. The FAMID will be loaded from the BLE_EN and ZIGBEE_EN IFR bits. 0010 KW20Z Family (802.15.4/ZigBee) 0011 KW30Z Family (BTLE) 0100 KW40Z Family (802.15.4/ZigBee or BTLE)
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 SUBFAMID	Kinetis Sub-Family ID. Specifies the Kinetis sub-family of the device. 00 KWx0 Subfamily 01 KWx1 Subfamily 10 KWx2 Subfamily 11 KWx3 Subfamily
23–20 SERIESID	Kinetis Series ID Specifies the Kinetis family of the device. 0101 KW family
19–16 SRAMSIZE	System SRAM Size Specifies the size of the System SRAM

Table continues on the next page...

SIM_SDID field descriptions (continued)

Field	Description
	20 KB
15–12 REVID	Device Revision Number Specifies the silicon implementation number for the device.
11–7 DIEID	Device Die Number Specifies the silicon implementation number for the device.
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PINID	Pin count Identification Specifies the pin count of the device. 0000 Reserved 0001 Reserved 0010 32-pin 0011 Reserved 0100 48-pin 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 CSP 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

11.2.7 System Clock Gating Control Register 4 (SIM_SCGC4)

Address: 4004_7000h base + 1034h offset = 4004_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1				0				0		0	0	CMP	0	0	
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	0	0	0	0	0	I2C1	I2C0	1		0	CMT	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

SIM_SCGC4 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 CMP	Comparator Clock Gate Control Controls the clock gate to the comparator module. 0 Clock disabled 1 Clock enabled
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 I2C1	I2C1 Clock Gate Control Controls the clock gate to the I^{2}C1 module. 0 Clock disabled 1 Clock enabled
6 I2C0	I2C0 Clock Gate Control Controls the clock gate to the I^{2}C0 module. 0 Clock disabled 1 Clock enabled
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
2 CMT	CMT Clock Gate Control Controls the clock gate to the CMT module. 0 Clock disabled 1 Clock enabled
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.8 System Clock Gating Control Register 5 (SIM_SCGC5)

Address: 4004_7000h base + 1038h offset = 4004_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	ZigBee	PHYDIG	BTLL	DCDC	RSIM	LTC	0		0	LPUART0	0	0		
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0		PORTC	PORTB	PORTA	1		0	TSI	0			1	LPTMR
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0

SIM_SCGC5 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

Field	Description
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 ZigBee	ZigBee Clock Gate Control This bit controls the clock gate to the ZigBee module. 0 Clock disabled 1 Clock enabled
28 PHYDIG	PHY Digital Clock Gate Control This bit controls the clock gate to the Physical Layer (PHY) Digital module. 0 Clock disabled 1 Clock enabled
27 BTLL	BTLL System Clock Gate Control This bit controls the clock gate to the BlueTooth Link Layer (BTLL) module. 0 Clock disabled 1 Clock enabled
26 DCDC	DCDC Clock Gate Control This bit controls the clock gate to the DCDC module. 0 Clock disabled 1 Clock enabled
25 RSIM	RSIM Clock Gate Control This bit controls the clock gate to the Radio SIM (RSIM) module. Always enabled.
24 LTC	LTC Clock Gate Control This bit controls the clock gate to the LTC Security module. 0 Clock disabled 1 Clock enabled
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 LPUART0	LPUART0 Clock Gate Control This bit controls the clock gate to the LPUART0 module. 0 Clock disabled 1 Clock enabled
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

Field	Description
13–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 PORTC	Port C Clock Gate Control Controls the clock gate to the Port C module. 0 Clock disabled 1 Clock enabled
10 PORTB	Port B Clock Gate Control Controls the clock gate to the Port B module. 0 Clock disabled 1 Clock enabled
9 PORTA	Port A Clock Gate Control Controls the clock gate to the Port A module. 0 Clock disabled 1 Clock enabled
8–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 TSI	TSI Access Control Controls software access to the TSI module. 0 Access disabled 1 Access enabled
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
0 LPTMR	Low Power Timer Access Control Controls software access to the Low Power Timer module. 0 Access disabled 1 Access enabled

11.2.9 System Clock Gating Control Register 6 (SIM_SCGC6)

Address: 4004_7000h base + 103Ch offset = 4004_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DAC0	0	RTC	0	ADC0	TPM2	TPM1	TPM0	PIT	0				0	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SPI1	SPI0	0		TRNG	0							DMAMUX	FTF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SIM_SCGC6 field descriptions

Field	Description
31 DAC0	DAC0 Clock Gate Control This bit controls the clock gate to the DAC0 module. 0 Clock disabled 1 Clock enabled
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 RTC	RTC Access Control Controls software access and interrupts to the RTC module. 0 Access and interrupts disabled 1 Access and interrupts enabled
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ADC0	ADC0 Clock Gate Control Controls the clock gate to the ADC0 module. 0 Clock disabled 1 Clock enabled
26 TPM2	TPM2 Clock Gate Control Controls the clock gate to the TPM2 module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
25 TPM1	TPM1 Clock Gate Control Controls the clock gate to the TPM1 module. 0 Clock disabled 1 Clock enabled
24 TPM0	TPM0 Clock Gate Control Controls the clock gate to the TPM0 module. 0 Clock disabled 1 Clock enabled
23 PIT	PIT Clock Gate Control This bit controls the clock gate to the PIT module. 0 Clock disabled 1 Clock enabled
22–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SPI1	SPI1 Clock Gate Control Controls the clock gate to the Serial Peripheral Interface (SPI1) module. 0 Clock disabled 1 Clock enabled
12 SPI0	SPI0 Clock Gate Control Controls the clock gate to the Serial Peripheral Interface (SPI0) module. 0 Clock disabled 1 Clock enabled
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 TRNG	TRNG Clock Gate Control Controls the clock gate to the Random Number Generator (TRNG) module. 0 Clock disabled 1 Clock enabled
8–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
1 DMAMUX	DMA Mux Clock Gate Control Controls the clock gate to the DMA Mux module. 0 Clock disabled 1 Clock enabled
0 FTF	Flash Memory Clock Gate Control Controls the clock gate to the flash memory. Flash reads are still supported while the flash memory is clock gated, but entry into low power modes is blocked. 0 Clock disabled 1 Clock enabled

11.2.10 System Clock Gating Control Register 7 (SIM_SCGC7)

Address: 4004_7000h base + 1040h offset = 4004_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DMA	0							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

SIM_SCGC7 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DMA	DMA Clock Gate Control Controls the clock gate to the DMA module. 0 Clock disabled 1 Clock enabled
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.11 System Clock Divider Register 1 (SIM_CLKDIV1)

NOTE

The CLKDIV1 register cannot be written to when the device is in VLPR mode.

NOTE

Reset value loaded during System Reset from FTFA_FOPT[LPBOOT]"/>).

Address: 4004_7000h base + 1044h offset = 4004_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTDIV1				0									OUTDIV4			0															
W																																
Reset	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:

- OUTDIV1 field: The reset value depends on the FTFA_FOPT[LPBOOT]. It is loaded with 0000 (divide by 1), 0001 (divide by 2), 0011 (divide by 4, or 0111 (divide by 8).

SIM_CLKDIV1 field descriptions

Field	Description
31–28 OUTDIV1	Clock 1 Output Divider value Sets the divide value for the core/system clock, as well as the bus/flash clocks. At the end of reset, it is loaded with 0000 (divide by one), 0001 (divide by two), 0011 (divide by four), or 0111 (divide by eight) depending on the setting of the FTFA_FOPT[LPBOOT]". 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
27–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_CLKDIV1 field descriptions (continued)

Field	Description
18–16 OUTDIV4	Clock 4 Output Divider value Sets the divide value for the bus and flash clock and is in addition to the System clock divide ratio. At the end of reset, it is loaded with 0001 (divide by 2). 000 Divide-by-1. 001 Divide-by-2. 010 Divide-by-3. 011 Divide-by-4. 100 Divide-by-5. 101 Divide-by-6. 110 Divide-by-7. 111 Divide-by-8.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.12 Flash Configuration Register 1 (SIM_FCFG1)

Address: 4004_7000h base + 104Ch offset = 4004_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				PFSIZE				0							
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														FLASHDOZE	FLASHDIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_FCFG1 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 PFSIZE	Program Flash Size Specifies the amount of program flash memory available on the device . Undefined values are reserved. 160 KB of program flash memory, 5 KB protection region
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FLASHDOZE	Flash Doze When set, flash memory is disabled for the duration of Doze mode. This field must be clear during VLP modes. The flash will be automatically enabled again at the end of Doze mode so interrupt vectors do not need to be relocated out of flash memory. The wake-up time from Doze mode is extended when this field is set. An attempt by the DMA or other bus master to access the flash memory when the flash is disabled will result in a bus error. 0 Flash remains enabled during Doze mode. 1 Flash is disabled for the duration of Doze mode.
0 FLASHDIS	Flash Disable Flash accesses are disabled (and generate a bus error) and the flash memory is placed in a low-power state. This field should not be changed during VLP modes. Relocate the interrupt vectors out of Flash memory before disabling the Flash. 0 Flash is enabled. 1 Flash is disabled.

11.2.13 Flash Configuration Register 2 (SIM_FCFG2)

This is read only register, any write to this register will cause transfer error.

Address: 4004_7000h base + 1050h offset = 4004_8050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAXADDR0							1	MAXADDR1						
W																
Reset	0	*	*	*	*	*	*	*	1	*	*	*	*	*	*	*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- MAXADDR0 field: Device specific value indicating amount of implemented flash.
- MAXADDR1 field: Device specific value indicating amount of implemented flash.

SIM_FCFG2 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 MAXADDR0	Max Address lock This field concatenated with 13 trailing zeros indicates the first invalid address of program flash (block 0). For example, if MAXADDR0 = 0x10, the first invalid address of program flash (block 0) is 0x0002_0000. This would be the MAXADDR0 value for a device with 128 KB program flash in flash block 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
22–16 MAXADDR1	This field concatenated with leading zeros plus the value of the MAXADDR1 field indicates the first invalid address of the second program flash block (flash block 1). For example, if MAXADDR0 = MAXADDR1 = 0x10 the first invalid address of flash block 1 is 0x2_0000 + 0x2_0000. This would be the MAXADDR1 value for a device with 256 KB program flash memory across two flash blocks.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.2.14 Unique Identification Register Mid-High (SIM_UIDMH)

Address: 4004_7000h base + 1058h offset = 4004_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																UID															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Notes:
- UID field: Device specific value.

SIM_UIDMH field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
UID	Unique Identification Unique identification for the device.

11.2.15 Unique Identification Register Mid Low (SIM_UIDML)

Address: 4004_7000h base + 105Ch offset = 4004_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Notes:
- UID field: Device specific value.

SIM_UIDML field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

11.2.16 Unique Identification Register Low (SIM_UIDL)

Address: 4004_7000h base + 1060h offset = 4004_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Notes:

- UID field: Device specific value.

SIM_UIDL field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

11.2.17 COP Control Register (SIM_COPC)

All of the bits in this register can be written only once after a reset, writing this register will also reset the COP counter.

Address: 4004_7000h base + 1100h offset = 4004_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								COPCLKSEL		COPDBGEN	COPSTPEN	COPT		COPCLKS	COPW
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

SIM_COPC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 COPCLKSEL	COP Clock Select This write-once field selects the clock source of the COP watchdog. 00 LPO clock (1 kHz) 01 MCGIRCLK 10 OSCERCLK 11 Bus clock
5 COPDBGEN	COP Debug Enable 0 COP is disabled and the counter is reset in Debug mode 1 COP is enabled in Debug mode
4 COPSTPEN	COP Stop Enable 0 COP is disabled and the counter is reset in Stop modes 1 COP is enabled in Stop modes
3–2 COPT	COP Watchdog Timeout This write-once field selects the timeout period of the COP. COPT along with the COPCLKS field define the COP timeout period. 00 COP disabled 01 COP timeout after 2^5 cycles for short timeout or 2^{13} cycles for long timeout 10 COP timeout after 2^8 cycles for short timeout or 2^{16} cycles for long timeout 11 COP timeout after 2^{10} cycles for short timeout or 2^{18} cycles for long timeout
1 COPCLKS	COP Clock Select This write-once field selects between a short timeout or a long timeout, the COP clock source is configured by COPCLKSEL. 0 COP configured for short timeout 1 COP configured for long timeout
0 COPW	COP Windowed Mode Windowed mode is supported for all COP clock sources, but only when the COP is configured for a long timeout. The COP window is opened three quarters through the timeout period and will generate a system reset if the COP is serviced outside of that time. 0 Normal mode 1 Windowed mode

11.2.18 Service COP (SIM_SRVCOP)

This is write only register, any read to this register will cause transfer error.

Address: 4004_7000h base + 1104h offset = 4004_8104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	Reserved																								SRVCOP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SRVCOP field descriptions

Field	Description
31–8 Reserved	This field is reserved.
SRVCOP	Service COP Register Write 0x55 and then 0xAA (in that order) to reset the COP timeout counter, writing any other value will generate a system reset.

Chapter 12
System Mode Controller (SMC)

12.1 Introduction

The System Mode Controller (SMC) is responsible for sequencing the system into and out of all low-power Stop and Run modes.

Specifically, it monitors events to trigger transitions between power modes while controlling the power, clocks, and memories of the system to achieve the power consumption and functionality of that mode.

This chapter describes all the available low-power modes, the sequence followed to enter/ exit each mode, and the functionality available while in each of the modes.

The SMC is able to function during even the deepest low power modes.

See AN4503: Power Management for Kinetis MCUs for further details on using the SMC.

12.2 Modes of operation

The ARM CPU has three primary modes of operation:

- Run
- Sleep
- Deep Sleep

The WFI or WFE instruction is used to invoke Sleep and Deep Sleep modes. Run, Wait, and Stop are the common terms used for the primary operating modes of Freescale microcontrollers.

The following table shows the translation between the ARM CPU modes and the Freescale MCU power modes.

ARM CPU mode	MCU mode
Sleep	Wait
Deep Sleep	Stop

Accordingly, the ARM CPU documentation refers to sleep and deep sleep, while the Freescale MCU documentation normally uses wait and stop.

In addition, Freescale MCUs also augment Stop, Wait, and Run modes in a number of ways. The power management controller (PMC) contains a run and a stop mode regulator. Run regulation is used in normal run, wait and stop modes. Stop mode regulation is used during all very low power and low leakage modes. During stop mode regulation, the bus frequencies are limited in the very low power modes.

The SMC provides the user with multiple power options. The Very Low Power Run (VLPR) mode can drastically reduce run time power when maximum bus frequency is not required to handle the application needs. From Normal Run mode, the Run Mode (RUNM) field can be modified to change the MCU into VLPR mode when limited frequency is sufficient for the application. From VLPR mode, a corresponding wait (VLPW) and stop (VLPS) mode can be entered.

Depending on the needs of the user application, a variety of stop modes are available that allow the state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. Several registers are used to configure the various modes of operation for the device.

The following table describes the power modes available for the device.

Table 12-1. Power modes

Mode	Description
RUN	The MCU can be run at full speed and the internal supply is fully regulated, that is, in run regulation. This mode is also referred to as Normal Run mode.
WAIT	The core clock is gated off. The system clock continues to operate. Bus clocks, if enabled, continue to operate. Run regulation is maintained.
STOP	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid.
VLPR	The core, system, bus, and flash clock maximum frequencies are restricted in this mode. See the Power Management chapter for details about the maximum allowable frequencies.
VLPW	The core clock is gated off. The system, bus, and flash clocks continue to operate, although their maximum frequency is restricted. See the Power Management chapter for details on the maximum allowable frequencies.
VLPS	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid.
LLS3	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low

Table continues on the next page...

Table 12-1. Power modes (continued)

Mode	Description
	leakage mode by reducing the voltage to internal logic. All system RAM contents, internal logic and I/O states are retained.
LLS2	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by reducing voltage to internal logicand powering down the system RAM2 partition. The system RAM1 partition, internal logic and I/O states are retained.[1]
VLLS3	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic. All system RAM contents are retained and I/O states are held. Internal logic states are not retained.
VLLS2	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and the system RAM2 partition. The system RAM1 partition contents are retained in this mode. Internal logic states are not retained. [2]
VLLS1	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and all system RAM. I/O states are held. Internal logic states are not retained.
VLLS0	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and all system RAM. I/O states are held. Internal logic states are not retained. The 1kHz LPO clock is disabled and the power on reset (POR) circuit can be optionally enabled using STOPCTRL[PORPO].

[1] See the devices' chip configuration details for the size and location of the system RAM partitions.
[2] See the devices' chip configuration details for the size and location of the system RAM partitions.

12.3 Memory map and register descriptions

Information about the registers related to the system mode controller can be found here.

Different SMC registers reset on different reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

NOTE

The SMC registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

NOTE

Before executing the WFI instruction, the last register written to must be read back. This ensures that all register writes associated with setting up the low power mode being entered have completed before the MCU enters the low power mode.

Failure to do this may result in the low power mode not being entered correctly.

SMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_E000	Power Mode Protection register (SMC_PMPROT)	8	R/W	00h	12.3.1/228
4007_E001	Power Mode Control register (SMC_PMCTRL)	8	R/W	00h	12.3.2/229
4007_E002	Stop Control Register (SMC_STOPCTRL)	8	R/W	03h	12.3.3/230
4007_E003	Power Mode Status register (SMC_PMSTAT)	8	R	01h	12.3.4/232

12.3.1 Power Mode Protection register (SMC_PMPROT)

This register provides protection for entry into any low-power run or stop mode. The enabling of the low-power run or stop mode occurs by configuring the Power Mode Control register (PMCTRL).

The PMPROT register can be written only once after any system reset.

If the MCU is configured for a disallowed or reserved power mode, the MCU remains in its current power mode. For example, if the MCU is in normal RUN mode and AVLP is 0, an attempt to enter VLPR mode using PMCTRL[RUNM] is blocked and PMCTRL[RUNM] remains 00b, indicating the MCU is still in Normal Run mode.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 0h offset = 4007_E000h

Bit	7	6	5	4	3	2	1	0
Read	0	0	AVLP	0	ALLS	0	AVLLS	0
Write	▨	▨		▨		▨		▨
Reset	0	0	0	0	0	0	0	0

SMC_PMPROT field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SMC_PMPROT field descriptions (continued)

Field	Description
5 AVLP	Allow Very-Low-Power Modes Provided the appropriate control bits are set up in PMCTRL, this write-once field allows the MCU to enter any very-low-power mode (VLPR, VLPW, and VLPS). 0 VLPR, VLPW, and VLPS are not allowed. 1 VLPR, VLPW, and VLPS are allowed.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ALLS	Allow Low-Leakage Stop Mode Provided the appropriate control bits are set up in PMCTRL, this write-once field allows the MCU to enter any low-leakage stop mode (LLS). 0 Any LLSx mode is not allowed 1 Any LLSx mode is allowed
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 AVLLS	Allow Very-Low-Leakage Stop Mode Provided the appropriate control bits are set up in PMCTRL, this write once bit allows the MCU to enter any very-low-leakage stop mode (VLLSx). 0 Any VLLSx mode is not allowed 1 Any VLLSx mode is allowed
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.3.2 Power Mode Control register (SMC_PMCTRL)

The PMCTRL register controls entry into low-power Run and Stop modes, provided that the selected power mode is allowed via an appropriate setting of the protection (PMPROT) register.

NOTE

This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 1h offset = 4007_E001h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	RUNM		0	STOPA	STOPM		
Write								
Reset	0	0	0	0	0	0	0	0

SMC_PMCTRL field descriptions

Field	Description
7 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
6–5 RUNM	Run Mode Control When written, causes entry into the selected run mode. Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. **NOTE:** RUNM may be set to VLPR only when PMSTAT=RUN. After being written to VLPR, RUNM should not be written back to RUN until PMSTAT=VLPR. 00 Normal Run mode (RUN) 01 Reserved 10 Very-Low-Power Run mode (VLPR) 11 Reserved
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 STOPA	Stop Aborted When set, this read-only status bit indicates an interrupt or reset occured during the previous stop mode entry sequence, preventing the system from entering that mode. This field is cleared by hardware at the beginning of any stop mode entry sequence and is set if the sequence was aborted. 0 The previous stop mode entry was successsful. 1 The previous stop mode entry was aborted.
STOPM	Stop Mode Control When written, controls entry into the selected stop mode when Sleep-Now or Sleep-On-Exit mode is entered with SLEEPDEEP=1 . Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. After any system reset, this field is cleared by hardware on any successful write to the PMPROT register. **NOTE:** When set to VLLSxor LLSx, the LLSM in the STOPCTRL register is used to further select the particular VLLSor LLS submode which will be entered. **NOTE:** When set to STOP, the PSTOPO bits in the STOPCTRL register can be used to select a Partial Stop mode if desired. 000 Normal Stop (STOP) 001 Reserved 010 Very-Low-Power Stop (VLPS) 011 Low-Leakage Stop (LLSx) 100 Very-Low-Leakage Stop (VLLSx) 101 Reserved 110 Reseved 111 Reserved

12.3.3 Stop Control Register (SMC_STOPCTRL)

The STOPCTRL register provides various control bits allowing the user to fine tune power consumption during the stop mode selected by the STOPM field.

NOTE

This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 2h offset = 4007_E002h

Bit	7	6	5	4	3	2	1	0
Read	PSTOPO		PORPO	0	Reserved	LLSM		
Write								
Reset	0	0	0	0	0	0	1	1

SMC_STOPCTRL field descriptions

Field	Description
7–6 PSTOPO	Partial Stop Option These bits control whether a Partial Stop mode is entered when STOPM=STOP. When entering a Partial Stop mode from RUN (or VLPR) mode, the PMC, MCG and flash remain fully powered, allowing the device to wakeup almost instantaneously at the expense of higher power consumption. In PSTOP2, only system clocks are gated allowing peripherals running on bus clock to remain fully functional. In PSTOP1, both system and bus clocks are gated. 00 STOP - Normal Stop mode 01 PSTOP1 - Partial Stop with both system and bus clocks disabled 10 PSTOP2 - Partial Stop with system clock disabled and bus clock enabled 11 Reserved
5 PORPO	POR Power Option This bit controls whether the POR detect circuit is enabled in VLLS0 mode. 0 POR detect circuit is enabled in VLLS0 1 POR detect circuit is disabled in VLLS0
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
LLSM	LLS or VLLS Mode Control This field controls which LLS orVLLS sub-mode to enter if STOPM = LLSx orVLLSx. 000 VLLS0 if PMCTRL[STOPM]=VLLSx, reserved if PMCTRL[STOPM]=LLSx 001 VLLS1 if PMCTRL[STOPM]=VLLSx, reserved if PMCTRL[STOPM]=LLSx 010 VLLS2 if PMCTRL[STOPM]=VLLSx, LLS2 if PMCTRL[STOPM]=LLSx 011 VLLS3 if PMCTRL[STOPM]=VLLSx, LLS3 if PMCTRL[STOPM]=LLSx 100 Reserved 101 Reserved 110 Reserved 111 Reserved

12.3.4 Power Mode Status register (SMC_PMSTAT)

PMSTAT is a read-only, one-hot register which indicates the current power mode of the system.

> **NOTE**
>
> This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 3h offset = 4007_E003h

Bit	7	6	5	4	3	2	1	0
Read	PMSTAT							
Write	▨							
Reset	0	0	0	0	0	0	0	1

SMC_PMSTAT field descriptions

Field	Description
PMSTAT	Power Mode Status **NOTE:** When debug is enabled, the PMSTAT will not update to STOP or VLPS **NOTE:** When a PSTOP mode is enabled, the PMSTAT will not update to STOP or VLPS 0000_0001 Current power mode is RUN. 0000_0010 Current power mode is STOP. 0000_0100 Current power mode is VLPR. 0000_1000 Current power mode is VLPW. 0001_0000 Current power mode is VLPS. 0010_0000 Current power mode is LLS. 0100_0000 Current power mode is VLLS. 1000_0000 Reserved

12.4 Functional description

12.4.1 Power mode transitions

The following figure shows the power mode state transitions available on the chip. Any reset always brings the MCU back to the normal RUN state.

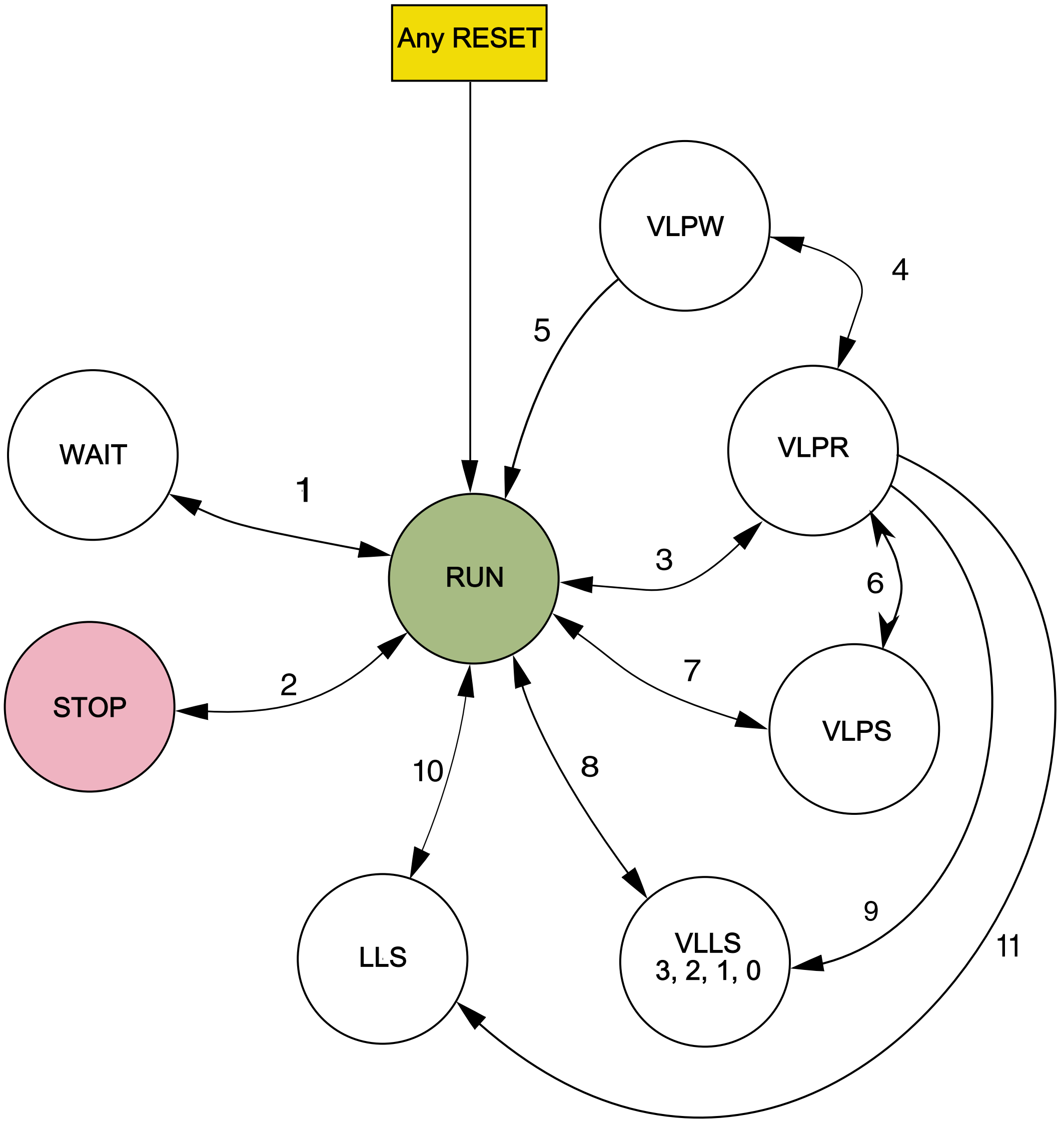


Figure 12-1. Power mode state diagram

The following table defines triggers for the various state transitions shown in the previous figure.

Table 12-2. Power mode transition triggers

Transition #	From	To	Trigger conditions
1	RUN	WAIT	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, controlled in System Control Register in ARM core. See note.
	WAIT	RUN	Interrupt or Reset
2	RUN	STOP	PMCTRL[RUNM]=00, PMCTRL[STOPM]=000[2]

Table continues on the next page...

Table 12-2. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
			Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	STOP	RUN	Interrupt or Reset
3	RUN	VLPR	The core, system, bus and flash clock frequencies and MCG clocking mode are restricted in this mode. See the Power Management chapter for the maximum allowable frequencies and MCG modes supported. Set PMPROT[AVLP]=1, PMCTRL[RUNM]=10.
	VLPR	RUN	Set PMCTRL[RUNM]=00 or Reset.
4	VLPR	VLPW	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, which is controlled in System Control Register in ARM core. See note.[1]
	VLPW	VLPR	Interrupt
5	VLPW	RUN	Reset
6	VLPR	VLPS	PMCTRL[STOPM]=000[3] or 010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	VLPR	Interrupt **NOTE:** If VLPS was entered directly from RUN (transition #7), hardware forces exit back to RUN and does not allow a transition to VLPR.
7	RUN	VLPS	PMPROT[AVLP]=1, PMCTRL[STOPM]=010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	RUN	Interrupt and VLPS mode was entered directly from RUN or Reset
8	RUN	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[LLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	VLLSx	RUN	Wakeup from enabled LLWU input source or RESET pin
9	VLPR	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[LLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

Table continues on the next page...

Table 12-2. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
10	RUN	LLSx	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, STOPCTRL[LLSM]=x (LLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	LLSx	RUN	Wakeup from enabled LLWU input source or RESET pin.
11	VLPR	LLSx	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

1. If debug is enabled, the core clock remains to support debug.
2. If PMCTRL[STOPM]=000 and STOPCTRL[PSTOPO]=01 or 10, then only a Partial Stop mode is entered instead of STOP
3. If PMCTRL[STOPM]=000 and STOPCTRL[PSTOPO]=00, then VLPS mode is entered instead of STOP. If PMCTRL[STOPM]=000 and STOPCTRL[PSTOPO]=01 or 10, then only a Partial Stop mode is entered instead of VLPS

12.4.2 Power mode entry/exit sequencing

When entering or exiting low-power modes, the system must conform to an orderly sequence to manage transitions safely.

The SMC manages the system's entry into and exit from all power modes. This diagram illustrates the connections of the SMC with other system components in the chip that are necessary to sequence the system through all power modes.

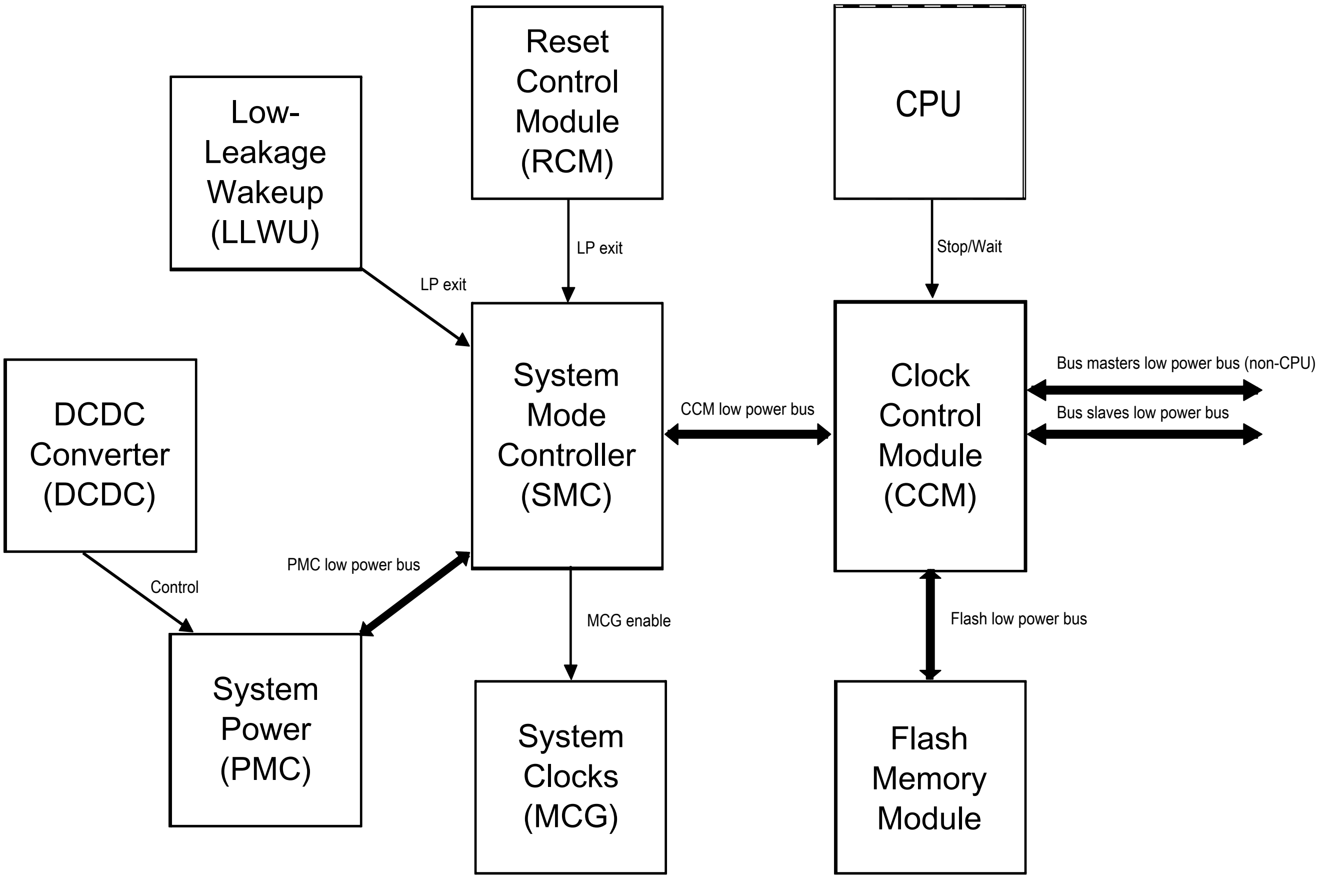


Figure 12-2. Low-power system components and connections

12.4.2.1 Stop mode entry sequence

Entry into a low-power stop mode (Stop, VLPS, LLS, VLLSx) is initiated by a CPU executing the WFI instruction. After the instruction is executed, the following sequence occurs:

1. The CPU clock is gated off immediately.
2. Requests are made to all non-CPU bus masters to enter Stop mode.
3. After all masters have acknowledged they are ready to enter Stop mode, requests are made to all bus slaves to enter Stop mode.
4. After all slaves have acknowledged they are ready to enter Stop mode, all system and bus clocks are gated off.
5. Clock generators are disabled in the MCG.
6. The on-chip regulator in the PMC and internal power switches are configured to meet the power consumption goals for the targeted low-power mode.

12.4.2.2 Stop mode exit sequence

Exit from a low-power stop mode is initiated either by a reset or an interrupt event. The following sequence then executes to restore the system to a run mode (RUN or VLPR):

1. The on-chip regulator in the PMC and internal power switches are restored.
2. Clock generators are enabled in the MCG.
3. System and bus clocks are enabled to all masters and slaves.
4. The CPU clock is enabled and the CPU begins servicing the reset or interrupt that initiated the exit from the low-power stop mode.

12.4.2.3 Aborted stop mode entry

If an interrupt or a reset occurs during a stop entry sequence, the SMC can abort the transition early and return to RUN mode without completely entering the stop mode. An aborted entry is possible only if the reset or interrupt occurs before the PMC begins the transition to stop mode regulation. After this point, the interrupt or reset is ignored until the PMC has completed its transition to stop mode regulation. When an aborted stop mode entry sequence occurs, SMC_PMCTRL[STOPA] is set to 1.

12.4.2.4 Transition to wait modes

For wait modes (WAIT and VLPW), the CPU clock is gated off while all other clocking continues, as in RUN and VLPR mode operation. Some modules that support stop-in-wait functionality have their clocks disabled in these configurations.

12.4.2.5 Transition from stop modes to Debug mode

The debugger module supports a transition from STOP, WAIT, VLPS, and VLPW back to a Halted state when the debugger has been enabled. As part of this transition, system clocking is re-established and is equivalent to the normal RUN and VLPR mode clocking configuration.

12.4.3 Run modes

The run modes supported by this device can be found here.

- Run (RUN)
- Very Low-Power Run (VLPR)

12.4.3.1 RUN mode

This is the normal operating mode for the device.

This mode is selected after any reset. When the ARM processor exits reset, it sets up the stack, program counter (PC), and link register (LR):

- The processor reads the start SP (SP_main) from vector-table offset 0x000
- The processor reads the start PC from vector-table offset 0x004
- LR is set to 0xFFFF_FFFF.

To reduce power in this mode, disable the clocks to unused modules using their corresponding clock gating control bits in the SIM's (or PCC's) registers.

12.4.3.2 Very-Low Power Run (VLPR) mode

In VLPR mode, the on-chip voltage regulator is put into a stop mode regulation state. In this state, the regulator is designed to supply enough current to the MCU over a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules using their corresponding clock gating control bits in the SIM's registers.

Before entering this mode, the following conditions must be met:

- The MCG must be configured in a mode which is supported during VLPR. See the Power Management details for information about these MCG modes.
- All clock monitors in the MCG must be disabled.
- The maximum frequencies of the system, bus, flash, and core are restricted. See the Power Management details about which frequencies are supported.
- Mode protection must be set to allow VLP modes, that is, PMPROT[AVLP] is 1.
- PMCTRL[RUNM] must be set to 10b to enter VLPR.
- Flash programming/erasing is not allowed.

NOTE

Do not increase the clock frequency while in VLPR mode, because the regulator is slow in responding and cannot manage fast load transitions. In addition, do not modify the clock source in the MCG module or any clock divider registers. Module clock enables in the SIM can be set, but not cleared.

To reenter Normal Run mode, clear PMCTRL[RUNM]. PMSTAT is a read-only status register that can be used to determine when the system has completed an exit to RUN mode. When PMSTAT=RUN, the system is in run regulation and the MCU can run at full speed in any clock mode. If a higher execution frequency is desired, poll PMSTAT until it is set to RUN when returning from VLPR mode.

Any reset always causes an exit from VLPR and returns the device to RUN mode after the MCU exits its reset flow.

12.4.4 Wait modes

This device contains two different wait modes which are listed here.

- Wait
- Very-Low Power Wait (VLPW)

12.4.4.1 WAIT mode

WAIT mode is entered when the ARM core enters the Sleep-Now or Sleep-On-Exit modes while SLEEPDEEP is cleared. The ARM CPU enters a low-power state in which it is not clocked, but peripherals continue to be clocked provided they are enabled. Clock gating to the peripheral is enabled via the SIM module.

When an interrupt request occurs, the CPU exits WAIT mode and resumes processing in RUN mode, beginning with the stacking operations leading to the interrupt service routine.

A system reset will cause an exit from WAIT mode, returning the device to normal RUN mode.

12.4.4.2 Very-Low-Power Wait (VLPW) mode

VLPW is entered by the entering the Sleep-Now or Sleep-On-Exit mode while SLEEPDEEP is cleared and the MCU is in VLPR mode.

In VLPW, the on-chip voltage regulator remains in its stop regulation state. In this state, the regulator is designed to supply enough current to the MCU over a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules by clearing the peripherals' corresponding clock gating control bits in the SIM (or PCC).

VLPR mode restrictions also apply to VLPW.

When an interrupt from VLPW occurs, the device returns to VLPR mode to execute the interrupt service routine.

A system reset will cause an exit from VLPW mode, returning the device to normal RUN mode.

12.4.5 Stop modes

This device contains a variety of stop modes to meet your application needs.

The stop modes range from:

- a stopped CPU, with all I/O, logic, and memory states retained, and certain asynchronous mode peripherals operating

to:

- a powered down CPU, with only I/O and a small register file retained, very few asynchronous mode peripherals operating, while the remainder of the MCU is powered down.

The choice of stop mode depends upon the user's application, and how power usage and state retention versus functional needs and recovery time may be traded off.

NOTE

All clock monitors must be disabled before entering these low-power modes: Stop, VLPS, VLPR, VLPW, LLSand VLLSx.

The various stop modes are selected by setting the appropriate fields in PMPROT and PMCTRL. The selected stop mode is entered during the sleep-now or sleep-on-exit entry with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The available stop modes are:

- Normal Stop (STOP)
- Very-Low Power Stop (VLPS)
- Low-Leakage Stop (LLS)
- Very-Low-Leakage Stop (VLLSx)

12.4.5.1 STOP mode

STOP mode is entered via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The MCG module can be configured to leave the reference clocks running.

A module capable of providing an asynchronous interrupt to the device takes the device out of STOP mode and returns the device to normal RUN mode. Refer to the device's Power Management chapter for peripheral, I/O, and memory operation in STOP mode. When an interrupt request occurs, the CPU exits STOP mode and resumes processing, beginning with the stacking operations leading to the interrupt service routine.

A system reset will cause an exit from STOP mode, returning the device to normal RUN mode via an MCU reset.

12.4.5.2 Very-Low-Power Stop (VLPS) mode

The two ways in which VLPS mode can be entered are listed here.

- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in VLPR mode and PMCTRL[STOPM] = 010 or 000.
- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in normal RUN mode and PMCTRL[STOPM] = 010. When VLPS is entered directly from RUN mode, exit to VLPR is disabled by hardware and the system will always exit back to RUN.

In VLPS, the on-chip voltage regulator remains in its stop regulation state as in VLPR.

A module capable of providing an asynchronous interrupt to the device takes the device out of VLPS and returns the device to VLPR mode.

A system reset will also cause a VLPS exit, returning the device to normal RUN mode.

12.4.5.3 Low-Leakage Stop (LLSx) modes

This device contains two Low-Leakage Stop modes: LLS3 and LLS2. LLS or LLSx is often used in this document to refer to both modes. All LLS modes can be entered from normal RUN or VLPR modes.

The MCU enters LLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, SLEEPDEEP is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 12-2.

In LLS, the on-chip voltage regulator is in stop regulation. Most of the peripherals are put in a state-retention mode that does not allow them to operate while in LLS.

Before entering LLS mode, the user should configure the Low-Leakage Wake-up (LLWU) module to enable the desired wake-up sources. The available wake-up sources in LLS are detailed in the chip configuration details for this device.

After wakeup from LLS, the device returns to normal RUN mode with a pending LLWU module interrupt. In the LLWU interrupt service routine (ISR), the user can poll the LLWU module wake-up flags to determine the source of the wakeup.

NOTE

> The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit Stop mode on an LLS recovery.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from LLS mode, returning the device to normal RUN mode. When LLS is exiting via the $\overline{\text{RESET}}$ pin, RCM_SRS[PIN] and RCM_SRS[WAKEUP] are set.

12.4.5.4 Very-Low-Leakage Stop (VLLSx) modes

This device contains these very low leakage modes:

- VLLS3
- VLLS2
- VLLS1
- VLLS0

VLLSx is often used in this document to refer to all of these modes.

All VLLSx modes can be entered from normal RUN or VLPR modes.

The MCU enters the configured VLLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, the SLEEPDEEP bit is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 12-2.

In VLLS, the on-chip voltage regulator is in its stop-regulation state while most digital logic is powered off.

Before entering VLLS mode, the user should configure the Low-Leakage Wake-up (LLWU) module to enable the desired wakeup sources. The available wake-up sources in VLLS are detailed in the chip configuration details for this device.

After wakeup from VLLS, the device returns to normal RUN mode with a pending LLWU interrupt. In the LLWU interrupt service routine (ISR), the user can poll the LLWU module wake-up flags to determine the source of the wake-up.

When entering VLLS, each I/O pin is latched as configured before executing VLLS. Because all digital logic in the MCU is powered off, all port and peripheral data is lost during VLLS. This information must be restored before PMC_REGSC[ACKISO] is set.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from any VLLS mode, returning the device to normal RUN mode. When exiting VLLS via the $\overline{\text{RESET}}$ pin, RCM_SRS[PIN] and RCM_SRS[WAKEUP] are set.

12.4.6 Debug in low power modes

When the MCU is secure, the device disables/limits debugger operation. When the MCU is unsecure, the ARM debugger can assert two power-up request signals:

- System power up, via SYSPWR in the Debug Port Control/Stat register
- Debug power up, via CDBGPWRUPREQ in the Debug Port Control/Stat register

When asserted while in RUN, WAIT, VLPR, or VLPW, the mode controller drives a corresponding acknowledge for each signal, that is, both CDBGPWRUPACK and CSYSPWRUPACK. When both requests are asserted, the mode controller handles attempts to enter STOP and VLPS by entering an emulated stop state. In this emulated stop state:

- the regulator is in run regulation,
- the MCG-generated clock source is enabled,
- all system clocks, except the core clock, are disabled,
- the debug module has access to core registers, and
- access to the on-chip peripherals is blocked.

No debug is available while the MCU is in LLS or VLLS modes. LLS is a state-retention mode and all debug operation can continue after waking from LLS, even in cases where system wakeup is due to a system reset event.

Entering into a VLLS mode causes all of the debug controls and settings to be powered off. To give time to the debugger to sync with the MCU, the MDM AP Control Register includes a Very-Low-Leakage Debug Request (VLLDBGREQ) bit that is set to configure the Reset Controller logic to hold the system in reset after the next recovery from a VLLS mode. This bit allows the debugger time to reinitialize the debug module before the debug session continues.

The MDM AP Control Register also includes a Very Low Leakage Debug Acknowledge (VLLDBGACK) bit that is set to release the ARM core being held in reset following a VLLS recovery. The debugger reinitializes all debug IP, and then asserts the VLLDBGACK control bit to allow the RCM to release the ARM core from reset and allow CPU operation to begin.

The VLLDBGACK bit is cleared by the debugger (or can be left set as is) or clears automatically due to the reset generated as part of the next VLLS recovery.

Chapter 13
Power Management Controller (PMC)

13.1 Introduction

The power management controller (PMC) contains the internal voltage regulator, power on reset (POR), low voltage detect system (LVD), and high voltage detect system (HVD).

See AN4503: Power Management for Kinetis MCUs for further details on using the PMC.

13.2 Features

A list of included PMC features can be found here.

- Internal voltage regulator
- Active POR providing brown-out detect
- Low-voltage detect supporting two low-voltage trip points with four warning levels per trip point

13.3 Low-voltage detect (LVD) system

This device includes a system to guard against low-voltage conditions. This protects memory contents and controls MCU system states during supply voltage variations.

The system is comprised of a power-on reset (POR) circuit and a LVD circuit with a user-selectable trip voltage: high (V_{LVDH}) or low (V_{LVDL}). The trip voltage is selected by LVDSC1[LVDV]. The LVD is disabled upon entering VLPx, LLS, and VLLSx modes.

Two flags are available to indicate the status of the low-voltage detect system:

- The Low Voltage Detect Flag in the Low Voltage Status and Control 1 Register (LVDSC1[LVDF]) operates in a level sensitive manner. LVDSC1[LVDF] is set when the supply voltage falls below the selected trip point (VLVD). LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC1[LVDF] remains set.
- The Low Voltage Warning Flag (LVWF) in the Low Voltage Status and Control 2 Register (LVDSC2[LVWF]) operates in a level sensitive manner. LVDSC2[LVWF] is set when the supply voltage falls below the selected monitor trip point (VLVW). LVDSC2[LVWF] is cleared by writing one to LVDSC2[LVWACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC2[LVWF] remains set.

13.3.1 LVD reset operation

By setting LVDSC1[LVDRE], the LVD generates a reset upon detection of a low-voltage condition. The low-voltage detection threshold is determined by LVDSC1[LVDV]. After an LVD reset occurs, the LVD system holds the MCU in reset until the supply voltage rises above this threshold. The LVD field in the SRS register of the RCM module (RCM_SRS[LVD]) is set following an LVD or power-on reset.

13.3.2 LVD interrupt operation

By configuring the LVD circuit for interrupt operation (LVDSC1[LVDIE] set and LVDSC1[LVDRE] clear), LVDSC1[LVDF] is set and an LVD interrupt request occurs upon detection of a low voltage condition. LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK].

13.3.3 Low-voltage warning (LVW) interrupt operation

The LVD system contains a Low-Voltage Warning Flag (LVWF) in the Low Voltage Detect Status and Control 2 Register to indicate that the supply voltage is approaching, but is above, the LVD voltage. The LVW also has an interrupt, which is enabled by setting LVDSC2[LVWIE]. If enabled, an LVW interrupt request occurs when LVDSC2[LVWF] is set. LVDSC2[LVWF] is cleared by writing 1 to LVDSC2[LVWACK].

LVDSC2[LVWV] selects one of the four trip voltages:

- Highest: V_{LVW4}
- Two mid-levels: V_{LVW3} and V_{LVW2}
- Lowest: V_{LVW1}

13.4 I/O retention

When in LLS mode, the I/O pins are held in their input or output state.

Upon wakeup, the PMC is re-enabled, goes through a power up sequence to full regulation, and releases the logic from state retention mode. The I/O are released immediately after a wake-up or reset event. In the case of LLS exit via a RESET pin, the I/O default to their reset state.

When in VLLS modes, the I/O states are held on a wake-up event (with the exception of wake-up by reset event) until the wake-up has been acknowledged via a write to REGSC[ACKISO]. In the case of VLLS exit via a RESET pin, the I/O are released and default to their reset state. In this case, no write to REGSC[ACKISO] is needed.

13.5 Memory map and register descriptions

Details about the PMC registers can be found here.

NOTE

Different portions of PMC registers are reset only by particular reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

The PMC registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

PMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D000	Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)	8	R/W	10h	13.5.1/248

Table continues on the next page...

PMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D001	Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)	8	R/W	00h	13.5.2/249
4007_D002	Regulator Status And Control register (PMC_REGSC)	8	R/W	04h	13.5.3/250

13.5.1 Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)

This register contains status and control bits to support the low voltage detect function. This register should be written during the reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC1 settings. To protect systems that must have LVD always on, configure the Power Mode Protection (PMPROT) register of the SMC module (SMC_PMPROT) to disallow any very low power or low leakage modes from being enabled.

See the device's data sheet for the exact LVD trip voltages.

NOTE

The LVDV bits are reset solely on a POR Only event. The register's other bits are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 0h offset = 4007_D000h

Bit	7	6	5	4	3	2	1	0
Read	LVDF	0	LVDIE	LVDRE	0		LVDV	
Write		LVDACK						
Reset	0	0	0	1	0	0	0	0

PMC_LVDSC1 field descriptions

Field	Description
7 LVDF	Low-Voltage Detect Flag This read-only status field indicates a low-voltage detect event. 0 Low-voltage event not detected 1 Low-voltage event detected

Table continues on the next page...

PMC_LVDSC1 field descriptions (continued)

Field	Description
6 LVDACK	Low-Voltage Detect Acknowledge This write-only field is used to acknowledge low voltage detection errors. Write 1 to clear LVDF. Reads always return 0.
5 LVDIE	Low-Voltage Detect Interrupt Enable Enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVDF = 1
4 LVDRE	Low-Voltage Detect Reset Enable This write-once bit enables LVDF events to generate a hardware reset. Additional writes are ignored. 0 LVDF does not generate hardware resets 1 Force an MCU reset when LVDF = 1
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LVDV	Low-Voltage Detect Voltage Select Selects the LVD trip point voltage (V_{LVD}). 00 Low trip point selected ($V_{LVD} = V_{LVDL}$) 01 High trip point selected ($V_{LVD} = V_{LVDH}$) 10 Reserved 11 Reserved

13.5.2 Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)

This register contains status and control bits to support the low voltage warning function.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC2 settings.

See the device's data sheet for the exact LVD trip voltages.

NOTE

The LVW trip voltages depend on LVWV and LVDV.

NOTE

LVWV is reset solely on a POR Only event. The other fields of the register are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 1h offset = 4007_D001h

Bit	7	6	5	4	3	2	1	0
Read	LVWF	0	LVWIE	0			LVWV	
Write		LVWACK						
Reset	0	0	0	0	0	0	0	0

PMC_LVDSC2 field descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag This read-only status field indicates a low-voltage warning event. LVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. LVWF may be 1 after power-on reset, therefore, to use LVW interrupt function, before enabling LVWIE, LVWF must be cleared by writing LVWACK first. 0 Low-voltage warning event not detected 1 Low-voltage warning event detected
6 LVWACK	Low-Voltage Warning Acknowledge This write-only field is used to acknowledge low voltage warning errors. Write 1 to clear LVWF. Reads always return 0.
5 LVWIE	Low-Voltage Warning Interrupt Enable Enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVWF = 1
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LVWV	Low-Voltage Warning Voltage Select Selects the LVW trip point voltage (V_{LVW}). The actual voltage for the warning depends on LVDSC1[LVDV]. 00 Low trip point selected ($V_{LVW} = V_{LVW1}$) 01 Mid 1 trip point selected ($V_{LVW} = V_{LVW2}$) 10 Mid 2 trip point selected ($V_{LVW} = V_{LVW3}$) 11 High trip point selected ($V_{LVW} = V_{LVW4}$)

13.5.3 Regulator Status And Control register (PMC_REGSC)

The PMC contains an internal voltage regulator. The voltage regulator design uses a bandgap reference that is also available through a buffer as input to certain internal peripherals, such as the CMP and ADC. The internal regulator provides a status bit (REGONS) indicating the regulator is in run regulation.

NOTE

This register is reset on Chip Reset Not VLLS and by reset types that trigger Chip Reset not VLLS. See the Reset section details for more information.

Address: 4007_D000h base + 2h offset = 4007_D002h

Bit	7	6	5	4	3	2	1	0
Read	0	VLPO	Reserved	0	ACKISO	REGONS	Reserved	BGBE
Write	▨			▨	w1c	▨		
Reset	0	0	0	0	0	1	0	0

PMC_REGSC field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 VLPO	VLPx Option When used in conjunction with BGEN, this bit allows additional clock sources and higher frequency operation (at the cost of higher power) to be selected during VLPx modes. 0 Operating frequencies and MCG clocking modes are restricted during VLPx modes as listed in the Power Management chapter. 1 If BGEN is also set, operating frequencies and MCG clocking modes are unrestricted during VLPx modes. Note that flash access frequency is still restricted however.
5 Reserved	This field is reserved.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ACKISO	Acknowledge Isolation Reading this field indicates whether certain peripherals and the I/O pads are in a latched state as a result of having been in a VLLS mode. Writing 1 to this field when it is set releases the I/O pads and certain peripherals to their normal run mode state. **NOTE:** After recovering from a VLLS mode, user should restore chip configuration before clearing ACKISO. In particular, pin configuration for enabled LLWU wakeup pins should be restored to avoid any LLWU flag from being falsely set when ACKISO is cleared. 0 Peripherals and I/O pads are in normal run state. 1 Certain peripherals and I/O pads are in an isolated and latched state.
2 REGONS	Regulator In Run Regulation Status This read-only field provides the current status of the internal voltage regulator. 0 Regulator is in stop regulation or in transition to/from it 1 Regulator is in run regulation
1 Reserved	This field is reserved. **NOTE:** This reserved bit must remain cleared (set to 0).
0 BGBE	Bandgap Buffer Enable

Table continues on the next page...

PMC_REGSC field descriptions (continued)

Field	Description
	Enables the bandgap buffer. 0 Bandgap buffer not enabled 1 Bandgap buffer enabled

Chapter 14
DCDC Converter (DCDC)

14.1 About this module

14.1.1 Introduction

14.1.2 Features

The DCDC module includes the following features:

- Single inductor, multiple outputs.
- Buck or Boost modes (pin selectable: CFG = VBAT → buck; CFG = 0 → boost).
- Continuous or pulsed operation (software configurable).
- Power switch input pin to allow external control of power up.
- Output signal to indicate power stability. Purpose is for the rest of the chip to use as a POR.
- Scaled battery output voltage suitable for analog-to-digital converter (ADC) utilization.
- Internal oscillator for support where the crystal oscillator is not present.

14.1.3 Block diagram

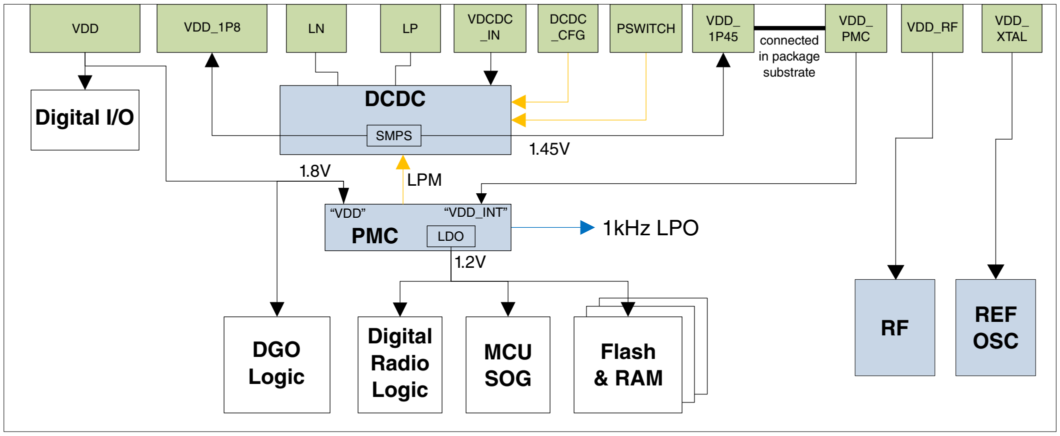


Figure 14-1. DCDC block Diagram

14.1.4 Configurations

This section shows the three DCDC module configurations.

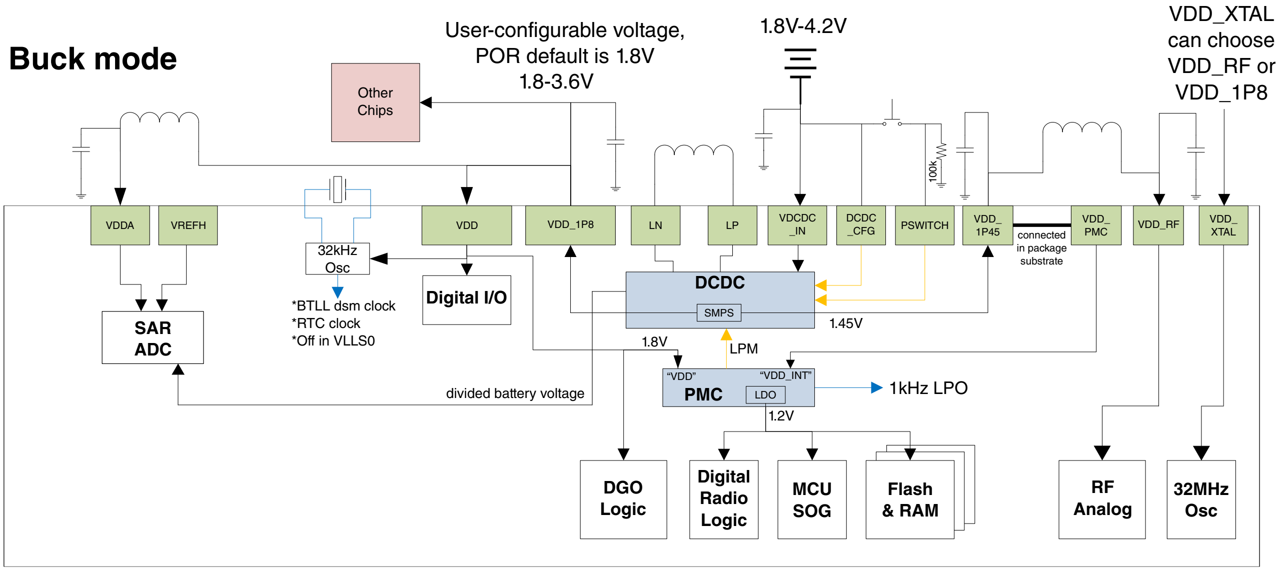


Figure 14-2. Buck Configuration

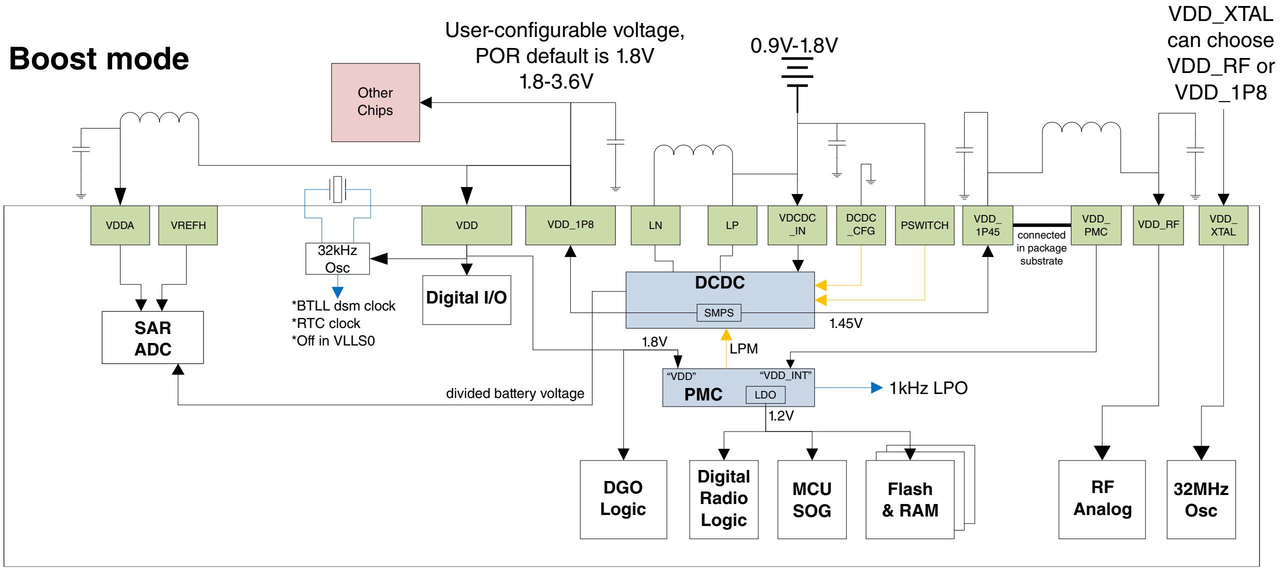


Figure 14-3. Boost Configuration

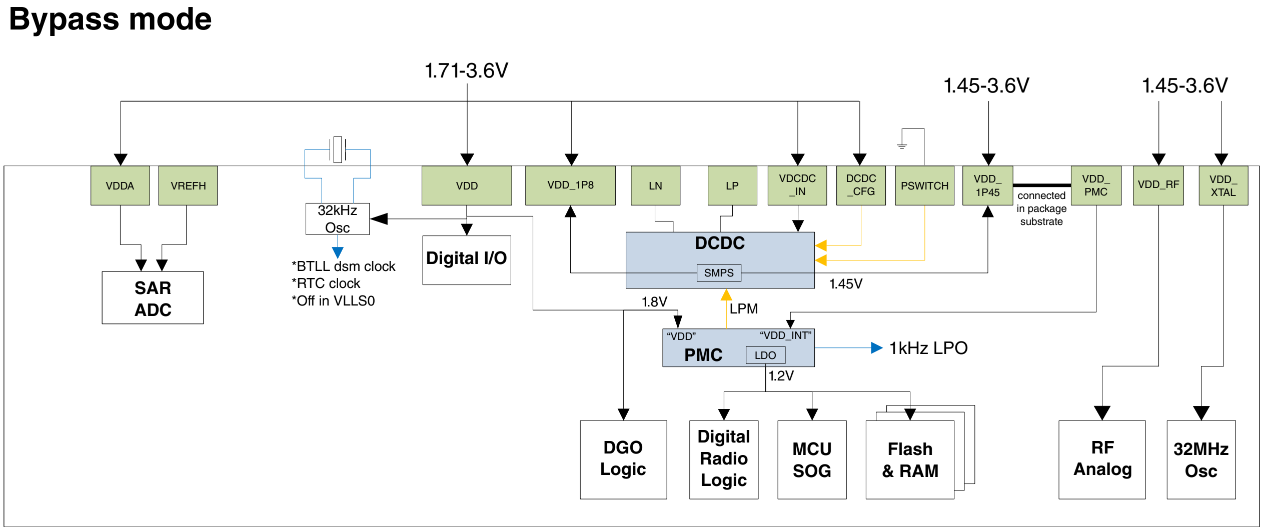


Figure 14-4. Bypass configuration

14.1.5 Functional Description

The DCDC converter module is a switching mode DC-DC converter supporting Buck, Boost and Bypass mode. It only requires single inductor to produce multiple switching outputs. The converter can be operated in continuous or pulsed mode. It is configurable through internal registers. An internal oscillator is included for use case where external oscillator is not available.

The DCDC converter produces two switching outputs in both Buck and Boost modes. They are VDD1p45 and VDD1p8, which can produce up to 25mA and 42.5mA continuous output current respectively. Typical switching frequency is 2MHz. DCDC conversion efficiency is typically 90%.

In Buck mode, it supports operation with battery voltage from 2.1V to 4.2V. In Boost mode, it supports operation with battery voltage from 0.9V to 1.8V. In Bypass mode, the DCDC converter is disabled. Individual supply signals of KW40Z need to be supplied with regulated supply accordingly.

Selection of operating mode is done by setting the pin DCDC_CFG. VDD1p45 regulated output supplies the radio block and the SOG. VDD1p8 regulated output supplies the rest of the device. VDD1p8 output can feed external circuitry with restricted load current up to around 40mA.

In Boost mode, the VDD1p45 will be 1.8V by default. Software need to measure the battery voltage with light load, before changing the output voltage. The output target voltage should be 50mV higher than battery voltage. e.g. when the battery voltage is less than 1.4V, can set the target output voltage to 1.45V. If the target voltage is set to lower than the battery voltage, it may trigger PMC low voltage reset.

PSWITCH pin can be used to wakeup the DCDC converter in buck mode. The level will be latched by the device. A toggle switch or a push button can be used with this signal for generating the wakeup signal.

14.1.6 Application Guideline

Continuous mode

Following registers setting are recommended to have better efficiency and ripple.

- DCDC_LOOPCTRL_EN_DF_HYST = 1
- DCDC_LOOPCTRL_EN_CM_HYST = 1
- DCDC_LOOPCTRL_HYST_SIGN = 1

In boost mode, POSLIMIT_BOOST_IN is set to small value by default. To limit startup voltage, set it to 0x12 after startup, to provide high current to output, especially when battery voltage is low.

Target voltage adjustment

To adjust target voltage of VDD1P8 and VDD1P45:

1. Clear DCDC_VDD1P8CTRL_DISABLE_STEP and DCDC_VDD1P45CTRL_DISABLE_STEP

2. Change target register bits DCDC_VDD1P45CTRL_TRG_BOOST, DCDC_VDD1P45CTRL_TRG_BUCK and DCDC_VDD1P8CTRL_TRG

DCDC_STS_DC_OK bit will be de-asserted after target register changes. After output voltage settling to new target value, DCDC_STS_DC_OK will be asserted.

Pulsed mode

Before entering pulsed mode, must set DCDC_VDD1P8CTRL_DISABLE_STEP and DCDC_VDD1P45CTRL_DISABLE_STEP bit to 1.

Following registers bit settings are highly recommended.

- DCDC_LOOPCTRL_EN_DF_HYST = 1
- DCDC_LOOPCTRL_EN_CM_HYST = 1
- DCDC_LOOPCTRL_HYST_SIGN = 1
- DCDC_LP_DF_CMP_ENABLE = 1

half/double FET

When current loading is low (<~5mA), it is recommended to use half FET which can improve efficiency. When current loading is high (>~40mA), it is recommended to use double FET which can improve the current drive capability and efficiency.

14.2 Memory map and register definition

14.2.1 Register Reset

All the registers will only be reset to default value after POR reset.

DCDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_A000	DCDC REGISTER 0 (DCDC_REG0)	32	R/W	0418_0000h	14.2.2/258
4005_A004	DCDC REGISTER 1 (DCDC_REG1)	32	R/W	0017_C21Ch	14.2.3/261
4005_A008	DCDC REGISTER 2 (DCDC_REG2)	32	R/W	0000_4009h	14.2.4/263
4005_A00C	DCDC REGISTER 3 (DCDC_REG3)	32	R/W	0000_A9C6h	14.2.5/265
4005_A010	DCDC REGISTER 4 (DCDC_REG4)	32	R/W	0000_0000h	14.2.6/268
4005_A018	DCDC REGISTER 6 (DCDC_REG6)	32	R/W	0000_0000h	14.2.7/269
4005_A01C	DCDC REGISTER 7 (DCDC_REG7)	32	R/W	0000_0000h	14.2.8/270

14.2.2 DCDC REGISTER 0 (DCDC_REG0)

Address: 4005_A000h base + 0h offset = 4005_A000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCDC_STS_DC_OK	VLPR_VLPW_CONFIG_DCDC_HP	VLPS_CONFIG_DCDC_HP	PSWITCH_STATUS	DCDC_XTALOK_DISABLE	PWD_CMP_OFFSET	DCDC_LESS_I	OFFSET_RSNS_LP_DISABLE	OFFSET_RSNS_LP_ADJ	HYST_LP_CMP_DISABLE	HYST_LP_COMP_ADJ	DCDC_LP_STATE_HYS_H		DCDC_LP_STATE_HYS_L		Reserved
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DCDC_VBAT_DIV_CTRL		DCDC_LP_DF_CMP_ENABLE	Reserved	Reserved	Reserved	Reserved	Reserved	DCDC_PWD_OSC_INT	DCDC_SEL_CLK	DCDC_DISABLE_AUTO_CLK_SWITCH	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCDC_REG0 field descriptions

Field	Description
31 DCDC_STS_DC_OK	Status register to indicate DCDC lock. The lock time depends on the loading and the DCDC mode. When changing output voltage target, it will take approximately (0.5ms* target change steps). In pulsed mode, it will take approximately 5ms. In startup, it will take approximately 50ms.
30 VLPR_VLPW_CONFIG_DCDC_HP	Selects behavior of DCDC in device VLPR and VLPW low power modes. Pulsed mode is a lower power mode. It can be used if the loads are small (<=0.5mA) in VLPx modes. 1'b1 DCDC works in continuous mode when SoC is in VLPR / VLPW modes. 1'b0 DCDC works in pulsed mode when SoC is in VLPR / VLPW modes.
29 VLPS_CONFIG_DCDC_HP	Selects behavior of DCDC in device VLPS low power mode. Pulsed mode is a lower power mode. It can be used if the loads are small (<=0.5mA) in VLPx modes. 1'b1 DCDC works in continuous mode when SOC is in VLPS modes. 1'b0 DCDC works in pulsed mode when SOC is in VLPS modes.
28 PSWITCH_STATUS	Status register to indicate PSWITCH status
27 DCDC_XTALOK_DISABLE	Disable xtalok detection circuit.
26 PWD_CMP_OFFSET	Power down output range comparator
25 DCDC_LESS_I	Reduce DCDC current. It will save approximately 20 µA in RUN.
24 OFFSET_RSNS_LP_DISABLE	Disable hysteresis in low power voltage sense.

Table continues on the next page...

DCDC_REG0 field descriptions (continued)

Field	Description
23 OFFSET_RSNS_LP_ADJ	Adjust hysteretic value in low power voltage sense.
22 HYST_LP_CMP_DISABLE	Disable hysteresis in low power comparator.
21 HYST_LP_COMP_ADJ	Adjust hysteretic value in low power comparator.
20–19 DCDC_LP_STATE_HYS_H	Configure the hysteretic upper threshold value in low power mode. It determines the hysteretic value of the output voltage in pulsed mode. 00 Target voltage value + 0 mV 01 Target voltage value + 25 mV 10 Target voltage value + 50 mV 11 Target voltage value + 75 mV
18–17 DCDC_LP_STATE_HYS_L	Configure the hysteretic lower threshold value in low power mode. It determines the hysteretic value of the output voltage in pulsed mode. 00 Target voltage value - 0 mV 01 Target voltage value - 25 mV 10 Target voltage value - 50 mV 11 Target voltage value - 75 mV
16 Reserved	This field is reserved. Reserved
15–12 Reserved	This field is reserved. Reserved
11–10 DCDC_VBAT_DIV_CTRL	Controls VBAT voltage divider. The divided VBAT output is input to an ADC channel which allows the battery voltage to be measured. 2'b00 OFF 2'b01 VBAT 2'b10 VBAT / 2 2'b11 VBAT / 4
9 DCDC_LP_DF_CMP_ENABLE	Enable low power differential comparators, to sense lower supply in pulsed mode. This can reduce the ripple in pulsed mode. 1'b1 DCDC compare the lower supply(relative to target value) with DCDC_LP_STATE_HYS_L. When it is lower than DCDC_LP_STATE_HYS_L, re-charge output. 1'b0 DCDC compare the common mode sense of supply(relative to target value) with DCDC_LP_STATE_HYS_L. When it is lower than DCDC_LP_STATE_HYS_L, re-charge output.
8 Reserved	This field is reserved. Reserved
7 Reserved	This field is reserved. Reserved
6 Reserved	This field is reserved. Reserved
5 Reserved	This field is reserved. Reserved

Table continues on the next page...

DCDC_REG0 field descriptions (continued)

Field	Description
4 Reserved	This field is reserved. Reserved
3 DCDC_PWD_OSC_INT	Power down internal oscillator. Only set this bit when 32M crystal oscillator is available.
2 DCDC_SEL_CLK	Select external clock for DCDC when DCDC_DISABLE_AUTO_CLK_SWITCH is set.
1 DCDC_DISABLE_AUTO_CLK_SWITCH	Disable automatic clock switch from internal oscillator to external clock.
0 Reserved	This field is reserved. Reserved

14.2.3 DCDC REGISTER 1 (DCDC_REG1)

Address: 4005_A000h base + 4h offset = 4005_A004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							DCDC_LOOPCTRL_EN_DF_HYST	DCDC_LOOPCTRL_EN_CM_HYST	DCDC_LOOPCTRL_DF_HST_THRESH	DCDC_LOOPCTRL_CM_HST_THRESH	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		POSLIMIT_BOOST_IN							POSLIMIT_BUCK_IN						
Reset	1	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0

DCDC_REG1 field descriptions

Field	Description
31–25 Reserved	This field is reserved. Reserved

Table continues on the next page...

DCDC_REG1 field descriptions (continued)

Field	Description
24 DCDC_ LOOPCTRL_EN_ DF_HYST	Enable hysteresis in switching converter differential mode analog comparators. This feature improves transient supply ripple and efficiency.
23 DCDC_ LOOPCTRL_EN_ CM_HYST	Enable hysteresis in switching converter common mode analog comparators. This feature improves transient supply ripple and efficiency.
22 DCDC_ LOOPCTRL_DF_ HST_THRESH	Enable hysteresis in switching converter differential mode analog comparators. This feature improves transient supply ripple and efficiency.
21 DCDC_ LOOPCTRL_ CM_HST_ THRESH	Enable hysteresis in switching converter common mode analog comparators. This feature improves transient supply ripple and efficiency.
20–14 Reserved	This field is reserved. Reserved.
13–7 POSLIMIT_ BOOST_IN	Upper limit duty cycle limit in DC-DC converter. This field limits the maximum VDDIO achievable for a given battery voltage, and its value may be increased if very low battery operation is met.
POSLIMIT_ BUCK_IN	Upper limit duty cycle limit in DC-DC converter. This field limits the maximum VDDIO achievable for a given battery voltage, and its value may be increased if very low battery operation is met.

14.2.4 DCDC REGISTER 2 (DCDC_REG2)

Address: 4005_A000h base + 8h offset = 4005_A008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						DCDC_BATTMONITOR_BATT_VAL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DCDC_BATTMONITOR_EN_BATADJ	Reserved	DCDC_LOOPCTRL_HYST_SIGN	Reserved	Reserved	Reserved		Reserved			Reserved				Reserved	
Reset	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1

DCDC_REG2 field descriptions

Field	Description
31–26 Reserved	This field is reserved. Reserved
25–16 DCDC_ BATTMONITOR_ BATT_VAL	Software should be configured to place the battery voltage in this register measured with an 8 mV LSB resolution through the ADC. This value is used by the DC-DC converter and must be proper configured before setting EN_BATADJ.
15 DCDC_ BATTMONITOR_ EN_BATADJ	This bit enables the DC-DC to improve efficiency and minimize ripple using the information from the BATT_VAL field. The BATT_VAL contains accurate information before setting EN_BATADJ.
14 Reserved	This field is reserved. Reserved
13 DCDC_ LOOPCTRL_ HYST_SIGN	Invert the sign of the hysteresis in DC-DC analog comparators. This bit is set when in Pulsed mode.
12 Reserved	This field is reserved. Reserved
11 Reserved	This field is reserved. Reserved

Table continues on the next page...

DCDC_REG2 field descriptions (continued)

Field	Description
10–9 Reserved	This field is reserved. Reserved
8–6 Reserved	This field is reserved. Reserved
5–2 Reserved	This field is reserved. Reserved
Reserved	This field is reserved. Reserved

14.2.5 DCDC REGISTER 3 (DCDC_REG3)

Address: 4005_A000h base + Ch offset = 4005_A00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	DCDC_VDD1P8CTRL_DISABLE_STEP	DCDC_VDD1P45CTRL_DISABLE_STEP	Reserved	Reserved	DCDC_MINPWR_HALF_FETS	DCDC_MINPWR_DOUBLE_FETS	DCDC_MINPWR_DC_HALFCLK	DCDC_MINPWR_HALF_FETS_PULSED	DCDC_MINPWR_DOUBLE_FETS_PULSED	DCDC_MINPWR_DC_HALFCLK_PULSED	DCDC_VDD1P45CTRL_ADJTN				Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCDC_VDD1P45CTRL_TRG_BOOST					DCDC_VDD1P45CTRL_TRG_BUCK					DCDC_VDD1P8CTRL_TRG					
W																
Reset	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	0

DCDC_REG3 field descriptions

Field	Description
31 Reserved	This field is reserved. Reserved
30 DCDC_VDD1P8CTRL_DISABLE_STEP	Disable stepping for VDD1P8. Must set this bit before enter low power modes.
29 DCDC_VDD1P45CTRL_DISABLE_STEP	Disable stepping for VDD1P45. Must set this bit before enter low power modes.
28 Reserved	This read-only field is reserved and always has the value 0. This field is reserved.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 DCDC_MINPWR_HALF_FETS	Use half switch FET for the continuous mode.
25 DCDC_MINPWR_DOUBLE_FETS	Use double switch FET for the continuous mode.
24 DCDC_MINPWR_DC_HALFCLK	Set DCDC clock to half frequency for the continuous mode.
23 DCDC_MINPWR_HALF_FETS_PULSED	Use half switch FET for the Pulsed mode.

Table continues on the next page...

DCDC_REG3 field descriptions (continued)

Field	Description
22 DCDC_MINPWR_DOUBLE_FETS_PULSED	Use double switch FET for the Pulsed mode.
21 DCDC_MINPWR_DC_HALFCLK_PULSED	Set DCDC clock to half frequency for the Pulsed mode.
20–17 DCDC_VDD1P45CTRL_ADJTN	Adjust value of duty cycle when switching between VDD1P45 and VDD1P8. The unit is 1/32 or 3.125%.
16 Reserved	This field is reserved. Reserved
15–11 DCDC_VDD1P45CTRL_TRG_BOOST	Target value of VDD1P45 in bost mode, 25 mV each step from 0x00 to 0x0F. In boost mode, DCDC boosts VDD1P45 to 1.8 V by default, software need to measure battery voltage in light load, then adjust the target value accordingly. If the total load current < 10mA, it is considered to be light load. It depends on the internal resistance of the battery type. 0x15 1.8 V 0x0F 1.65 V 0x07 1.45 V 0x00 1.275 V
10–6 DCDC_VDD1P45CTRL_TRG_BUCK	Target value of VDD1p45 in buck mode, 25 mV each step from 0x00 to 0x0F 0x0F 1.65 V 0x07 1.45 V 0x00 1.275 V
DCDC_VDD1P8CTRL_TRG	Target value of VDD1P8, 25 mV each step in two ranges, from 0x00 to 0x11 and 0x20 to 0x3F. 0x00 1.65 V 0x06 1.8 V 0x11 2.075 V 0x20 2.8 V 0x34 3.3 V 0x3F 3.575 V

14.2.6 DCDC REGISTER 4 (DCDC_REG4)

Address: 4005_A000h base + 10h offset = 4005_A010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UNLOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DCDC_SW_ SHUTDOWN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCDC_REG4 field descriptions

Field	Description
31–16 UNLOCK	0x3E77 KEY—Key needed to unlock HW_POWER_RESET register. Write 0x3E77 to unlock this register and allow other bits to be changed. **NOTE:** This register must be unlocked on a write-by-write basis, so the UNLOCK bit can contain the correct key value during all writes to this register in order to update any other bit values in the register.
15–1 Reserved	This field is reserved. Reserved
0 DCDC_SW_ SHUTDOWN	Shut down DCDC in buck mode. DCDC can be turned on by pulling PSWITCH to high momentarily (min 50 ms).

14.2.7 DCDC REGISTER 6 (DCDC_REG6)

Address: 4005_A000h base + 18h offset = 4005_A018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PSWITCH_INT_STS	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												PSWITCH_INT_MUTE	PSWITCH_INT_CLEAR	PSWITCH_INT_FALL_EN	PSWITCH_INT_RISE_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCDC_REG6 field descriptions

Field	Description
31 PSWITCH_INT_STS	PSWITCH edge detection interrupt status
30–4 Reserved	This field is reserved. Reserved
3 PSWITCH_INT_MUTE	Mask interrupt to SoC, edge detection result can be read from PSIWTCH_INT_STS.
2 PSWITCH_INT_CLEAR	Write 1 to clear interrupt. Set to 0 after clear.
1 PSWITCH_INT_FALL_EN	Enable falling edge detect for interrupt.
0 PSWITCH_INT_RISE_EN	Enable rising edge detect for interrupt.

14.2.8 DCDC REGISTER 7 (DCDC_REG7)

Address: 4005_A000h base + 1Ch offset = 4005_A01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											PULSE_RUN_SPEEDUP	INTEGRATOR_VALUE_SEL	INTEGRATOR_VALUE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTEGRATOR_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCDC_REG7 field descriptions

Field	Description
31–21 Reserved	This field is reserved. Reserved

Table continues on the next page...

DCDC_REG7 field descriptions (continued)

Field	Description
20 PULSE_RUN_ SPEEDUP	Enable pulse run speedup. Before setting this bit, INTEGRATOR_VALUE_SEL must be set to 1'b1 and integrator value must be programmed.
19 INTEGRATOR_ VALUE_SEL	Select the integrator value from above register or saved value in hardware. 1'b0 Select the saved value in hardware. 1'b1 Select the integrator value in this register.
INTEGRATOR_ VALUE	Integrator value which can be loaded in pulsed mode. Software can program this value according to battery voltage and VDD1P45 output target value before goes to the pulsed mode. It is signed number. The register value = (Dutycycle * 32 - 16) * 2 ^ 13 For buck mode, dutycycle = VDD1P45 / Vbat. For boost mode, dutycycle = (VDD1P45 - Vbat) / VDD1P45.

Chapter 15
Low-Leakage Wakeup Unit (LLWU)

15.1 Introduction

The LLWU module allows the user to select up to 16 external pins and up to internal modules as interrupt wake-up sources from low-leakage power modes.

The input sources are described in the device's chip configuration details. Each of the available wake-up sources can be individually enabled.

The $\overline{\text{RESET}}$ pin is an additional source for triggering an exit from low-leakage power modes, and causes the MCU to exit both LLS and VLLS through a reset flow.

The LLWU module also includes two optional digital pin filters for the external wakeup pins.

See AN4503: Power Management for Kinetis MCUs for further details on using the LLWU.

15.1.1 Features

The LLWU module features include:

- Support for up to 16 external input pins and up to internal modules with individual enable bits for MCU interrupt from low leakage modes
- Input sources may be external pins or from internal peripherals capable of running in LLS or VLLS. See the chip configuration information for wakeup input sources for this device.
- External pin wake-up inputs, each of which is programmable as falling-edge, rising-edge, or any change

- Wake-up inputs that are activated after MCU enters a low-leakage power mode
- Optional digital filters provided to qualify an external pin detect. Note that when the LPO clock is disabled, the filters are disabled and bypassed.

15.1.2 Modes of operation

The LLWU module becomes functional on entry into a low-leakage power mode. After recovery from LLS, the LLWU is immediately disabled. After recovery from VLLS, the LLWU continues to detect wake-up events until the user has acknowledged the wake-up via a write to PMC_REGSC[ACKISO].

15.1.2.1 LLS mode

Wake-up events due to external pin inputs (LLWU_Px) and internal module interrupt inputs (LLWU_MxIF) result in an interrupt flow when exiting LLS.

NOTE

The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit Stop mode on an LLS recovery.

15.1.2.2 VLLS modes

All wakeup and reset events result in VLLS exit via a reset flow.

15.1.2.3 Non-low leakage modes

The LLWU is not active in all non-low leakage modes where detection and control logic are in a static state. The LLWU registers are accessible in non-low leakage modes and are available for configuring and reading status when bus transactions are possible.

When the wake-up pin filters are enabled, filter operation begins immediately. If a low leakage mode is entered within five LPO clock cycles of an active edge, the edge event will be detected by the LLWU.

15.1.2.4 Debug mode

When the chip is in Debug mode and then enters LLS or a VLLSx mode, no debug logic works in the fully-functional low-leakage mode. Upon an exit from the LLS or VLLSx mode, the LLWU becomes inactive.

15.1.3 Block diagram

The following figure is the block diagram for the LLWU module.

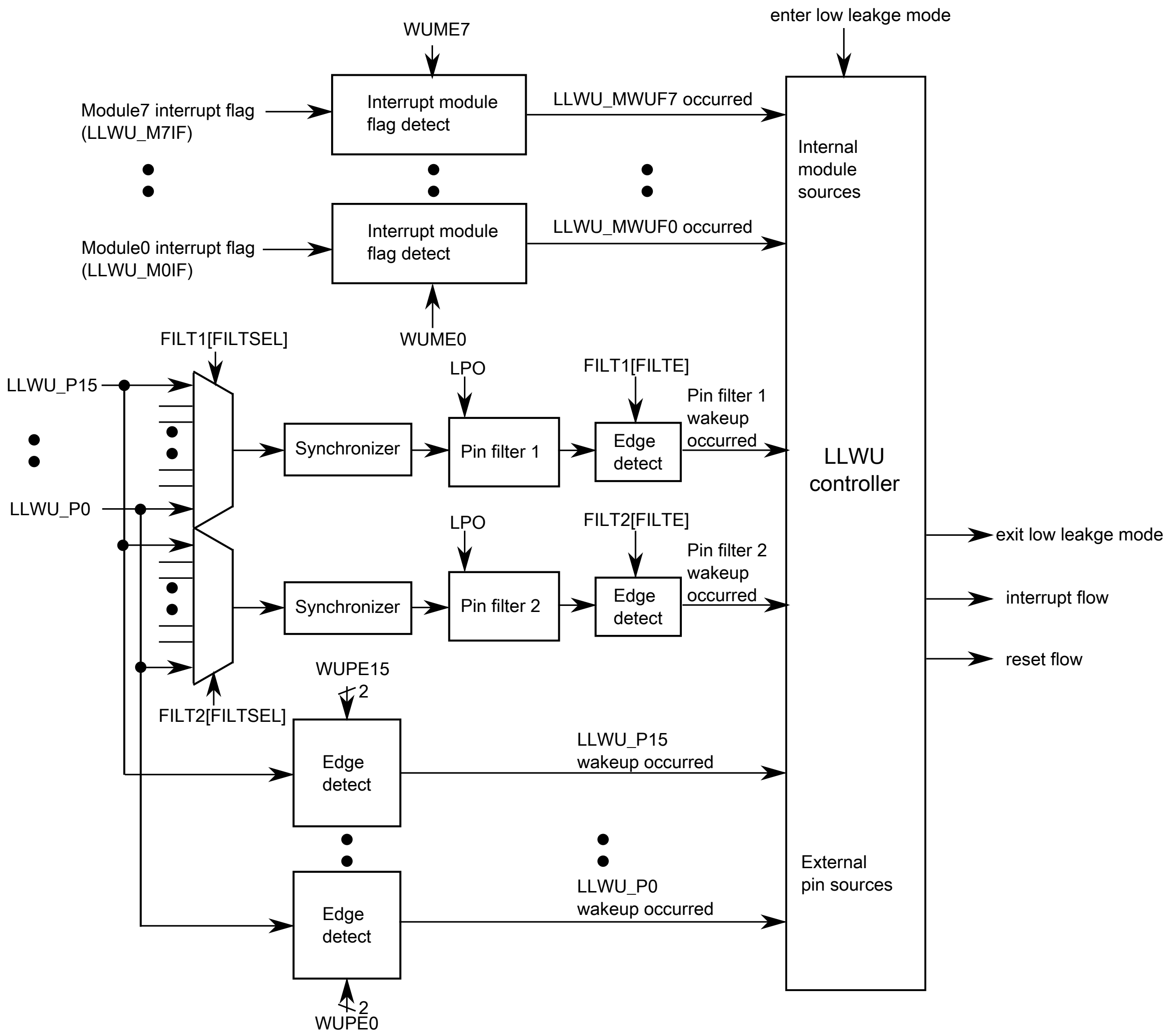


Figure 15-1. LLWU block diagram

15.2 LLWU signal descriptions

The signal properties of LLWU are shown in the table found here.

The external wakeup input pins can be enabled to detect either rising-edge, falling-edge, or on any change.

Table 15-1. LLWU signal descriptions

Signal	Description	I/O
LLWU_Pn	Wakeup inputs (n = 0-15)	I

15.3 Memory map/register definition

The LLWU includes the following registers:

- Wake-up source enable registers
    - Enable external pin input sources
    - Enable internal peripheral interrupt sources
- Wake-up flag registers
    - Indication of wakeup source that caused exit from a low-leakage power mode includes external pin or internal module interrupt
- Wake-up pin filter enable registers

NOTE

The LLWU registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

All LLWU registers are reset by Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. Each register's displayed reset value represents this subset of reset types. LLWU registers are unaffected by reset types that do not trigger Chip Reset not VLLS. For more information about the types of reset on this chip, refer to the Introduction details.

LLWU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_C000	LLWU Pin Enable 1 register (LLWU_PE1)	8	R/W	00h	15.3.1/277
4007_C001	LLWU Pin Enable 2 register (LLWU_PE2)	8	R/W	00h	15.3.2/278
4007_C002	LLWU Pin Enable 3 register (LLWU_PE3)	8	R/W	00h	15.3.3/279
4007_C003	LLWU Pin Enable 4 register (LLWU_PE4)	8	R/W	00h	15.3.4/280
4007_C004	LLWU Module Enable register (LLWU_ME)	8	R/W	00h	15.3.5/281
4007_C005	LLWU Flag 1 register (LLWU_F1)	8	R/W	00h	15.3.6/283
4007_C006	LLWU Flag 2 register (LLWU_F2)	8	R/W	00h	15.3.7/285
4007_C007	LLWU Flag 3 register (LLWU_F3)	8	R	00h	15.3.8/286
4007_C008	LLWU Pin Filter 1 register (LLWU_FILT1)	8	R/W	00h	15.3.9/288
4007_C009	LLWU Pin Filter 2 register (LLWU_FILT2)	8	R/W	00h	15.3.10/289

15.3.1 LLWU Pin Enable 1 register (LLWU_PE1)

LLWU_PE1 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P3–LLWU_P0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 0h offset = 4007_C000h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE3		WUPE2		WUPE1		WUPE0	
Reset	0	0	0	0	0	0	0	0

LLWU_PE1 field descriptions

Field	Description
7–6 WUPE3	Wakeup Pin Enable For LLWU_P3 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE2	Wakeup Pin Enable For LLWU_P2 Enables and configures the edge detection for the wakeup pin.

Table continues on the next page...

LLWU_PE1 field descriptions (continued)

Field	Description
	00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE1	Wakeup Pin Enable For LLWU_P1 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE0	Wakeup Pin Enable For LLWU_P0 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

15.3.2 LLWU Pin Enable 2 register (LLWU_PE2)

LLWU_PE2 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P7–LLWU_P4.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 1h offset = 4007_C001h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE7		WUPE6		WUPE5		WUPE4	
Reset	0	0	0	0	0	0	0	0

LLWU_PE2 field descriptions

Field	Description
7–6 WUPE7	Wakeup Pin Enable For LLWU_P7 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input

Table continues on the next page...

LLWU_PE2 field descriptions (continued)

Field	Description
	01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE6	Wakeup Pin Enable For LLWU_P6 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE5	Wakeup Pin Enable For LLWU_P5 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE4	Wakeup Pin Enable For LLWU_P4 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

15.3.3 LLWU Pin Enable 3 register (LLWU_PE3)

LLWU_PE3 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P11–LLWU_P8.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 2h offset = 4007_C002h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE11		WUPE10		WUPE9		WUPE8	
Reset	0	0	0	0	0	0	0	0

LLWU_PE3 field descriptions

Field	Description
7–6 WUPE11	Wakeup Pin Enable For LLWU_P11 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE10	Wakeup Pin Enable For LLWU_P10 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE9	Wakeup Pin Enable For LLWU_P9 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE8	Wakeup Pin Enable For LLWU_P8 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

15.3.4 LLWU Pin Enable 4 register (LLWU_PE4)

LLWU_PE4 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P15–LLWU_P12.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 3h offset = 4007_C003h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE15		WUPE14		WUPE13		WUPE12	
Reset	0	0	0	0	0	0	0	0

LLWU_PE4 field descriptions

Field	Description
7–6 WUPE15	Wakeup Pin Enable For LLWU_P15 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE14	Wakeup Pin Enable For LLWU_P14 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE13	Wakeup Pin Enable For LLWU_P13 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE12	Wakeup Pin Enable For LLWU_P12 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

15.3.5 LLWU Module Enable register (LLWU_ME)

LLWU_ME contains the bits to enable the internal module flag as a wakeup input source for inputs MWUF7–MWUF0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 4h offset = 4007_C004h

Bit	7	6	5	4	3	2	1	0
Read Write	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
Reset	0	0	0	0	0	0	0	0

LLWU_ME field descriptions

Field	Description
7 WUME7	Wakeup Module Enable For Module 7 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
6 WUME6	Wakeup Module Enable For Module 6 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
5 WUME5	Wakeup Module Enable For Module 5 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
4 WUME4	Wakeup Module Enable For Module 4 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
3 WUME3	Wakeup Module Enable For Module 3 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
2 WUME2	Wakeup Module Enable For Module 2 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
1 WUME1	Wakeup Module Enable for Module 1 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

Table continues on the next page...

LLWU_ME field descriptions (continued)

Field	Description
0 WUME0	Wakeup Module Enable For Module 0 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

15.3.6 LLWU Flag 1 register (LLWU_F1)

LLWU_F1 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 5h offset = 4007_C005h

Bit	7	6	5	4	3	2	1	0
Read	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F1 field descriptions

Field	Description
7 WUF7	Wakeup Flag For LLWU_P7 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF7. 0 LLWU_P7 input was not a wakeup source 1 LLWU_P7 input was a wakeup source
6 WUF6	Wakeup Flag For LLWU_P6 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF6.

Table continues on the next page...

LLWU_F1 field descriptions (continued)

Field	Description
	0 LLWU_P6 input was not a wakeup source 1 LLWU_P6 input was a wakeup source
5 WUF5	Wakeup Flag For LLWU_P5 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF5. 0 LLWU_P5 input was not a wakeup source 1 LLWU_P5 input was a wakeup source
4 WUF4	Wakeup Flag For LLWU_P4 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF4. 0 LLWU_P4 input was not a wakeup source 1 LLWU_P4 input was a wakeup source
3 WUF3	Wakeup Flag For LLWU_P3 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF3. 0 LLWU_P3 input was not a wake-up source 1 LLWU_P3 input was a wake-up source
2 WUF2	Wakeup Flag For LLWU_P2 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF2. 0 LLWU_P2 input was not a wakeup source 1 LLWU_P2 input was a wakeup source
1 WUF1	Wakeup Flag For LLWU_P1 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF1. 0 LLWU_P1 input was not a wakeup source 1 LLWU_P1 input was a wakeup source
0 WUF0	Wakeup Flag For LLWU_P0 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF0. 0 LLWU_P0 input was not a wakeup source 1 LLWU_P0 input was a wakeup source

15.3.7 LLWU Flag 2 register (LLWU_F2)

LLWU_F2 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 6h offset = 4007_C006h

Bit	7	6	5	4	3	2	1	0
Read	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF9	WUF8
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F2 field descriptions

Field	Description
7 WUF15	Wakeup Flag For LLWU_P15 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF15. 0 LLWU_P15 input was not a wakeup source 1 LLWU_P15 input was a wakeup source
6 WUF14	Wakeup Flag For LLWU_P14 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF14. 0 LLWU_P14 input was not a wakeup source 1 LLWU_P14 input was a wakeup source
5 WUF13	Wakeup Flag For LLWU_P13 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF13. 0 LLWU_P13 input was not a wakeup source 1 LLWU_P13 input was a wakeup source

Table continues on the next page...

LLWU_F2 field descriptions (continued)

Field	Description
4 WUF12	Wakeup Flag For LLWU_P12 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF12. 0 LLWU_P12 input was not a wakeup source 1 LLWU_P12 input was a wakeup source
3 WUF11	Wakeup Flag For LLWU_P11 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF11. 0 LLWU_P11 input was not a wakeup source 1 LLWU_P11 input was a wakeup source
2 WUF10	Wakeup Flag For LLWU_P10 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF10. 0 LLWU_P10 input was not a wakeup source 1 LLWU_P10 input was a wakeup source
1 WUF9	Wakeup Flag For LLWU_P9 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF9. 0 LLWU_P9 input was not a wakeup source 1 LLWU_P9 input was a wakeup source
0 WUF8	Wakeup Flag For LLWU_P8 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF8. 0 LLWU_P8 input was not a wakeup source 1 LLWU_P8 input was a wakeup source

15.3.8 LLWU Flag 3 register (LLWU_F3)

LLWU_F3 contains the wakeup flags indicating which internal wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

For internal peripherals that are capable of running in a low-leakage power mode, such as a real time clock module or CMP module, the flag from the associated peripheral is accessible as the MWUFx bit. The flag will need to be cleared in the peripheral instead of writing a 1 to the MWUFx bit.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 7h offset = 4007_C007h

Bit	7	6	5	4	3	2	1	0
Read	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
Write								
Reset	0	0	0	0	0	0	0	0

LLWU_F3 field descriptions

Field	Description
7 MWUF7	Wakeup flag For module 7 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 7 input was not a wakeup source 1 Module 7 input was a wakeup source
6 MWUF6	Wakeup flag For module 6 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 6 input was not a wakeup source 1 Module 6 input was a wakeup source
5 MWUF5	Wakeup flag For module 5 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 5 input was not a wakeup source 1 Module 5 input was a wakeup source
4 MWUF4	Wakeup flag For module 4 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 4 input was not a wakeup source 1 Module 4 input was a wakeup source
3 MWUF3	Wakeup flag For module 3 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 3 input was not a wakeup source 1 Module 3 input was a wakeup source
2 MWUF2	Wakeup flag For module 2

Table continues on the next page...

LLWU_F3 field descriptions (continued)

Field	Description
	Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 2 input was not a wakeup source 1 Module 2 input was a wakeup source
1 MWUF1	Wakeup flag For module 1 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 1 input was not a wakeup source 1 Module 1 input was a wakeup source
0 MWUF0	Wakeup flag For module 0 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 0 input was not a wakeup source 1 Module 0 input was a wakeup source

15.3.9 LLWU Pin Filter 1 register (LLWU_FILT1)

LLWU_FILT1 is a control and status register that is used to enable/disable the digital filter 1 features for an external pin.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT1 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF.

Table continues on the next page...

LLWU_FILT1 field descriptions (continued)

Field	Description
	0 Pin Filter 1 was not a wakeup source 1 Pin Filter 1 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

15.3.10 LLWU Pin Filter 2 register (LLWU_FILT2)

LLWU_FILT2 is a control and status register that is used to enable/disable the digital filter 2 features for an external pin.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 9h offset = 4007_C009h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT2 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF.

Table continues on the next page...

LLWU_FILT2 field descriptions (continued)

Field	Description
	0 Pin Filter 2 was not a wakeup source 1 Pin Filter 2 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

15.4 Functional description

Thie low-leakage wakeup unit (LLWU) module allows internal peripherals and external input pins as a source of wakeup from low-leakage modes.

It is operational only in LLS and VLLSx modes.

The LLWU module contains pin enables for each external pin and internal module. For each external pin, the user can disable or select the edge type for the wakeup with the following options:

- Falling-edge
- Rising-edge
- Either-edge

When an external pin is enabled as a wakeup source, the pin must be configured as an input pin.

The LLWU implements optional 3-cycle glitch filters, based on the LPO clock. A detected external pin is required to remain asserted until the enabled glitch filter times out. Additional latency of up to 2 cycles is due to synchronization, which results in a total of up to 5 cycles of delay before the detect circuit alerts the system to the wakeup or reset event when the filter function is enabled. Two wakeup detect filters are available for selected external pins. Glitch filtering is not provided on the internal modules.

For internal module interrupts, the WUMEx bit enables the associated module interrupt as a wakeup source.

15.4.1 LLS mode

Wakeup events triggered from either an external pin input or an internal module interrupt, result in a CPU interrupt flow to begin user code execution.

15.4.2 VLLS modes

For any wakeup from VLLS, recovery is always via a reset flow and RCM_SRS[WAKEUP] is set indicating the low-leakage mode was active. State retention data is lost and I/O will be restored after PMC_REGSC[ACKISO] has been written.

A VLLS exit event due to $\overline{\text{RESET}}$ pin assertion causes an exit via a system reset. State retention data is lost and the I/O states immediately return to their reset state. The RCM_SRS[WAKEUP] and RCM_SRS[PIN] bits are set and the system executes a reset flow before CPU operation begins with a reset vector fetch.

15.4.3 Initialization

For an enabled peripheral wakeup input, the peripheral flag must be cleared by software before entering LLS or VLLSx mode to avoid an immediate exit from the mode.

Flags associated with external input pins, filtered and unfiltered, must also be cleared by software prior to entry to LLS or VLLSx mode.

After enabling an external pin filter or changing the source pin, wait at least five LPO clock cycles before entering LLS or VLLSx mode to allow the filter to initialize.

NOTE

> After recovering from a VLLS mode, user must restore chip configuration before clearing PMC_REGSC[ACKISO]. In particular, pin configuration for enabled LLWU wake-up pins must be restored to avoid any LLWU flag from being falsely set when PMC_REGSC[ACKISO] is cleared.
>
> The signal selected as a wake-up source pin must be a digital pin, as selected in the pin mux control.

Chapter 16
Reset Control Module (RCM)

16.1 Introduction

Information found here describes the registers of the Reset Control Module (RCM). The RCM implements many of the reset functions for the chip. See the chip's reset chapter for more information.

See AN4503: Power Management for Kinetis MCUs for further details on using the RCM.

16.2 Reset memory map and register descriptions

The RCM Memory Map/Register Definition can be found here.

The Reset Control Module (RCM) registers provide reset status information and reset filter control.

NOTE

> The RCM registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

RCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_F000	System Reset Status Register 0 (RCM_SRS0)	8	R	82h	16.2.1/294
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h	16.2.2/295
4007_F004	Reset Pin Filter Control register (RCM_RPFC)	8	R/W	00h	16.2.3/296
4007_F005	Reset Pin Filter Width register (RCM_RPFW)	8	R/W	00h	16.2.4/297

16.2.1 System Reset Status Register 0 (RCM_SRS0)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x82
- LVD (without POR) — 0x02
- VLLS mode wakeup due to $\overline{\text{RESET}}$ pin assertion — 0x41
- VLLS mode wakeup due to other wakeup sources — 0x01
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0		LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

RCM_SRS0 field descriptions

Field	Description
7 POR	Power-On Reset Indicates a reset has been caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR 1 Reset caused by POR
6 PIN	External Reset Pin Indicates a reset has been caused by an active-low level on the external $\overline{\text{RESET}}$ pin. 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Indicates a reset has been caused by the watchdog timer timing out. This reset source can be blocked by disabling the watchdog. 0 Reset not caused by watchdog timeout 1 Reset caused by watchdog timeout
4–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

RCM_SRS0 field descriptions (continued)

Field	Description
2 LOC	Loss-of-Clock Reset Indicates a reset has been caused by a loss of external clock. The MCG clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed MCG description for information on enabling the clock monitor. 0 Reset not caused by a loss of external clock. 1 Reset caused by a loss of external clock.
1 LVD	Low-Voltage Detect Reset If PMC_LVDSC1[LVDRE] is set and the supply drops below the LVD trip voltage, an LVD reset occurs. This field is also set by POR. 0 Reset not caused by LVD trip or POR 1 Reset caused by LVD trip or POR
0 WAKEUP	Low Leakage Wakeup Reset Indicates a reset has been caused by an enabled LLWU module wakeup source while the chip was in a low leakage mode. In LLS mode, the $\overline{\text{RESET}}$ pin is the only wakeup source that can cause this reset. Any enabled wakeup source in a VLLSx mode causes a reset. This bit is cleared by any reset except WAKEUP. 0 Reset not caused by LLWU module wakeup source 1 Reset caused by LLWU module wakeup source

16.2.2 System Reset Status Register 1 (RCM_SRS1)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x00
- LVD (without POR) — 0x00
- VLLS mode wakeup — 0x00
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 1h offset = 4007_F001h

Bit	7	6	5	4	3	2	1	0
Read	0	0	SACKERR	0	MDM_AP	SW	LOCKUP	0
Write								
Reset	0	0	0	0	0	0	0	0

RCM_SRS1 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SACKERR	Stop Mode Acknowledge Error Reset Indicates that after an attempt to enter Stop mode, a reset has been caused by a failure of one or more peripherals to acknowledge within approximately one second to enter stop mode. 0 Reset not caused by peripheral failure to acknowledge attempt to enter stop mode 1 Reset caused by peripheral failure to acknowledge attempt to enter stop mode
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 MDM_AP	MDM-AP System Reset Request Indicates a reset has been caused by the host debugger system setting of the System Reset Request bit in the MDM-AP Control Register. 0 Reset not caused by host debugger system setting of the System Reset Request bit 1 Reset caused by host debugger system setting of the System Reset Request bit
2 SW	Software Indicates a reset has been caused by software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register in the ARM core. 0 Reset not caused by software setting of SYSRESETREQ bit 1 Reset caused by software setting of SYSRESETREQ bit
1 LOCKUP	Core Lockup Indicates a reset has been caused by the ARM core indication of a LOCKUP event. 0 Reset not caused by core LOCKUP event 1 Reset caused by core LOCKUP event
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.2.3 Reset Pin Filter Control register (RCM_RPFC)

NOTE

The reset values of bits 2-0 are for Chip POR only. They are unaffected by other reset types.

NOTE

The bus clock filter is reset when disabled or when entering stop mode. The LPO filter is reset when disabled .

Address: 4007_F000h base + 4h offset = 4007_F004h

Bit	7	6	5	4	3	2	1	0
Read	0					RSTFLTSS	RSTFLTSRW	
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFC field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RSTFLTSS	Reset Pin Filter Select in Stop Mode Selects how the reset pin filter is enabled in Stop and VLPS modes , and also during LLS and VLLS modes. On exit from VLLS mode, this bit should be reconfigured before clearing PMC_REGSC[ACKISO]. 0 All filtering disabled 1 LPO clock filter enabled
RSTFLTSRW	Reset Pin Filter Select in Run and Wait Modes Selects how the reset pin filter is enabled in run and wait modes. 00 All filtering disabled 01 Bus clock filter enabled for normal operation 10 LPO clock filter enabled for normal operation 11 Reserved

16.2.4 Reset Pin Filter Width register (RCM_RPFW)

NOTE

The reset values of the bits in the RSTFLTSEL field are for Chip POR only. They are unaffected by other reset types.

Address: 4007_F000h base + 5h offset = 4007_F005h

Bit	7	6	5	4	3	2	1	0
Read	0			RSTFLTSEL				
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFW field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RSTFLTSEL	Reset Pin Filter Bus Clock Select Selects the reset pin bus clock filter width.

Table continues on the next page...

RCM_RPFW field descriptions (continued)

Field	Description
	00000 Bus clock filter count is 1
	00001 Bus clock filter count is 2
	00010 Bus clock filter count is 3
	00011 Bus clock filter count is 4
	00100 Bus clock filter count is 5
	00101 Bus clock filter count is 6
	00110 Bus clock filter count is 7
	00111 Bus clock filter count is 8
	01000 Bus clock filter count is 9
	01001 Bus clock filter count is 10
	01010 Bus clock filter count is 11
	01011 Bus clock filter count is 12
	01100 Bus clock filter count is 13
	01101 Bus clock filter count is 14
	01110 Bus clock filter count is 15
	01111 Bus clock filter count is 16
	10000 Bus clock filter count is 17
	10001 Bus clock filter count is 18
	10010 Bus clock filter count is 19
	10011 Bus clock filter count is 20
	10100 Bus clock filter count is 21
	10101 Bus clock filter count is 22
	10110 Bus clock filter count is 23
	10111 Bus clock filter count is 24
	11000 Bus clock filter count is 25
	11001 Bus clock filter count is 26
	11010 Bus clock filter count is 27
	11011 Bus clock filter count is 28
	11100 Bus clock filter count is 29
	11101 Bus clock filter count is 30
	11110 Bus clock filter count is 31
	11111 Bus clock filter count is 32

Chapter 17
Bit Manipulation Engine (BME)

The Bit Manipulation Engine (BME) provides hardware support for atomic read-modify-write memory operations to the peripheral address space in ARM®Cortex™-M0+ based microcontrollers. This architectural capability is also known as "decorated storage." By combining the basic load and store instructions of the Cortex-M instruction set architecture (v6M, v7M) with the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers.

17.1 Introduction

The Bit Manipulation Engine (BME) provides hardware support for atomic read-modify-write memory operations to the peripheral address space in Cortex-M0+ based microcontrollers.

This architectural capability is also known as "decorated storage" as it defines a mechanism for providing additional semantics for load and store operations to memory-mapped peripherals beyond just the reading and writing of data values to the addressed memory locations. In the BME definition, the "decoration", that is, the additional semantic information, is encoded into the peripheral address used to reference the memory.

By combining the basic load and store instructions of the ARM Cortex-M instruction set architecture (v6M, v7M) with the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers. The resulting architectural capability defined by this core platform function is targeted at the manipulation of n-bit fields in peripheral registers and is consistent with I/O hardware addressing in the Embedded C standard. For most BME commands, a single core read or write bus cycle is converted into an atomic read-modify-write, that is, an indivisible "read followed by a write" bus sequence.

BME decorated references are only available on system bus transactions generated by the processor core and targeted at the standard 512 KB peripheral address space based at 0x4000_0000[1]. The decoration semantic is embedded into address bits[28:19], creating a 448 MB space at addresses 0x4400_0000–0x5FFF_FFFF for AIPS; these bits are stripped out of the actual address sent to the peripheral bus controller and used by the BME to define and control its operation.

17.1.1 Overview

The following figure is a generic block diagram of the processor core and platform for this class of ultra low-end microcontrollers.

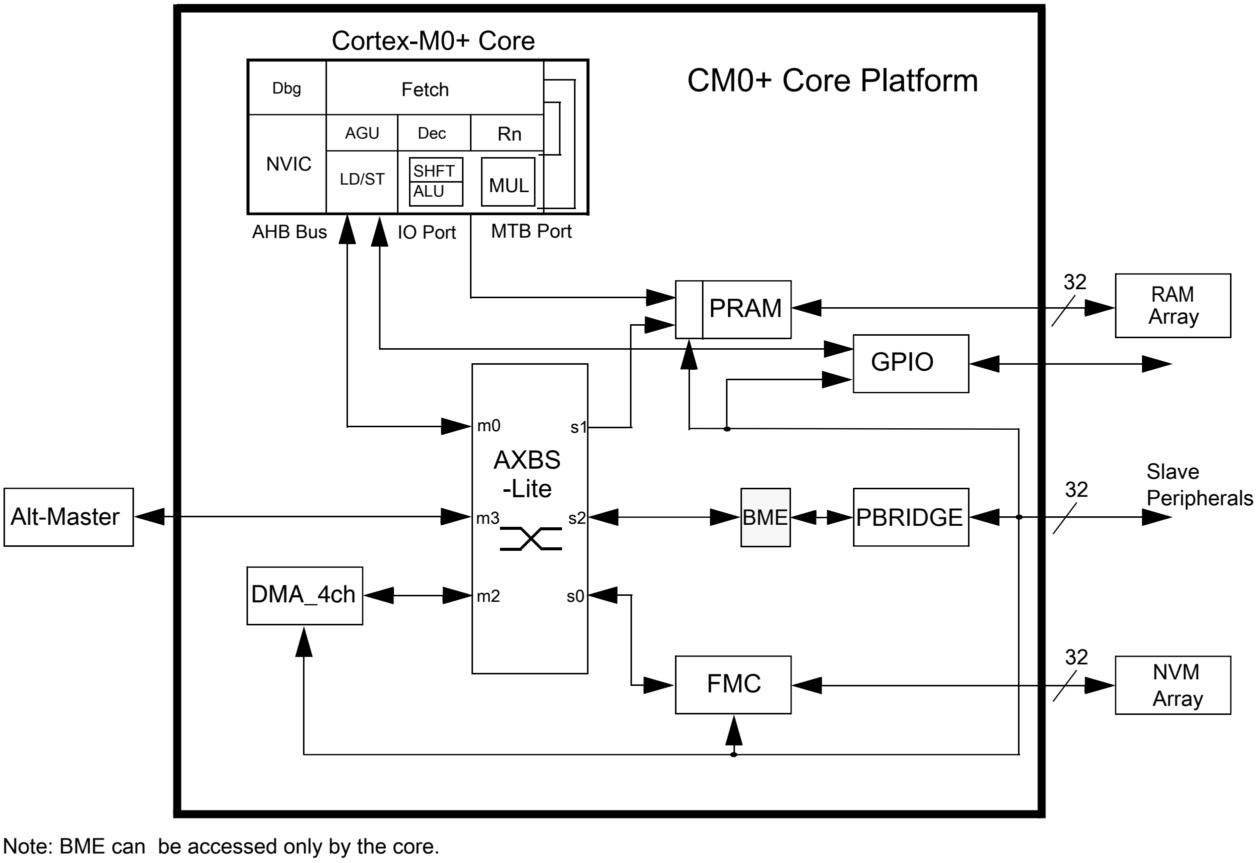


Figure 17-1. Cortex-M0+ core platform block diagram

[1] To be perfectly accurate, the peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses. This organization provides compatibility with the Kinetis K Family. Attempted accesses to the memory space located between 0x4008_0000 - 0x400F_EFFF are error terminated due to an illegal address.

As shown in the block diagram, the BME module interfaces to a crossbar switch AHB slave port as its primary input and sources an AHB bus output to the Peripheral Bridge (PBRIDGE) controller. The BME hardware microarchitecture is a 2-stage pipeline design matching the protocol of the AMBA-AHB system bus interfaces. The PBRIDGE module converts the AHB system bus protocol into the IPS/APB protocol used by the attached slave peripherals.

17.1.2 Features

The key features of the BME include:

- Lightweight implementation of decorated storage for selected address spaces
- Additional access semantics encoded into the reference address
- Resides between a crossbar switch slave port and a peripheral bridge bus controller
- Two-stage pipeline design matching the AHB system bus protocol
- Combinationally passes non-decorated accesses to peripheral bridge bus controller
- Conversion of decorated loads and stores from processor core into atomic read-modify-writes
- Decorated loads support unsigned bit field extracts, load-and-{set,clear} 1-bit operations
- Decorated stores support bit field inserts, logical AND, OR, and XOR operations
- Support for byte, halfword and word-sized decorated operations
- Supports minimum signal toggling on AHB output bus to reduce power dissipation

17.1.3 Modes of operation

The BME module does not support any special modes of operation. As a memory-mapped device located on a crossbar slave AHB system bus port, BME responds strictly on the basis of memory addresses for accesses to the peripheral bridge bus controller.

All functionality associated with the BME module resides in the core platform's clock domain; this includes its connections with the crossbar slave port and the PBRIDGE bus controller.

17.2 Memory map and register definition

The BME module provides a memory-mapped capability and does not include any programming model registers.

The exact set of functions supported by the BME are detailed in the Functional description.

The peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses; the decorated address space is mapped to the 448 MB region located at 0x4400_0000–0x5FFF_FFFF.

17.3 Functional description

Information found here details the specific functions supported by the BME.

Recall the combination of the basic load and store instructions of the Cortex-M instruction set architecture (v6M, v7M) plus the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers. The resulting architectural capability defined by this core platform function is targeted at the manipulation of n-bit fields in peripheral registers and is consistent with I/O hardware addressing in the Embedded C standard. For most BME commands, a single core read or write bus cycle is converted into an atomic read-modify-write, that is, an indivisible "read followed by a write" bus sequence.

Consider decorated store operations first, then decorated loads.

17.3.1 BME decorated stores

The functions supported by the BME's decorated stores include three logical operators (AND, OR, XOR) plus a bit field insert.

For all these operations, BME converts a single decorated AHB store transaction into a 2-cycle atomic read-modify-write sequence, where the combined read-modify operation is performed in the first AHB data phase, and then the write is performed in the second AHB data phase.

A generic timing diagram of a decorated store showing a peripheral bit field insert operation is shown as follows:

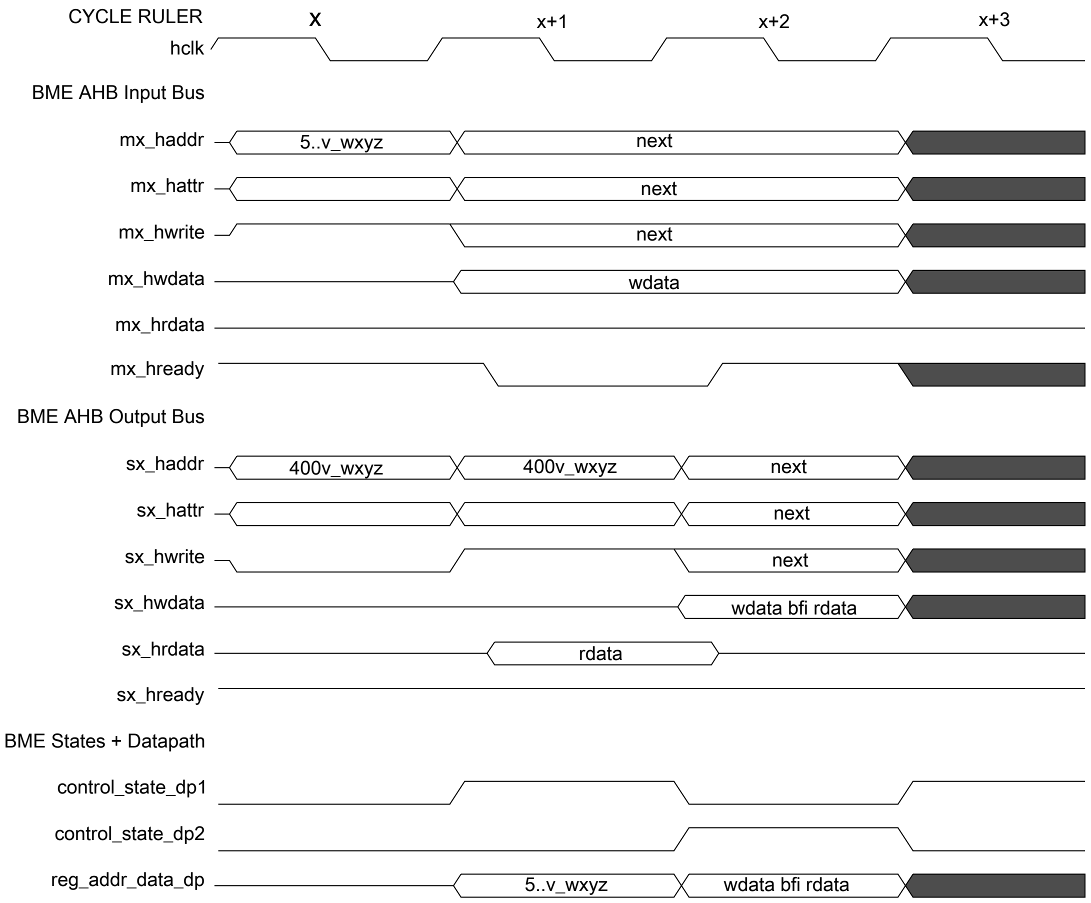


Figure 17-2. Decorated store: bit field insert timing diagram

All the decorated store operations follow the same execution template shown in Figure 17-2, a two-cycle read-modify-write operation:

1. Cycle x, 1st AHB address phase: Write from input bus is translated into a read operation on the output bus using the actual memory address (with the decoration removed) and then captured in a register.
2. Cycle x+1, 2nd AHB address phase: Write access with the registered (but actual) memory address is output
3. Cycle x+1, 1st AHB data phase: Memory read data is modified using the input bus write data and the function defined by the decoration and captured in a data register; the input bus cycle is stalled.
4. Cycle x+2, 2nd AHB data phase: Registered write data is sourced onto the output write data bus.

NOTE

Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

17.3.1.1 Decorated store logical AND (AND)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read;
2. It is then modified by performing a logical AND operation using the write data operand sourced for the system bus cycle
3. Finally, the result of the AND operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ioandb	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr																			
ioandh	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr																			0
ioandw	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr																		0	0

Figure 17-3. Decorated store address: logical AND

See Figure 17-3, where addr[30:29] = 10 for peripheral, addr[28:26] = 001 specifies the AND operation, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated AND write operation is defined in the following pseudo-code as:

```
ioand<sz>(accessAddress, wdata)                 // decorated store AND
tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
tmp   =  tmp & wdata                            // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp     // memory write
```

where the operand size <sz> is defined as b(yte, 8-bit), h(alfword, 16-bit) and w(ord, 32-bit). This notation is used throughout the document.

In the cycle definition tables, the notations AHB_ap and AHB_dp refer to the address and data phases of the BME AHB transaction. The cycle-by-cycle BME operations are detailed in the following table.

Table 17-1. Cycle definitions of decorated store: logical AND

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert	Recirculate captured addr + attr to memory as slave_wt	<next>

Table continues on the next page...

Table 17-1. Cycle definitions of decorated store: logical AND (continued)

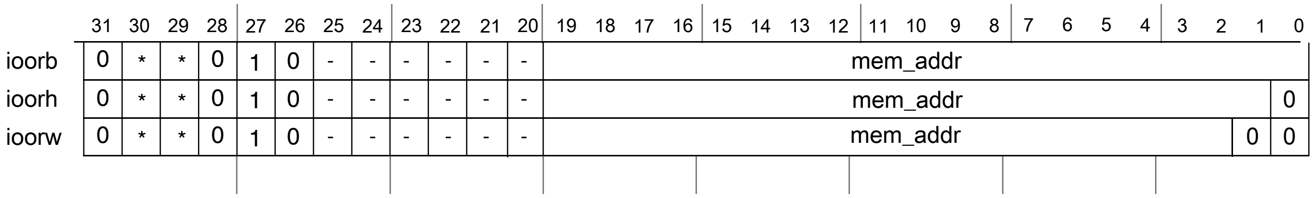
Pipeline stage	Cycle		
	x	x+1	x+2
	master_wt to slave_rd; Capture address, attributes		
BME AHB_dp	<previous>	Perform memory read; Form (rdata & wdata) and capture destination data in register	Perform write sending registered data to memory

17.3.1.2 Decorated store logical OR (OR)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read.
2. It is then modified by performing a logical OR operation using the write data operand sourced for the system bus cycle.
3. Finally, the result of the OR operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ioorb	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr																			
ioorh	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr																			0
ioorw	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr																		0	0

Figure 17-4. Decorated address store: logical OR

See Figure 17-4,where addr[30:29] =10 for peripheral, addr[28:26] = 010 specifies the OR operation, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated OR write operation is defined in the following pseudo-code as:

```
ioor<sz>(accessAddress, wdata)                    // decorated store OR

tmp   =  mem[accessAddress & 0xE00FFFFF, size]   // memory read
tmp   =  tmp | wdata                             // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp      // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 17-2. Cycle definitions of decorated store: logical OR

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form (rdata | wdata) and capture destination data in register	Perform write sending registered data to memory

17.3.1.3 Decorated store logical XOR (XOR)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read.
2. It is then modified by performing a logical XOR (exclusive-OR) operation using the write data operand sourced for the system bus cycle.
3. Finally, the result of the XOR operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ioxorb	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr																			
ioxorh	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr																			0
ioxorw	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr																		0	0

Figure 17-5. Decorated address store: logical XOR

See Figure 17-5, where addr[30:29] =10 for peripheral, addr[28:26] = 011 specifies the XOR operation, and mem_addr[19:0] specifies the address offset into the peripheral space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated XOR write operation is defined in the following pseudo-code as:

```
ioxor<sz>(accessAddress, wdata)                // decorated store XOR

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
tmp   =  tmp ^ wdata                           // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp    // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 17-3. Cycle definitions of decorated store: logical XOR

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form (rdata ^ wdata) and capture destination data in register	Perform write sending registered data to memory

17.3.1.4 Decorated store bit field insert (BFI)

This command inserts a bit field contained in the write data operand, defined by LSB position (b) and the bit field width (w+1), into the memory "container" defined by the access size associated with the store instruction using an atomic read-modify-write sequence.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

NOTE

> For the word sized operation, the maximum bit field width is 16 bits. The core performs the required write data lane replication on byte and halfword transfers.

The BFI operation can be used to insert a single bit into a peripheral. For this case, the w field is simply set to 0, indicating a bit field width of 1.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
iobfib	0	*	*	1	-	-	b	b	b	-	w	w	w	mem_addr																		
iobfih	0	*	*	1	-	b	b	b	b	w	w	w	w	mem_addr																		0
iobfiw	0	*	*	1	b	b	b	b	b	w	w	w	w	mem_addr																	0	0

Figure 17-6. Decorated address store: bit field insert

where addr[30:29] =10 for peripheral,addr[28] = 1 signals a BFI operation, addr[27:23] is "b", the LSB identifier, addr[22:19] is "w", the bit field width minus 1 identifier, and addr[18:0] specifies the address offset into the peripheral space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care". Note, unlike the other decorated store operations, BFI uses addr[19] as the least significant bit in the "w" specifier and not as an address bit.

The decorated BFI write operation is defined in the following pseudo-code as:

```
iobfi<sz>(accessAddress, wdata)                  // decorated bit field insert

tmp   =  mem[accessAddress & 0xE007FFFF, size]   // memory read
mask  = ((1 << (w+1)) - 1) << b                  // generate bit mask
tmp   =  tmp   & ~mask                           // modify
      |  wdata &  mask
mem[accessAddress & 0xE007FFFF, size] = tmp      // memory write
```

The write data operand (wdata) associated with the store instruction contains the bit field to be inserted. It must be properly aligned within a right-aligned container, that is, within the lower 8 bits for a byte operation, the lower 16 bits for a halfword, or the entire 32 bits for a word operation.

To illustrate, consider the following example of the insertion of the 3-bit field "xyz" into an 8-bit memory container, initially set to "abcd_efgh". For all cases, w is 2, signaling a bit field width of 3.

```
if b = 0 and the decorated store (strb) Rt register[7:0] = ----_-xyz,
   then destination is "abcd_exyz"
if b = 1 and the decorated store (strb) Rt register[7:0] = ----_xyz-,
   then destination is "abcd_xyzh"
if b = 2 and the decorated store (strb) Rt register[7:0] = ---x_yz--,
   then destination is "abcx_yzgh"
if b = 3 and the decorated store (strb) Rt register[7:0] = --xy_z---,
   then destination is "abxy_zfgh"
if b = 4 and the decorated store (strb) Rt register[7:0] = -xyz_----,
   then destination is "axyz_efgh"
if b = 5 and the decorated store (strb) Rt register[7:0] = xyz-_----,
   then destination is "xyzd_efgh"
if b = 6 and the decorated store (strb) Rt register[7:0] = yz--_----,
   then destination is "yzcd_efgh"
if b = 7 and the decorated store (strb) Rt register[7:0] = z---_----,
   then destination is "zbcd_efgh"
```

Note from the example, when the starting bit position plus the field width exceeds the container size, only part of the source bit field is inserted into the destination memory location. Stated differently, if (b + w+1) > container_width, only the low-order "container_width - b" bits are actually inserted.

The cycle-by-cycle BME operations are detailed in the following table.

Table 17-4. Cycle definitions of decorated store: bit field insert

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Form bitwise ((mask) ? wdata : rdata)) and capture destination data in register	Perform write sending registered data to memory

17.3.2 BME decorated loads

The functions supported by the BME's decorated loads include two single-bit load-and-{set, clear} operators plus unsigned bit field extracts.

For the two load-and-{set, clear} operations, BME converts a single decorated AHB load transaction into a two-cycle atomic read-modify-write sequence, where the combined read-modify operations are performed in the first AHB data phase, and then the write is performed in the second AHB data phase as the original read data is returned to the processor core. For an unsigned bit field extract, the decorated load transaction is stalled for one cycle in the BME as the data field is extracted, then aligned and returned to the processor in the second AHB data phase. This is the only decorated transaction that is not an atomic read-modify-write, as it is a simple data read.

A generic timing diagram of a decorated load showing a peripheral load-and-set 1-bit operation is shown as follows.

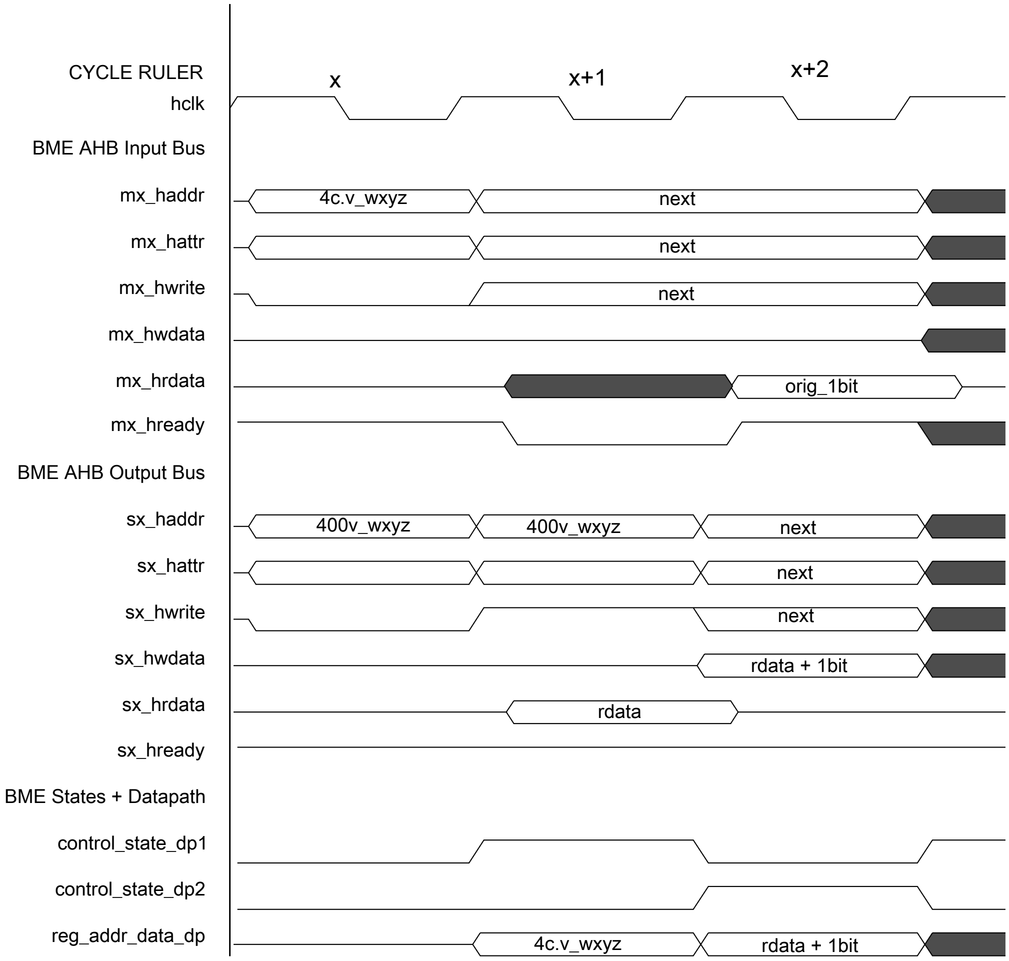


Figure 17-7. Decorated load: load-and-set 1-bit field insert timing diagram

Decorated load-and-{set, clear} 1-bit operations follow the execution template shown in the above figure: a 2-cycle read-modify-write operation:

1. Cycle x, first AHB address phase: Read from input bus is translated into a read operation on the output bus with the actual memory address (with the decoration removed) and then captured in a register
2. Cycle x+1, second AHB address phase: Write access with the registered (but actual) memory address is output
3. Cycle x+1, first AHB data phase: The "original" 1-bit memory read data is captured in a register, while the 1-bit field is set or clear based on the function defined by the decoration with the modified data captured in a register; the input bus cycle is stalled
4. Cycle x+2, second AHB data phase: The selected original 1-bit is right-justified, zero-filled and then driven onto the input read data bus, while the registered write data is sourced onto the output write data bus

NOTE

> Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

A generic timing diagram of a decorated load showing an unsigned peripheral bit field operation is shown in the following figure.

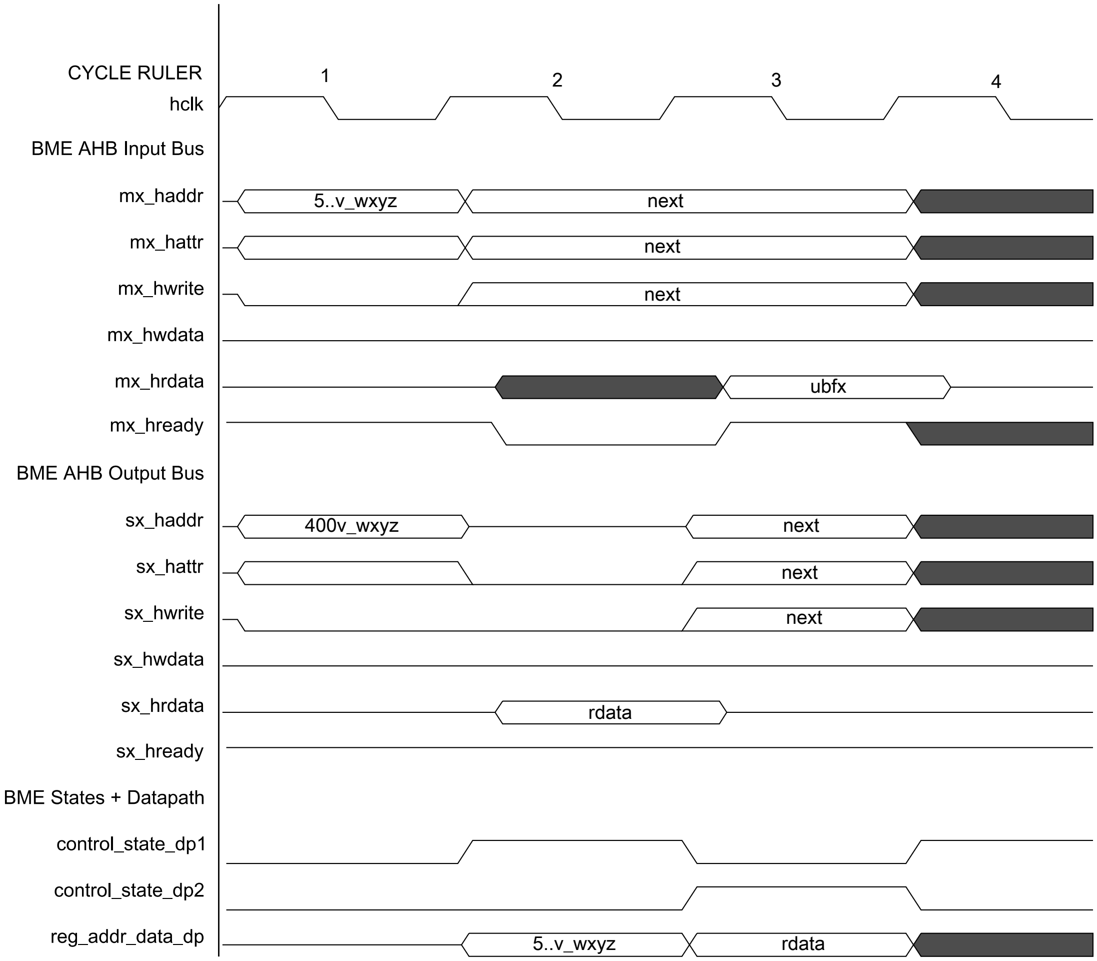


Figure 17-8. Decorated load: unsigned bit field insert timing diagram

The decorated unsigned bit field extract follows the same execution template shown in the above figure, a 2-cycle read operation:

- Cycle x, 1st AHB address phase: Read from input bus is translated into a read operation on the output bus with the actual memory address (with the decoration removed) and then captured in a register
- Cycle x+1, 2nd AHB address phase: Idle cycle

- Cycle x+1, 1st AHB data phase: A bit mask is generated based on the starting bit position and the field width; the mask is AND'ed with the memory read data to isolate the bit field; the resulting data is captured in a data register; the input bus cycle is stalled
- Cycle x+2, 2nd AHB data phase: Registered data is logically right-aligned for proper alignment and driven onto the input read data bus

NOTE

Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

17.3.2.1 Decorated load: load-and-clear 1 bit (LAC1)

This command loads a 1-bit field defined by the LSB position (b) into the core's general purpose destination register (Rt) and zeroes the bit in the memory space after performing an atomic read-modify-write sequence.

The extracted 1-bit data field from the memory address is right-justified and zero-filled in the operand returned to the core.

The data size is specified by the read operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
iolaclb	0	*	*	0	1	0	-	-	b	b	b	-	mem_addr																			
iolaclh	0	*	*	0	1	0	-	b	b	b	b	-	mem_addr																			0
iolaclw	0	*	*	0	1	0	b	b	b	b	b	-	mem_addr																		0	0

Figure 17-9. Decorated load address: load-and-clear 1 bit

See Figure 17-9, where addr[30:29] = 10 for peripheral, addr[28:26] = 010 specifies the load-and-clear 1 bit operation, addr[25:21] is "b", the bit identifier, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care".

The decorated load-and-clear 1-bit read operation is defined in the following pseudo-code as:

```
rdata =  iolac1<sz>(accessAddress)              // decorated load-and-clear 1

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
mask  =  1 << b                                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
tmp   =  tmp & ~mask                            // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp     // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 17-5. Cycle definitions of decorated load: load-and-clear 1 bit

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Extract bit from rdata; Form (rdata & ~mask) and capture destination data in register	Return extracted bit to master; Perform write sending registered data to memory

17.3.2.2 Decorated Load: Load-and-Set 1 Bit (LAS1)

This command loads a 1-bit field defined by the LSB position (b) into the core's general purpose destination register (Rt) and sets the bit in the memory space after performing an atomic read-modify-write sequence.

The extracted one bit data field from the memory address is right justified and zero filled in the operand returned to the core.

The data size is specified by the read operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

	31	30	29	28	27	26	25	24	23	22	21	20	19 … 2	1	0
iolaslb	0	*	*	0	1	1	-	-	b	b	b	-	mem_addr		
iolaslh	0	*	*	0	1	1	-	b	b	b	b	-	mem_addr		0
iolaslw	0	*	*	0	1	1	b	b	b	b	b	-	mem_addr	0	0

Figure 17-10. Decorated load address: load-and-set 1 bit

where addr[30:29] = 10 for peripheral, addr[28:26] = 011 specifies the load-and-set 1 bit operation, addr[25:21] is "b", the bit identifier, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care".

The decorated Load-and-Set 1 Bit read operation is defined in the following pseudo-code as:

```
rdata =  iolas1<sz>(accessAddress)              // decorated load-and-set 1

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
mask  =  1 << b                                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
```

```
tmp    =  tmp |  mask                              // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp      // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 17-6. Cycle definitions of decorated load: load-and-set 1-bit

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Extract bit from rdata; Form (rdata I mask) and capture destination data in register	Return extracted bit to master; Perform write sending registered data to memory

17.3.2.3 Decorated load unsigned bit field extract (UBFX)

This command extracts a bit field defined by LSB position (b) and the bit field width (w +1) from the memory "container" defined by the access size associated with the load instruction using a two-cycle read sequence.

The extracted bit field from the memory address is right-justified and zero-filled in the operand returned to the core. Recall this is the only decorated operation that does not perform a memory write, that is, UBFX only performs a read.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). Note for the word sized operation, the maximum bit field width is 16 bits.

The use of a UBFX operation is recommended to extract a single bit. For this case, the w field is simply set to 0, indicating a bit field width of 1.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ioubfxb	0	*	*	1	-	-	b	b	b	-	w	w	w	mem_addr																		
ioubfxh	0	*	*	1	-	b	b	b	b	w	w	w	w	mem_addr																		0
ioubfxw	0	*	*	1	b	b	b	b	b	w	w	w	w	mem_addr																	0	0

Figure 17-11. Decorated load address: unsigned bit field extract

See Figure 17-11, where addr[30:29] = 10 for peripheral, addr[28] = 1 specifies the unsigned bit field extract operation, addr[27:23] is "b", the LSB identifier, addr[22:19] is "w", the bit field width minus 1 identifier, and mem_addr[18:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care". Note, unlike the other decorated load operations, UBFX uses addr[19] as the least significant bit in the "w" specifier and not as an address bit.

The decorated unsigned bit field extract read operation is defined in the following pseudo-code as:

```
rdata =  ioubfx<sz>(accessAddress)              // unsigned bit field extract

tmp   =  mem[accessAddress & 0xE007FFFF, size]  // memory read
mask  = ((1 << (w+1)) - 1) << b                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
```

Like the BFI operation, when the starting bit position plus the field width exceeds the container size, only part of the source bit field is extracted from the destination memory location. Stated differently, if (b + w+1) > container_width, only the low-order "container_width - b" bits are actually extracted. The cycle-by-cycle BME operations are detailed in the following table.

Table 17-7. Cycle definitions of decorated load: unsigned bit field extract

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Idle AHB address phase	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Form (rdata & mask) and capture destination data in register	Logically right shift registered data; Return justified rdata to master

17.3.3 Additional details on decorated addresses and GPIO accesses

As previously noted, the peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses. This memory layout provides compatibility with the Kinetis K Family and provides 129 address "slots", each 4 KB in size.

The GPIO address space is multiply-mapped by the hardware: it appears at the "standard" system address 0x400F_F000 and is physically located in the address slot corresponding to address 0x4000_F000. Decorated loads and stores create a slight complication involving accesses to the GPIO. Recall the use of address[19] varies by decorated operation; for AND, OR, XOR, LAC1 and LAS1, this bit functions as a true address bit, while for BFI and UBFX, this bit defines the least significant bit of the "w" bit field specifier.

As a result, undecorated GPIO references and decorated AND, OR, XOR, LAC1 and LAS1 operations can use the standard 0x400F_F000 base address, while decorated BFI and UBFX operations must use the alternate 0x4000_F000 base address. Another implementation can simply use 0x400F_F000 as the base address for all undecorated GPIO accesses and 0x4000_F000 as the base address for all decorated accesses. Both implementations are supported by the hardware.

Table 17-8. Decorated peripheral and GPIO address details

Peripheral address space	Description
0x4000_0000–0x4007_FFFF	Undecorated (normal) peripheral accesses
0x4008_0000–0x400F_EFFF	Illegal addresses; attempted references are aborted and error terminated
0x400F_F000–0x400F_FFFF	Undecorated (normal) GPIO accesses using standard address
0x4010_0000–0x43FF_FFFF	Illegal addresses; attempted references are aborted and error terminated
0x4400_0000–0x4FFF_FFFF	Decorated AND, OR, XOR, LAC1, LAS1 references to peripherals and GPIO based at either 0x4000_F000 or 0x400F_F000
0x5000_0000–0x5FFF_FFFF	Decorated BFI, UBFX references to peripherals and GPIO only based at 0x4000_F000

17.4 Application information

In this section, GNU assembler macros with C expression operands are presented as examples of the required instructions to perform decorated operations.

This section specifically presents a partial bme.h file defining the assembly language expressions for decorated logical stores: AND, OR, and XOR. Comparable functions for BFI and the decorated loads are more complex and available in the complete BME header file.

These macros use the same function names presented in Functional description.

```
#define IOANDW(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<26);"        \
          "orr    r3, %[addr];"         \
          "mov    r2, %[wdata];"        \
          "str    r2, [r3];"            \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOANDH(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<26);"        \
          "orr    r3, %[addr];"         \
          "mov    r2, %[wdata];"        \
          "strh   r2, [r3];"            \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOANDB(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<26);"        \
          "orr    r3, %[addr];"         \
          "mov    r2, %[wdata];"        \
          "strb   r2, [r3];"            \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");
```

```
#define IOORW(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<27);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "str    r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOORH(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<27);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strh   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOORB(ADDR,WDATA)              \
    __asm("ldr    r3, =(1<<27);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strb   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORW(ADDR,WDATA)             \
    __asm("ldr    r3, =(3<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "str    r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORH(ADDR,WDATA)             \
    __asm("ldr    r3, =(3<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strh   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORB(ADDR,WDATA)             \
    __asm("ldr    r3, =(3<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strb   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");
```

Chapter 18
Miscellaneous Control Module (MCM)

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions for the platform (RPP).

18.1 Introduction

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

18.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration
- Crossbar master arbitration policy selection
- Flash controller speculation buffer and cache configurations

18.2 Memory map/register descriptions

The memory map and register descriptions found here describe the registers using byte addresses. The registers can be written only when in supervisor mode.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_3008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	0007h	18.2.1/320

Table continues on the next page...

MCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_300A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	0005h	18.2.2/321
F000_300C	Platform Control Register (MCM_PLACR)	32	R/W	0000_0050h	18.2.3/321
F000_3040	Compute Operation Control Register (MCM_CPO)	32	R/W	0000_0000h	18.2.4/324

18.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device’s crossbar switch.

Address: F000_3000h base + 8h offset = F000_3008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent. 1 A bus slave connection to AXBS input port *n* is present.

18.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: F000_3000h base + Ah offset = F000_300Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port. 0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

18.2.3 Platform Control Register (MCM_PLACR)

The PLACR register selects the arbitration policy for the crossbar masters and configures the flash memory controller.

The speculation buffer and cache in the flash memory controller is configurable via PLACR[15:10].

The speculation buffer is enabled only for instructions after reset. It is possible to have these states for the speculation buffer:

DFCS	EFDS	Description
0	0	Speculation buffer is on for instruction and off for data.
0	1	Speculation buffer is on for instruction and on for data.
1	X	Speculation buffer is off.

The cache in flash controller is enabled and caching both instruction and data type fetches after reset. It is possible to have these states for the cache:

DFCC	DFCIC	DFCDA	Description
0	0	0	Cache is on for both instruction and data.
0	0	1	Cache is on for instruction and off for data.
0	1	0	Cache is off for instruction and on for data.
0	1	1	Cache is off for both instruction and data.
1	X	X	Cache is off.

Address: F000_3000h base + Ch offset = F000_300Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															ESFC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DFCS	EFDS	DFCC	DFCIC	DFCDA	0	ARB	0								
W						CFCC										
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

MCM_PLACR field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ESFC	Enable Stalling Flash Controller Enables stalling flash controller when flash is busy.

Table continues on the next page...

MCM_PLACR field descriptions (continued)

Field	Description
	When software needs to access the flash memory while a flash memory resource is being manipulated by a flash command, software can enable a stall mechanism to avoid a read collision. The stall mechanism allows software to execute code from the same block on which flash operations are being performed. However, software must ensure the sector the flash operations are being performed on is not the same sector from which the code is executing. ESFC enables the stall mechanism. This bit must be set only just before the flash operation is executed and must be cleared when the operation completes. 0 Disable stalling flash controller when flash is busy. 1 Enable stalling flash controller when flash is busy.
15 DFCS	Disable Flash Controller Speculation Disables flash controller speculation. 0 Enable flash controller speculation. 1 Disable flash controller speculation.
14 EFDS	Enable Flash Data Speculation Enables flash data speculation. 0 Disable flash data speculation. 1 Enable flash data speculation.
13 DFCC	Disable Flash Controller Cache Disables flash controller cache. 0 Enable flash controller cache. 1 Disable flash controller cache.
12 DFCIC	Disable Flash Controller Instruction Caching Disables flash controller instruction caching. 0 Enable flash controller instruction caching. 1 Disable flash controller instruction caching.
11 DFCDA	Disable Flash Controller Data Caching Disables flash controller data caching. 0 Enable flash controller data caching 1 Disable flash controller data caching.
10 CFCC	Clear Flash Controller Cache Writing a 1 to this field clears the cache. Writing a 0 to this field is ignored. This field always reads as 0.
9 ARB	Arbitration select 0 Fixed-priority arbitration for the crossbar masters 1 Round-robin arbitration for the crossbar masters
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

18.2.4 Compute Operation Control Register (MCM_CPO)

This register controls the Compute Operation.

Address: F000_3000h base + 40h offset = F000_3040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CPOWOI	CPOACK	CPOREQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_CPO field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CPOWOI	Compute Operation Wake-up on Interrupt 0 No effect. 1 When set, the CPOREQ is cleared on any interrupt or exception vector fetch.
1 CPOACK	Compute Operation Acknowledge 0 Compute operation entry has not completed or compute operation exit has completed. 1 Compute operation entry has completed or compute operation exit has not completed.
0 CPOREQ	Compute Operation Request This bit is auto-cleared by vector fetching if CPOWOI = 1.

Table continues on the next page...

MCM_CPO field descriptions (continued)

Field	Description
	0 Request is cleared. 1 Request Compute Operation.

Chapter 19
Micro Trace Buffer (MTB)

This module explains how microcontrollers using the Cortex-M0+ processor core support CoreSight Micro Trace Buffer to provide program trace capabilities. It also explains about change-of-flow data packets in a user-defined region of the system RAM and DWT (Data Watchpoint and Trace) module that allows a user to define watchpoint addresses, or an address and data value, that when triggered can be used to start or stop the program trace recording.

19.1 Introduction

Microcontrollers using the Cortex-M0+ processor core include support for a CoreSight Micro Trace Buffer to provide program trace capabilities.

The proper name for this function is the CoreSight Micro Trace Buffer for the Cortex-M0+ Processor; in this document, it is simply abbreviated as the MTB.

The simple program trace function creates instruction address change-of-flow data packets in a user-defined region of the system RAM. Accordingly, the system RAM controller manages requests from two sources:

- AMBA-AHB reads and writes from the system bus
- program trace packet writes from the processor

As part of the MTB functionality, there is a DWT (Data Watchpoint and Trace) module that allows the user to define watchpoint addresses, or optionally, an address and data value, that when triggered, can be used to start or stop the program trace recording.

This document details the functionality of both the MTB_RAM and MTB_DWT capabilities.

19.1.1 Overview

A generic block diagram of the processor core and platform for this class of ultra low-end microcontrollers is shown as follows:

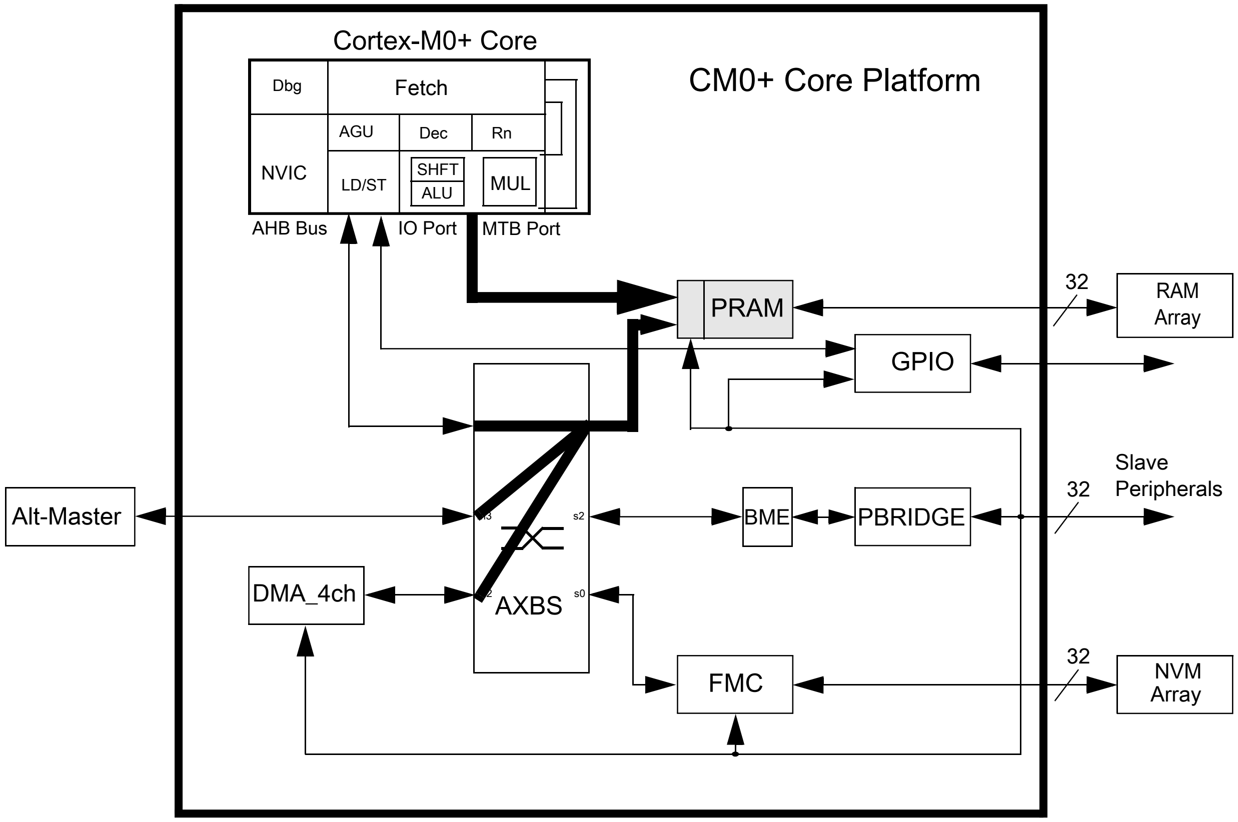


Figure 19-1. Generic Cortex-M0+ core platform block diagram

As shown in the block diagram, the platform RAM (PRAM) controller connects to two input buses:

- the crossbar slave port for system bus accesses
- a "private execution MTB port" from the core

The logical paths from the crossbar master input ports to the PRAM controller are highlighted along with the private execution trace port from the processor core. The private MTB port signals the instruction address information needed for the 64-bit program trace packets written into the system RAM. The PRAM controller output interfaces to the attached RAM array. In this document, the PRAM controller is the MTB_RAM controller.

The following information is taken from the ARM CoreSight Micro Trace Buffer documentation.

"The execution trace packet consists of a pair of 32-bit words that the MTB generates when it detects the processor PC value changes non-sequentially. A non-sequential PC change can occur during branch instructions or during exception entry.

The processor can cause a trace packet to be generated for any instruction.

The following figure shows how the execution trace information is stored in memory as a sequence of packets.

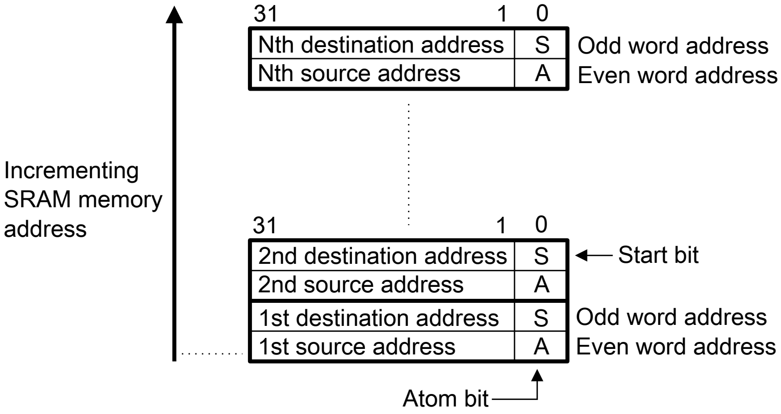


Figure 19-2. MTB execution trace storage format

The first, lower addressed, word contains the source of the branch, the address it branched from. The value stored only records bits[31:1] of the source address, because Thumb instructions are at least halfword aligned. The least significant bit of the value is the A-bit. The A-bit indicates the atomic state of the processor at the time of the branch, and can differentiate whether the branch originated from an instruction in a program, an exception, or a PC update in Debug state. When it is zero the branch originated from an instruction, when it is one the branch originated from an exception or PC update in Debug state. This word is always stored at an even word location.

The second, higher addressed word contains the destination of the branch, the address it branched to. The value stored only records bits[31:1] of the branch address. The least significant bit of the value is the S-bit. The S-bit indicates where the trace started. An S-bit value of 1 indicates where the first packet after the trace started and a value of 0 is used for other packets. Because it is possible to start and stop tracing multiple times in a trace session, the memory might contain several packets with the S-bit set to 1. This word is always stored in the next higher word in memory, an odd word address.

When the A-bit is set to 1, the source address field contains the architecturally-preferred return address for the exception. For example, if an exception was caused by an SVC instruction, then the source address field contains the address of the following instruction. This is different from the case where the A-bit is set to 0. In this case, the source address contains the address of the branch instruction.

For an exception return operation, two packets are generated:

- The first packet has the:
    - Source address field set to the address of the instruction that causes the exception return, BX or POP.
    - Destination address field set to bits[31:1] of the EXC_RETURN value. See the ARM v6-M Architecture Reference Manual.
    - The A-bit set to 0.
- The second packet has the:
    - Source address field set to bits[31:1] of the EXC_RETURN value.
    - Destination address field set to the address of the instruction where execution commences.
    - A-bit set to 1."

Given the recorded change-of-flow trace packets in system RAM and the memory image of the application, a debugger can read out the data and create an instruction-by-instruction program trace. In keeping with the low area and power implementation cost design targets, the MTB trace format is less efficient than other CoreSight trace modules, for example, the ETM (Embedded Trace Macrocell). Since each branch packet is 8 bytes in size, a 1 KB block of system RAM can contain 128 branches. Using the Dhrystone 2.1 benchmark's dynamic runtime as an example, this corresponds to about 875 instructions per KB of trace RAM, or with a zero wait state memory, this corresponds to approximately 1600 processor cycles per KB. This metric is obviously very sensitive to the runtime characteristics of the user code.

The MTB_DWT function (not shown in the core platform block diagram) monitors the processor address and data buses so that configurable watchpoints can be detected to trigger the appropriate response in the MTB recording.

19.1.2 Features

The key features of the MTB_RAM and MTB_DWT include:

- Memory controller for system RAM and Micro Trace Buffer for program trace packets
- Read/write capabilities for system RAM accesses, write-only for program trace packets
- Supports zero wait state response to system bus accesses when no trace data is being written
- Can buffer two AHB address phases and one data write for system RAM accesses
- Supports 64-bit program trace packets including source and destination instruction addresses

- Program trace information in RAM available to MCU's application code or external debugger
- Program trace watchpoint configuration accessible by MCU's application code or debugger
- Location and size of RAM trace buffer is configured by software
- Two DWT comparators (addresses or address + data) provide programmable start/ stop recording
- CoreSight compliant debug functionality

19.1.3 Modes of operation

The MTB_RAM and MTB_DWT functions do not support any special modes of operation. The MTB_RAM controller, as a memory-mapped device located on the platform's slave AHB system bus, responds strictly on the basis of memory addresses for accesses to its attached RAM array. The MTB private execution bus provides program trace packet write information to the RAM controller. Both the MTB_RAM and MTB_DWT modules are memory-mapped, so their programming models can be accessed.

All functionality associated with the MTB_RAM and MTB_DWT modules resides in the core platform's clock domain; this includes its connections with the RAM array.

19.2 External signal description

The MTB_RAM and MTB_DWT modules do not directly support any external interfaces.

The internal interface includes a standard AHB bus with a 32-bit datapath width from the appropriate crossbar slave port plus the private execution trace bus from the processor core. The signals in the private execution trace bus are detailed in the following table taken from the ARM CoreSight Micro Trace Buffer documentation. The signal direction is defined as viewed by the MTB_RAM controller.

Table 19-1. Private execution trace port from the core to MTB_RAM

Signal	Direction	Description
LOCKUP	Input	Indicates the processor is in the Lockup state. This signal is driven LOW for cycles when the processor is executing normally and driven HIGH for every cycle the processor is waiting in the Lockup state. This signal is valid on every cycle.
IAESEQ	Input	Indicates the next instruction address in execute, IAEX, is sequential, that is non-branching.

Table continues on the next page...

Table 19-1. Private execution trace port from the core to MTB_RAM (continued)

Signal	Direction	Description
IAEXEN	Input	IAEX register enable.
IAEX[30:0]	Input	Registered address of the instruction in the execution stage, shifted right by one bit, that is, PC >> 1.
ATOMIC	Input	Indicates the processor is performing non-instruction related activities.
EDBGRQ	Output	Request for the processor to enter the Debug state, if enabled, and halt.

In addition, there are two signals formed by the MTB_DWT module and driven to the MTB_RAM controller: TSTART (trace start) and TSTOP (trace stop). These signals can be configured using the trace watchpoints to define programmable addresses and data values to affect the program trace recording state.

19.3 Memory map and register definition

The MTB_RAM and MTB_DWT modules each support a sparsely-populated 4 KB address space for their programming models. For each address space, there are a variety of control and configurable registers near the base address, followed by a large unused address space and finally a set of CoreSight registers to support dynamic determination of the debug configuration for the device.

Accesses to the programming model follow standard ARM conventions. Taken from the ARM CoreSight Micro Trace Buffer documentation, these are:

- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in UNPREDICTABLE behavior.
- The behavior of the MTB is UNPREDICTABLE if the registers with UNKNOWN reset values are not programmed prior to enabling trace.
- Unless otherwise stated in the accompanying text:
  - Do not modify reserved register bits
  - Ignore reserved register bits on reads
  - All register bits are reset to a logic 0 by a system or power-on reset
  - Use only word size, 32-bit, transactions to access all registers

19.3.1 MTB_RAM Memory Map

MTB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_0000	MTB Position Register (MTB_POSITION)	32	R/W	Undefined	19.3.1.1/ 334
F000_0004	MTB Master Register (MTB_MASTER)	32	R/W	See section	19.3.1.2/ 336
F000_0008	MTB Flow Register (MTB_FLOW)	32	R/W	Undefined	19.3.1.3/ 337
F000_000C	MTB Base Register (MTB_BASE)	32	R	Undefined	19.3.1.4/ 339
F000_0F00	Integration Mode Control Register (MTB_MODECTRL)	32	R	0000_0000h	19.3.1.5/ 340
F000_0FA0	Claim TAG Set Register (MTB_TAGSET)	32	R	0000_0000h	19.3.1.6/ 340
F000_0FA4	Claim TAG Clear Register (MTB_TAGCLEAR)	32	R	0000_0000h	19.3.1.7/ 341
F000_0FB0	Lock Access Register (MTB_LOCKACCESS)	32	R	0000_0000h	19.3.1.8/ 341
F000_0FB4	Lock Status Register (MTB_LOCKSTAT)	32	R	0000_0000h	19.3.1.9/ 342
F000_0FB8	Authentication Status Register (MTB_AUTHSTAT)	32	R	0000_0000h	19.3.1.10/ 342
F000_0FBC	Device Architecture Register (MTB_DEVICEARCH)	32	R	4770_0A31h	19.3.1.11/ 343
F000_0FC8	Device Configuration Register (MTB_DEVICECFG)	32	R	0000_0000h	19.3.1.12/ 343
F000_0FCC	Device Type Identifier Register (MTB_DEVICETYPID)	32	R	0000_0031h	19.3.1.13/ 344
F000_0FD0	Peripheral ID Register (MTB_PERIPHID4)	32	R	See section	19.3.1.14/ 344
F000_0FD4	Peripheral ID Register (MTB_PERIPHID5)	32	R	See section	19.3.1.14/ 344
F000_0FD8	Peripheral ID Register (MTB_PERIPHID6)	32	R	See section	19.3.1.14/ 344
F000_0FDC	Peripheral ID Register (MTB_PERIPHID7)	32	R	See section	19.3.1.14/ 344
F000_0FE0	Peripheral ID Register (MTB_PERIPHID0)	32	R	See section	19.3.1.14/ 344
F000_0FE4	Peripheral ID Register (MTB_PERIPHID1)	32	R	See section	19.3.1.14/ 344
F000_0FE8	Peripheral ID Register (MTB_PERIPHID2)	32	R	See section	19.3.1.14/ 344
F000_0FEC	Peripheral ID Register (MTB_PERIPHID3)	32	R	See section	19.3.1.14/ 344

Table continues on the next page...

MTB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_0FF0	Component ID Register (MTB_COMPID0)	32	R	See section	19.3.1.15/ 345
F000_0FF4	Component ID Register (MTB_COMPID1)	32	R	See section	19.3.1.15/ 345
F000_0FF8	Component ID Register (MTB_COMPID2)	32	R	See section	19.3.1.15/ 345
F000_0FFC	Component ID Register (MTB_COMPID3)	32	R	See section	19.3.1.15/ 345

19.3.1.1 MTB Position Register (MTB_POSITION)

The MTB_POSITION register contains the Trace Write Address Pointer and Wrap fields. This register can be modified by the explicit programming model writes. It is also automatically updated by the MTB hardware when trace packets are being recorded.

The base address of the system RAM in the memory map dictates special consideration for the placement of the MTB. Consider the following guidelines:

For the standard configuration where the size of the MTB is ≤ 25% of the total RAM capacity, it is recommended the MTB be based at the address defined by the MTB_BASE register. The read-only MTB_BASE register is defined by the expression (0x2000_0000 - (RAM_Size/4)). For this configuration, the MTB_POSITION register is initialized to MTB_BASE & 0x0000_7FF8.

If the size of the MTB is more than 25% but less than or equal to 50% of the total RAM capacity, it is recommended the MTB be based at address 0x2000_0000. In this configuration, the MTB_POSITION register is initialized to (0x2000_0000 & 0x0000_7FF8) = 0x0000_00000.

Following these two suggested placements provides a full-featured circular memory buffer containing program trace packets.

In the unlikely event an even larger trace buffer is required, a write-once capacity of 75% of the total RAM capacity can be based at address 0x2000_0000. The MTB_POSITION register is initialized to (0x2000_0000 & 0x0000_7FF8) = 0x0000_0000. However, this configuration cannot support operation as a circular queue and instead requires the use of the MTB_FLOW[WATERMARK] capability to automatically disable tracing or halting the processor as the number of packet writes approach the buffer capacity. See the MTB_FLOW register description for more details.

Address: F000_0000h base + 0h offset = F000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POINTER															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POINTER													WRAP	0	
W															▨	
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	0	0

* Notes:

- x = Undefined at reset.

MTB_POSITION field descriptions

Field	Description

31–3
POINTER
	Trace Packet Address Pointer[28:0]
	Because a packet consists of two words, the POINTER field is the address of the first word of a packet. This field contains bits[31:3] of the RAM address where the next trace packet is written. Therefore, it points to an unused location and is automatically incremented.
	A debug agent can calculate the system memory map address for the current location in the MTB using the following "generic" equation:
	Given mtb_size = 1 << (MTB_MASTER[MASK] + 4),
	systemAddress = MTB_BASE + (((MTB_POSITION & 0xFFFF_FFF8) + (mtb_size - (MTB_BASE & (mtb_size-1)))) & 0x0000_7FF8);
	For this device, a simpler expression also applies. See the following pseudo-code:
	if ((MTB_POSITION >> 13) == 0x3) systemAddress = (0x1FFF << 16) + (0x1 << 15) + (MTB_POSITION & 0x7FF8); else systemAddress = (0x2000 << 16) + (0x0 << 15) + (MTB_POSITION & 0x7FF8);
	NOTE: The size of the RAM is parameterized and the most significant bits of the POINTER field are RAZ/WI.
	For these devices, POSITION[31:15] == POSITION[POINTER[28:12]] are RAZ/WI. Therefore, the active bits in this field are POSITION[14:3] == POSITION[POINTER[11:0]].

2 WRAP	WRAP This field is set to 1 automatically when the POINTER value wraps as determined by the MTB_MASTER[MASK] field in the MASTER Trace Control Register. A debug agent might use the WRAP field to determine whether the trace information above and below the pointer address is valid.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.3.1.2 MTB Master Register (MTB_MASTER)

The MTB_MASTER register contains the main program trace enable plus other trace controls. This register can be modified by the explicit programming model writes. MTB_MASTER[EN] and MTB_MASTER[HALTREQ] fields are also automatically updated by the MTB hardware.

Before MTB_MASTER[EN] or MTB_MASTER[TSTARTEN] are set to 1, the software must initialize the MTB_POSITION and MTB_FLOW registers.

If MTB_FLOW[WATERMARK] is used to stop tracing or to halt the processor, MTB_MASTER[MASK] must still be set to a value that prevents MTB_POSITION[POINTER] from wrapping before it reaches the MTB_FLOW[WATERMARK] value.

NOTE

The format of this mask field is different than MTBDWT_MASKn[MASK].

Address: F000_0000h base + 4h offset = F000_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						HALTREQ	RAMPRIV	SFRWPRIV	TSTOPEN	TSTARTEN	MASK				
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

MTB_MASTER field descriptions

Field	Description
31 EN	Main Trace Enable When this field is 1, trace data is written into the RAM memory location addressed by MTB_POSITION[POINTER]. The MTB_POSITION[POINTER] value auto increments after the trace data packet is written. EN can be automatically set to 0 using the MTB_FLOW[WATERMARK] field and the MTB_FLOW[AUTOSTOP] bit.

Table continues on the next page...

MTB_MASTER field descriptions (continued)

Field	Description
	EN is automatically set to 1 if TSTARTEN is 1 and the TSTART signal is HIGH. EN is automatically set to 0 if TSTOPEN is 1 and the TSTOP signal is HIGH. **NOTE:** If EN is set to 0 because MTB_FLOW[WATERMARK] is set, then it is not automatically set to 1 if TSTARTEN is 1 and the TSTART input is HIGH. In this case, tracing can only be restarted if MTB_FLOW[WATERMARK] or MTB_POSITION[POINTER] value is changed by software.
30–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 HALTREQ	Halt Request This field is connected to the halt request signal of the trace logic, EDBGRQ. When HALTREQ is set to 1, the EDBFGRQ is asserted if DBGEN (invasive debug enable, one of the debug authentication interface signals) is also HIGH. HALTREQ can be automatically set to 1 using MTB_FLOW[WATERMARK].
8 RAMPRIV	RAM Privilege If this field is 0, then user or privileged AHB read and write accesses to the RAM are permitted. If this field is 1, then only privileged AHB read and write accesses to the RAM are permitted and user accesses are RAZ/WI. The HPROT[1] signal determines if an access is a user or privileged mode reference.
7 SFRWPRIV	Special Function Register Write Privilege If this field is 0, then user or privileged AHB read and write accesses to the MTB_RAM Special Function Registers (programming model) are permitted. If this field is 1, then only privileged write accesses are permitted; user write accesses are ignored. The HPROT[1] signal determines if an access is user or privileged. Note MTB_RAM SFR read access are not controlled by this bit and are always permitted.
6 TSTOPEN	Trace Stop Input Enable If this field is 1 and the TSTOP signal is HIGH, then EN is set to 0. If a trace packet is being written to memory, the write is completed before tracing is stopped.
5 TSTARTEN	Trace Start Input Enable If this field is 1 and the TSTART signal is HIGH, then EN is set to 1. Tracing continues until a stop condition occurs.
MASK	Mask This value determines the maximum size of the trace buffer in RAM. It specifies the most-significant bit of the MTB_POSITION[POINTER] field that can be updated by automatic increment. If the trace tries to advance past this power of 2, the MTB_POSITION[WRAP] bit is set to 1, the MTB_POSITION[MASK+3:3] == MTB_POSITION[POINTER[MASK:0]] bits are set to 0, and the MTB_POSITION[14:MASK+3] == MTB_POSITION[POINTER[11:MASK+1]] bits remain unchanged. This field causes the trace packet information to be stored in a circular buffer of size 2^[MASK+4] bytes, that can be positioned in memory at multiples of this size. As detailed in the MTB_POSITION description, typical "upper limits" for the MTB size are RAM_Size/4 or RAM_Size/2. Values greater than the maximum have the same effect as the maximum.

19.3.1.3 MTB Flow Register (MTB_FLOW)

The MTB_FLOW register contains the watermark address and the autostop/autohalt control bits.

If tracing is stopped using the watermark autostop feature, it cannot be restarted until software clears the watermark autostop. This can be achieved in one of the following ways:

- Changing the MTB_POSITION[POINTER] field value to point to the beginning of the trace buffer, or
- Setting MTB_FLOW[AUTOSTOP] = 0.

A debug agent can use MTB_FLOW[AUTOSTOP] to fill the trace buffer once only without halting the processor.

A debug agent can use MTB_FLOW[AUTOHALT] to fill the trace buffer once before causing the Cortex-M0+ processor to enter the Debug state. To enter Debug state, the Cortex-M0+ processor might have to perform additional branch type operations. Therefore, the MTB_FLOW[WATERMARK] field must be set below the final entry in the trace buffer region.

Address: F000_0000h base + 8h offset = F000_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WATERMARK															
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WATERMARK													0	AUTOHALT	AUTOSTOP
W														▨		
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	0	x*	x*

* Notes:

- x = Undefined at reset.

MTB_FLOW field descriptions

Field	Description
31–3 WATERMARK	WATERMARK[28:0] This field contains an address in the same format as the MTB_POSITION[POINTER] field. When MTB_POSITION[POINTER] matches the WATERMARK field value, actions defined by the AUTOHALT and AUTOSTOP bits are performed.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MTB_FLOW field descriptions (continued)

Field	Description
1 AUTOHALT	AUTOHALT If this field is 1 and WATERMARK is equal to MTB_POSITION[POINTER], then MTB_MASTER[HALTREQ] is automatically set to 1. If the DBGEN signal is HIGH, the MTB asserts this halt request to the Cortex-M0+ processor by asserting the EDBGRQ signal.
0 AUTOSTOP	AUTOSTOP If this field is 1 and WATERMARK is equal to MTB_POSITION[POINTER], then MTB_MASTER[EN] is automatically set to 0. This stops tracing.

19.3.1.4 MTB Base Register (MTB_BASE)

The read-only MTB_BASE Register indicates where the RAM is located in the system memory map. This register is provided to enable auto discovery of the MTB RAM location, by a debug agent and is defined by a hardware design parameter. For this device, the base address is defined by the expression: MTB_BASE[BASEADDR] = 0x2000_0000 - (RAM_Size/4)

Address: F000_0000h base + Ch offset = F000_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BASEADDR																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

MTB_BASE field descriptions

Field	Description
BASEADDR	BASEADDR This value is defined with a hardwired signal and the expression: 0x2000_0000 - (RAM_Size/4). For example, if the total RAM capacity is 16 KB, this field is 0x1FFF_F000.

19.3.1.5 Integration Mode Control Register (MTB_MODECTRL)

This register enables the device to switch from a functional mode, or default behavior, into integration mode. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + F00h offset = F000_0F00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MODECTRL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_MODECTRL field descriptions

Field	Description
MODECTRL	MODECTRL Hardwired to 0x0000_0000

19.3.1.6 Claim TAG Set Register (MTB_TAGSET)

The Claim Tag Set Register returns the number of bits that can be set on a read, and enables individual bits to be set on a write. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FA0h offset = F000_0FA0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TAGSET																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_TAGSET field descriptions

Field	Description
TAGSET	TAGSET Hardwired to 0x0000_0000

19.3.1.7 Claim TAG Clear Register (MTB_TAGCLEAR)

The read/write Claim Tag Clear Register is used to read the claim status on debug resources. A read indicates the claim tag status. Writing 1 to a specific bit clears the corresponding claim tag to 0. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FA4h offset = F000_0FA4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TAGCLEAR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_TAGCLEAR field descriptions

Field	Description
TAGCLEAR	TAGCLEAR Hardwired to 0x0000_0000

19.3.1.8 Lock Access Register (MTB_LOCKACCESS)

The Lock Access Register enables a write access to component registers. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FB0h offset = F000_0FB0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKACCESS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_LOCKACCESS field descriptions

Field	Description
LOCKACCESS	Hardwired to 0x0000_0000

19.3.1.9 Lock Status Register (MTB_LOCKSTAT)

The Lock Status Register indicates the status of the lock control mechanism. This register is used in conjunction with the Lock Access Register. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FB4h offset = F000_0FB4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKSTAT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_LOCKSTAT field descriptions

Field	Description
LOCKSTAT	LOCKSTAT Hardwired to 0x0000_0000

19.3.1.10 Authentication Status Register (MTB_AUTHSTAT)

The Authentication Status Register reports the required security level and current status of the security enable bit pairs. Where functionality changes on a given security level, this change must be reported in this register. It is connected to specific signals used during the auto-discovery process by an external debug agent.

MTB_AUTHSTAT[3:2] indicates if nonsecure, noninvasive debug is enabled or disabled, while MTB_AUTHSTAT[1:0] indicates the enabled/disabled state of nonsecure, invasive debug. For both 2-bit fields, 0b10 indicates the functionality is disabled and 0b11 indicates it is enabled.

Address: F000_0000h base + FB8h offset = F000_0FB8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												1	BIT2	1	BIT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_AUTHSTAT field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	BIT3 This read-only field is reserved and always has the value 1.
2 BIT2	BIT2 Connected to NIDEN or DBGEN signal.
1 Reserved	BIT1 This read-only field is reserved and always has the value 1.
0 BIT0	Connected to DBGEN.

19.3.1.11 Device Architecture Register (MTB_DEVICEARCH)

This register indicates the device architecture. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FBCh offset = F000_0FBCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICEARCH																															
W																																
Reset	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1

MTB_DEVICEARCH field descriptions

Field	Description
DEVICEARCH	DEVICEARCH Hardwired to 0x4770_0A31.

19.3.1.12 Device Configuration Register (MTB_DEVICECFG)

This register indicates the device configuration. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FC8h offset = F000_0FC8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICECFG																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DEVICECFG field descriptions

Field	Description
DEVICECFG	DEVICECFG Hardwired to 0x0000_0000.

19.3.1.13 Device Type Identifier Register (MTB_DEVICETYPID)

This register indicates the device type ID. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FCCh offset = F000_0FCCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICETYPID																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

MTB_DEVICETYPID field descriptions

Field	Description
DEVICETYPID	DEVICETYPID Hardwired to 0x0000_0031.

19.3.1.14 Peripheral ID Register (MTB_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FD0h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PERIPHID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

MTB_PERIPHID*n* field descriptions

Field	Description
PERIPHID	PERIPHID Peripheral ID4 is hardwired to 0x0000_0004; ID0 to 0x0000_0032; ID1 to 0x0000_00B9; ID2 to 0x0000_000B; and all the others to 0x0000_0000.

19.3.1.15 Component ID Register (MTB_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FF0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COMPID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

MTB_COMPID*n* field descriptions

Field	Description
COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0090; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

19.3.2 MTB_DWT Memory Map

The MTB_DWT programming model supports a very simplified subset of the v7M debug architecture and follows the standard ARM DWT definition.

MTB_DWT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1000	MTB DWT Control Register (MTB_DWT_CTRL)	32	R	2F00_0000h	19.3.2.1/ 346
F000_1020	MTB_DWT Comparator Register (MTB_DWT_COMP0)	32	R/W	0000_0000h	19.3.2.2/ 347
F000_1024	MTB_DWT Comparator Mask Register (MTB_DWT_MASK0)	32	R/W	0000_0000h	19.3.2.3/ 348
F000_1028	MTB_DWT Comparator Function Register 0 (MTB_DWT_FCT0)	32	R/W	0000_0000h	19.3.2.4/ 349
F000_1030	MTB_DWT Comparator Register (MTB_DWT_COMP1)	32	R/W	0000_0000h	19.3.2.2/ 347
F000_1034	MTB_DWT Comparator Mask Register (MTB_DWT_MASK1)	32	R/W	0000_0000h	19.3.2.3/ 348
F000_1038	MTB_DWT Comparator Function Register 1 (MTB_DWT_FCT1)	32	R/W	0000_0000h	19.3.2.5/ 351
F000_1200	MTB_DWT Trace Buffer Control Register (MTB_DWT_TBCTRL)	32	R/W	2000_0000h	19.3.2.6/ 352

Table continues on the next page...

MTB_DWT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1FC8	Device Configuration Register (MTB_DWT_DEVICECFG)	32	R	0000_0000h	19.3.2.7/ 354
F000_1FCC	Device Type Identifier Register (MTB_DWT_DEVICETYPID)	32	R	0000_0004h	19.3.2.8/ 354
F000_1FD0	Peripheral ID Register (MTB_DWT_PERIPHID4)	32	R	See section	19.3.2.9/ 355
F000_1FD4	Peripheral ID Register (MTB_DWT_PERIPHID5)	32	R	See section	19.3.2.9/ 355
F000_1FD8	Peripheral ID Register (MTB_DWT_PERIPHID6)	32	R	See section	19.3.2.9/ 355
F000_1FDC	Peripheral ID Register (MTB_DWT_PERIPHID7)	32	R	See section	19.3.2.9/ 355
F000_1FE0	Peripheral ID Register (MTB_DWT_PERIPHID0)	32	R	See section	19.3.2.9/ 355
F000_1FE4	Peripheral ID Register (MTB_DWT_PERIPHID1)	32	R	See section	19.3.2.9/ 355
F000_1FE8	Peripheral ID Register (MTB_DWT_PERIPHID2)	32	R	See section	19.3.2.9/ 355
F000_1FEC	Peripheral ID Register (MTB_DWT_PERIPHID3)	32	R	See section	19.3.2.9/ 355
F000_1FF0	Component ID Register (MTB_DWT_COMPID0)	32	R	See section	19.3.2.10/ 355
F000_1FF4	Component ID Register (MTB_DWT_COMPID1)	32	R	See section	19.3.2.10/ 355
F000_1FF8	Component ID Register (MTB_DWT_COMPID2)	32	R	See section	19.3.2.10/ 355
F000_1FFC	Component ID Register (MTB_DWT_COMPID3)	32	R	See section	19.3.2.10/ 355

19.3.2.1 MTB DWT Control Register (MTB_DWT_CTRL)

The MTBDWT_CTRL register provides read-only information on the watchpoint configuration for the MTB_DWT.

Address: F000_1000h base + 0h offset = F000_1000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NUMCMP				DWTCFGCTRL																											
W																																
Reset	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_CTRL field descriptions

Field	Description
31–28 NUMCMP	Number of comparators The MTB_DWT implements two comparators.
DWTCFGCTRL	DWT configuration controls This field is hardwired to 0xF00_0000, disabling all the remaining DWT functionality. The specific fields and their state are: MTBDWT_CTRL[27] = NOTRCPKT = 1, trace sample and exception trace is not supported MTBDWT_CTRL[26] = NOEXTTRIG = 1, external match signals are not supported MTBDWT_CTRL[25] = NOCYCCNT = 1, cycle counter is not supported MTBDWT_CTRL[24] = NOPRFCNT = 1, profiling counters are not supported MTBDWT_CTRL[22] = CYCEBTENA = 0, no POSTCNT underflow packets generated MTBDWT_CTRL[21] = FOLDEVTENA = 0, no folded instruction counter overflow events MTBDWT_CTRL[20] = LSUEVTENA = 0, no LSU counter overflow events MTBDWT_CTRL[19] = SLEEPEVTENA = 0, no sleep counter overflow events MTBDWT_CTRL[18] = EXCEVTENA = 0, no exception overhead counter events MTBDWT_CTRL[17] = CPIEVTENA = 0, no CPI counter overflow events MTBDWT_CTRL[16] = EXCTRCENA = 0, generation of exception trace disabled MTBDWT_CTRL[12] = PCSAMPLENA = 0, no periodic PC sample packets generated MTBDWT_CTRL[11:10] = SYNCTAP = 0, no synchronization packets MTBDWT_CTRL[9] = CYCTAP = 0, cycle counter is not supported MTBDWT_CTRL[8:5] = POSTINIT = 0, cycle counter is not supported MTBDWT_CTRL[4:1] = POSTPRESET = 0, cycle counter is not supported MTBDWT_CTRL[0] = CYCCNTENA = 0, cycle counter is not supported

19.3.2.2 MTB_DWT Comparator Register (MTB_DWT_COMP*n*)

The MTBDWT_COMPn registers provide the reference value for comparator n.

Address: F000_1000h base + 20h offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_COMP*n* field descriptions

Field	Description
COMP	Reference value for comparison If MTBDWT_COMP0 is used for a data value comparator and the access size is byte or halfword, the data value must be replicated across all appropriate byte lanes of this register. For example, if the data is a

MTB_DWT_COMP*n* field descriptions (continued)

Field	Description
	byte-sized "x" value, then COMP[31:24] = COMP[23:16] = COMP[15:8] = COMP[7:0] = "x". Likewise, if the data is a halfword-size "y" value, then COMP[31:16] = COMP[15:0] = "y".

19.3.2.3 MTB_DWT Comparator Mask Register (MTB_DWT_MASK*n*)

The MTBDWT_MASKn registers define the size of the ignore mask applied to the reference address for address range matching by comparator n. Note the format of this mask field is different than the MTB_MASTER[MASK].

Address: F000_1000h base + 24h offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											MASK				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_MASK*n* field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MASK	MASK The value of the ignore mask, 0-31 bits, is applied to address range matching. MASK = 0 is used to include all bits of the address in the comparison, except if MASK = 0 and the comparator is configured to watch instruction fetch addresses, address bit [0] is ignored by the hardware since all fetches must be at least halfword aligned. For MASK != 0 and regardless of watch type, address bits [x-1:0] are ignored in the address comparison. Using a mask means the comparator matches on a range of addresses, defined by the unmasked most significant bits of the address, bits [31:x]. The maximum MASK value is 24, producing a 16 Mbyte mask. An attempted write of a MASK value > 24 is limited by the MTBDWT hardware to 24. If MTBDWT_COMP0 is used as a data value comparator, then MTBDWT_MASK0 should be programmed to zero.

19.3.2.4 MTB_DWT Comparator Function Register 0 (MTB_DWT_FCT0)

The MTBDWT_FCTn registers control the operation of comparator n.

Address: F000_1000h base + 28h offset = F000_1028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MATCHED	0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATAVADDR0				DATAVSIZE		0	DATAVMATCH	0				FUNCTION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_FCT0 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 MATCHED	Comparator match If this read-only flag is asserted, it indicates the operation defined by the FUNCTION field occurred since the last read of the register. Reading the register clears this bit. 0 No match. 1 Match occurred.

Table continues on the next page...

MTB_DWT_FCT0 field descriptions (continued)

Field	Description
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 DATAVADDR0	Data Value Address 0 Since the MTB_DWT implements two comparators, the DATAVADDR0 field is restricted to values {0,1}. When the DATAVMATCH bit is asserted, this field defines the comparator number to use for linked address comparison. If MTBDWT_COMP0 is used as a data watchpoint and MTBDWT_COMP1 as an address watchpoint, DATAVADDR0 must be set.
11–10 DATAVSIZE	Data Value Size For data value matching, this field defines the size of the required data comparison. 00 Byte. 01 Halfword. 10 Word. 11 Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DATAVMATCH	Data Value Match When this field is 1, it enables data value comparison. For this implementation, MTBDWT_COMP0 supports address or data value comparisons; MTBDWT_COMP1 only supports address comparisons. 0 Perform address comparison. 1 Perform data value comparison.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FUNCTION	Function Selects the action taken on a comparator match. If MTBDWT_COMP0 is used for a data value and MTBDWT_COMP1 for an address value, then MTBDWT_FCT1[FUNCTION] must be set to zero. For this configuration, MTBDWT_MASK1 can be set to a non-zero value, so the combined comparators match on a range of addresses. 0000 Disabled. 0100 Instruction fetch. 0101 Data operand read. 0110 Data operand write. 0111 Data operand (read + write). others Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.

19.3.2.5 MTB_DWT Comparator Function Register 1 (MTB_DWT_FCT1)

The MTBDWT_FCTn registers control the operation of comparator n. Since the MTB_DWT only supports data value comparisons on comparator 0, there are several fields in the MTBDWT_FCT1 register that are RAZ/WI (bits 12, 11:10, 8).

Address: F000_1000h base + 38h offset = F000_1038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MATCHED	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FUNCTION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_FCT1 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 MATCHED	Comparator match If this read-only flag is asserted, it indicates the operation defined by the FUNCTION field occurred since the last read of the register. Reading the register clears this bit.

Table continues on the next page...

MTB_DWT_FCT1 field descriptions (continued)

Field	Description
	0 No match. 1 Match occurred.
23–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FUNCTION	Function Selects the action taken on a comparator match. If MTBDWT_COMP0 is used for a data value and MTBDWT_COMP1 for an address value, then MTBDWT_FCT1[FUNCTION] must be set to zero. For this configuration, MTBDWT_MASK1 can be set to a non-zero value, so the combined comparators match on a range of addresses. 0000 Disabled. 0100 Instruction fetch. 0101 Data operand read. 0110 Data operand write. 0111 Data operand (read + write). others Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.

19.3.2.6 MTB_DWT Trace Buffer Control Register (MTB_DWT_TBCTRL)

The MTBDWT_TBCTRL register defines how the watchpoint comparisons control the actual trace buffer operation.

Recall the MTB supports starting and stopping the program trace based on the watchpoint comparisons signaled via TSTART and TSTOP. The watchpoint comparison signals are enabled in the MTB's control logic by setting the appropriate enable bits, MTB_MASTER[TSTARTEN, TSTOPEN]. In the event of simultaneous assertion of both TSTART and TSTOP, TSTART takes priority.

Address: F000_1000h base + 200h offset = F000_1200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NUMCOMP				0											
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														ACOMP1	ACOMP0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_TBCTRL field descriptions

Field	Description
31–28 NUMCOMP	Number of Comparators This read-only field specifies the number of comparators in the MTB_DWT. This implementation includes two registers.
27–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ACOMP1	Action based on Comparator 1 match When the MTBDWT_FCT1[MATCHED] is set, it indicates MTBDWT_COMP1 address compare has triggered and the trace buffer's recording state is changed. 0 Trigger TSTOP based on the assertion of MTBDWT_FCT1[MATCHED]. 1 Trigger TSTART based on the assertion of MTBDWT_FCT1[MATCHED].
0 ACOMP0	Action based on Comparator 0 match When the MTBDWT_FCT0[MATCHED] is set, it indicates MTBDWT_COMP0 address compare has triggered and the trace buffer's recording state is changed. The assertion of MTBDWT_FCT0[MATCHED] is caused by the following conditions: • Address match in MTBDWT_COMP0 when MTBDWT_FCT0[DATAVMATCH] = 0 • Data match in MTBDWT_COMP0 when MTBDWT_FCT0[DATAVMATCH, DATAVADDR0] = {1,0} • Data match in MTBDWT_COMP0 and address match in MTBDWT_COMP1 when MTBDWT_FCT0[DATAVMATCH, DATAVADDR0] = {1,1} 0 Trigger TSTOP based on the assertion of MTBDWT_FCT0[MATCHED]. 1 Trigger TSTART based on the assertion of MTBDWT_FCT0[MATCHED].

19.3.2.7 Device Configuration Register (MTB_DWT_DEVICECFG)

This register indicates the device configuration. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FC8h offset = F000_1FC8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICECFG																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_DWT_DEVICECFG field descriptions

Field	Description
DEVICECFG	DEVICECFG Hardwired to 0x0000_0000.

19.3.2.8 Device Type Identifier Register (MTB_DWT_DEVICETYPID)

This register indicates the device type ID. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FCCh offset = F000_1FCCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICETYPID																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

MTB_DWT_DEVICETYPID field descriptions

Field	Description
DEVICETYPID	DEVICETYPID Hardwired to 0x0000_0004.

19.3.2.9 Peripheral ID Register (MTB_DWT_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FD0h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PERIPHID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

MTB_DWT_PERIPHID*n* field descriptions

Field	Description
PERIPHID	PERIPHID Peripheral ID1 is hardwired to 0x0000_00E0; ID2 to 0x0000_0008; and all the others to 0x0000_0000.

19.3.2.10 Component ID Register (MTB_DWT_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FF0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COMPID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

MTB_DWT_COMPID*n* field descriptions

Field	Description
COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0090; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

19.3.3 System ROM Memory Map

The System ROM Table registers are also mapped into a sparsely-populated 4 KB address space.

For core configurations like that supported by Cortex-M0+, ARM recommends that a debugger identifies and connects to the debug components using the CoreSight debug infrastructure.

ARM recommends that a debugger follows the flow as shown in the following figure to discover the components in the CoreSight debug infrastructure. In this case, a debugger reads the peripheral and component ID registers for each CoreSight component in the CoreSight system.

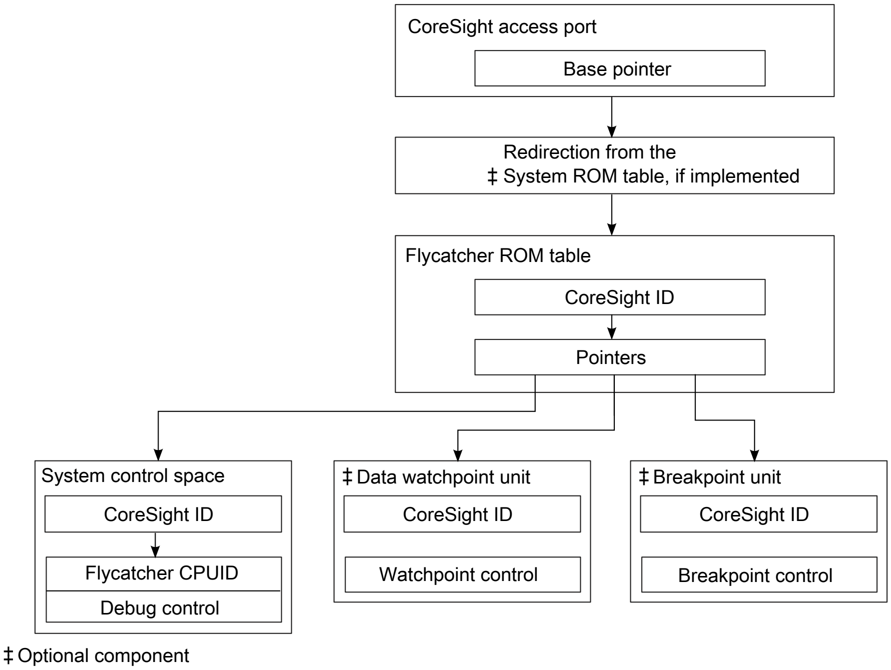


Figure 19-3. CoreSight discovery process

ROM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_2000	Entry (ROM_ENTRY0)	32	R	See section	19.3.3.1/ 357
F000_2004	Entry (ROM_ENTRY1)	32	R	See section	19.3.3.1/ 357
F000_2008	Entry (ROM_ENTRY2)	32	R	See section	19.3.3.1/ 357
F000_200C	End of Table Marker Register (ROM_TABLEMARK)	32	R	0000_0000h	19.3.3.2/ 358
F000_2FCC	System Access Register (ROM_SYSACCESS)	32	R	0000_0001h	19.3.3.3/ 358
F000_2FD0	Peripheral ID Register (ROM_PERIPHID4)	32	R	See section	19.3.3.4/ 359

Table continues on the next page...

ROM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_2FD4	Peripheral ID Register (ROM_PERIPHID5)	32	R	See section	19.3.3.4/ 359
F000_2FD8	Peripheral ID Register (ROM_PERIPHID6)	32	R	See section	19.3.3.4/ 359
F000_2FDC	Peripheral ID Register (ROM_PERIPHID7)	32	R	See section	19.3.3.4/ 359
F000_2FE0	Peripheral ID Register (ROM_PERIPHID0)	32	R	See section	19.3.3.4/ 359
F000_2FE4	Peripheral ID Register (ROM_PERIPHID1)	32	R	See section	19.3.3.4/ 359
F000_2FE8	Peripheral ID Register (ROM_PERIPHID2)	32	R	See section	19.3.3.4/ 359
F000_2FEC	Peripheral ID Register (ROM_PERIPHID3)	32	R	See section	19.3.3.4/ 359
F000_2FF0	Component ID Register (ROM_COMPID0)	32	R	See section	19.3.3.5/ 359
F000_2FF4	Component ID Register (ROM_COMPID1)	32	R	See section	19.3.3.5/ 359
F000_2FF8	Component ID Register (ROM_COMPID2)	32	R	See section	19.3.3.5/ 359
F000_2FFC	Component ID Register (ROM_COMPID3)	32	R	See section	19.3.3.5/ 359

19.3.3.1 Entry (ROM_ENTRY*n*)

The System ROM Table begins with "n" relative 32-bit addresses, one for each debug component present in the device and terminating with an all-zero value signaling the end of the table at the "n+1"-th value.

It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + 0h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENTRY																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

ROM_ENTRY*n* field descriptions

Field	Description
ENTRY	ENTRY

ROM_ENTRY*n* field descriptions (continued)

Field	Description
	Entry 0 (MTB) is hardwired to 0xFFFF_E003; Entry 1 (MTBDWT) to 0xFFFF_F003; Entry 2 (CM0+ ROM Table) to 0xF00F_D003.

19.3.3.2 End of Table Marker Register (ROM_TABLEMARK)

This register indicates end of table marker. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + Ch offset = F000_200Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MARK																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROM_TABLEMARK field descriptions

Field	Description
MARK	MARK Hardwired to 0x0000_0000

19.3.3.3 System Access Register (ROM_SYSACCESS)

This register indicates system access. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FCCh offset = F000_2FCCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYSACCESS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ROM_SYSACCESS field descriptions

Field	Description
SYSACCESS	SYSACCESS Hardwired to 0x0000_0001

19.3.3.4 Peripheral ID Register (ROM_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FD0h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PERIPHID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

ROM_PERIPHID*n* field descriptions

Field	Description
PERIPHID	PERIPHID Peripheral ID1 is hardwired to 0x0000_00E0; ID2 to 0x0000_0008; and all the others to 0x0000_0000.

19.3.3.5 Component ID Register (ROM_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FF0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COMPID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

ROM_COMPID*n* field descriptions

Field	Description
COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0010; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

Chapter 20
Crossbar Switch Lite (AXBS-Lite)

This chapter provides information on the layout, configuration, and programming of the crossbar switch. The crossbar switch connects bus masters and bus slaves using a crossbar switch structure.

20.1 Introduction

The information found here provides information on the layout, configuration, and programming of the crossbar switch.

The crossbar switch connects bus masters and bus slaves using a crossbar switch structure. This structure allows up to four bus masters to access different bus slaves simultaneously, while providing arbitration among the bus masters when they access the same slave.

20.1.1 Features

The crossbar switch includes these features:

- Symmetric crossbar bus switch implementation
    - Allows concurrent accesses from different masters to different slaves
- Up to single-clock 32-bit transfer
- Programmable configuration for fixed-priority or round-robin slave port arbitration (see the chip-specific information).

20.2 Memory Map / Register Definition

This crossbar switch is designed for minimal gate count. It, therefore, has no memory-mapped configuration registers.

Please see the chip-specific information for information on whether the arbitration method in the crossbar switch is programmable, and by which module.

20.3 Functional Description

20.3.1 General operation

When a master accesses the crossbar switch, the access is immediately taken. If the targeted slave port of the access is available, then the access is immediately presented on the slave port. Single-clock or zero-wait-state accesses are possible through the crossbar. If the targeted slave port of the access is busy or parked on a different master port, the requesting master simply sees wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request depends on each master's priority level and the responding slave's access time.

Because the crossbar switch appears to be just another slave to the master device, the master device has no knowledge of whether it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting, it simply waits.

A master is given control of the targeted slave port only after a previous access to a different slave port completes, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when:

- A higher priority master has:
  - An outstanding request to one slave port that has a long response time and
  - A pending access to a different slave port, and
- A lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

After the master has control of the slave port it is targeting, the master remains in control of the slave port until it relinquishes the slave port by running an IDLE cycle or by targeting a different slave port for its next access.

The master can also lose control of the slave port if another higher-priority master makes a request to the slave port.

The crossbar terminates all master IDLE transfers, as opposed to allowing the termination to come from one of the slave buses. Additionally, when no master is requesting access to a slave port, the crossbar drives IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port.

When a slave bus is being idled by the crossbar, it remains parked with the last master to use the slave port. This is done to save the initial clock of arbitration delay that otherwise would be seen if the same master had to arbitrate to gain control of the slave port.

20.3.2 Arbitration

The crossbar switch supports two arbitration algorithms:

- Fixed priority
- Round-robin

The selection of the global slave port arbitration algorithm is described in the crossbar switch chip-specific information.

20.3.2.1 Arbitration during undefined length bursts

All lengths of burst accesses lock out arbitration until the last beat of the burst.

20.3.2.2 Fixed-priority operation

When operating in fixed-priority mode, each master is assigned a unique priority level with the highest numbered master having the highest priority (for example, in a system with 5 masters, master 1 has lower priority than master 3). If two masters request access to the same slave port, the master with the highest priority gains control over the slave port.

NOTE

> In this arbitration mode, a higher-priority master can monopolize a slave port, preventing accesses from any lower-priority master to the port.

When a master makes a request to a slave port, the slave port checks whether the new requesting master's priority level is higher than that of the master that currently has control over the slave port, unless the slave port is in a parked state. The slave port performs an arbitration check at every clock edge to ensure that the proper master, if any, has control of the slave port.

The following table describes possible scenarios based on the requesting master port:

Table 20-1. How the Crossbar Switch grants control of a slave port to a master

When	Then the Crossbar Switch grants control to the requesting master
Both of the following are true: • The current master is not running a transfer. • The new requesting master's priority level is higher than that of the current master.	At the next clock edge
The requesting master's priority level is lower than the current master.	At the conclusion of one of the following cycles: • An IDLE cycle • A non-IDLE cycle to a location other than the current slave port

20.3.2.3 Round-robin priority operation

When operating in round-robin mode, each master is assigned a relative priority based on the master port number. This relative priority is compared to the master port number (ID) of the last master to perform a transfer on the slave bus. The highest priority requesting master becomes owner of the slave bus at the next transfer boundary. Priority is based on how far ahead the ID of the requesting master is to the ID of the last master.

After granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line is granted access to the slave port at the next transfer boundary, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the crossbar is implemented with master ports 0, 1, 4, and 5. If the last master of the slave port was master 1, and master 0, 4, and 5 make simultaneous requests, they are serviced in the order: 4 then 5 then 0.

The round-robin arbitration mode generally provides a more fair allocation of the available slave-port bandwidth (compared to fixed priority) as the fixed master priority does not affect the master selection.

20.4 Initialization/application information

No initialization is required for the crossbar switch.

See the AXBS section of the configuration chapter for the reset state of the arbitration scheme.

Chapter 21
Peripheral Bridge (AIPS-Lite)

The Peripheral Bridge (AIPS-Lite) converts the crossbar switch interface to an interface to access a majority of peripherals on the device. The peripheral bridge supports up to 128 peripherals, including separate clock enable inputs for each of the slots to accommodate slower peripherals.

21.1 Introduction

The peripheral bridge converts the crossbar switch interface to an interface that can access most of the slave peripherals on this chip.

The peripheral bridge occupies 64 MB of the address space, which is divided into peripheral slots of 4 KB. (It might be possible that all the peripheral slots are not used. See the memory map chapter for details on slot assignments.) The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

21.1.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width

21.1.2 General operation

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge.

The register maps of the peripherals are located on 4-KB boundaries. Each peripheral is allocated one or more 4-KB block(s) of the memory map.

21.2 Memory map/register definition

The AIPS module(s) on this device do(es) not contain any user-programmable registers.

AIPS memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Master Privilege Register A (AIPS_MPRA)	32	R/W	See section	21.2.1/366
20	Peripheral Access Control Register (AIPS_PACRA)	32	R/W	See section	21.2.2/368
24	Peripheral Access Control Register (AIPS_PACRB)	32	R/W	See section	21.2.2/368
28	Peripheral Access Control Register (AIPS_PACRC)	32	R/W	See section	21.2.2/368
2C	Peripheral Access Control Register (AIPS_PACRD)	32	R/W	See section	21.2.2/368
40	Peripheral Access Control Register (AIPS_PACRE)	32	R/W	See section	21.2.3/374
44	Peripheral Access Control Register (AIPS_PACRF)	32	R/W	See section	21.2.3/374
48	Peripheral Access Control Register (AIPS_PACRG)	32	R/W	See section	21.2.3/374
4C	Peripheral Access Control Register (AIPS_PACRH)	32	R/W	See section	21.2.3/374
50	Peripheral Access Control Register (AIPS_PACRI)	32	R/W	See section	21.2.3/374
54	Peripheral Access Control Register (AIPS_PACRJ)	32	R/W	See section	21.2.3/374
58	Peripheral Access Control Register (AIPS_PACRK)	32	R/W	See section	21.2.3/374
5C	Peripheral Access Control Register (AIPS_PACRL)	32	R/W	See section	21.2.3/374
60	Peripheral Access Control Register (AIPS_PACRM)	32	R/W	See section	21.2.3/374
64	Peripheral Access Control Register (AIPS_PACRN)	32	R/W	See section	21.2.3/374
68	Peripheral Access Control Register (AIPS_PACRO)	32	R/W	See section	21.2.3/374
6C	Peripheral Access Control Register (AIPS_PACRP)	32	R/W	See section	21.2.3/374

21.2.1 Master Privilege Register A (AIPS_MPRA)

The MPRA specifies identical 4-bit fields defining the access-privilege level associated with a bus master to various peripherals on the chip. The register provides one field per bus master.

NOTE

At reset, the default value loaded into the MPRA fields is chip-specific. See the chip configuration details for the value of a particular device.

A register field that maps to an unimplemented master or peripheral behaves as read-only-zero.

Each master is assigned a logical ID from 0 to 15. See the master logical ID assignment table in the chip-specific AIPS information.

Address: 0h base + 0h offset = 0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MTR0	MTW0	MPL0	0	MTR2	MTW2	MPL2	0	MTR3	MTW3	MPL3				
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- The reset value is chip-dependent and can be found in the chip-specific AIPS information.

AIPS_MPRA field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 MTR0	Master 0 Trusted For Read Determines whether the master is trusted for read accesses. 0 This master is not trusted for read accesses. 1 This master is trusted for read accesses.
29 MTW0	Master 0 Trusted For Writes Determines whether the master is trusted for write accesses. 0 This master is not trusted for write accesses. 1 This master is trusted for write accesses.
28 MPL0	Master 0 Privilege Level Specifies how the privilege level of the master is determined. 0 Accesses from this master are forced to user-mode. 1 Accesses from this master are not forced to user-mode.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 MTR2	Master 2 Trusted For Read Determines whether the master is trusted for read accesses.

Table continues on the next page...

AIPS_MPRA field descriptions (continued)

Field	Description
	0 This master is not trusted for read accesses. 1 This master is trusted for read accesses.
21 MTW2	Master 2 Trusted For Writes Determines whether the master is trusted for write accesses. 0 This master is not trusted for write accesses. 1 This master is trusted for write accesses.
20 MPL2	Master 2 Privilege Level Specifies how the privilege level of the master is determined. 0 Accesses from this master are forced to user-mode. 1 Accesses from this master are not forced to user-mode.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 MTR3	Master 3 Trusted For Read Determines whether the master is trusted for read accesses. 0 This master is not trusted for read accesses. 1 This master is trusted for read accesses.
17 MTW3	Master 3 Trusted For Writes Determines whether the master is trusted for write accesses. 0 This master is not trusted for write accesses. 1 This master is trusted for write accesses.
16 MPL3	Master 3 Privilege Level Specifies how the privilege level of the master is determined. 0 Accesses from this master are forced to user-mode. 1 Accesses from this master are not forced to user-mode.

21.2.2 Peripheral Access Control Register (AIPS_PACR*n*)

Each PACR register consists of eight 4-bit PACR fields. Each PACR field defines the access levels for a particular peripheral. The mapping between a peripheral and its PACR field is shown in the table below. The peripheral assignment to each PACR is defined by the memory map slot that the peripheral is assigned to. See this chip's memory map for the assignment of a particular peripheral.

The following table shows the location of each peripheral slot's PACR field in the PACR registers.

Offset	Register	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0x20	PACRA	PACR0	PACR1	PACR2	PACR3	PACR4	PACR5	PACR6	PACR7
0x24	PACRB	PACR8	PACR9	PACR10	PACR11	PACR12	PACR13	PACR14	PACR15
0x28	PACRC	PACR16	PACR17	PACR18	PACR19	PACR20	PACR21	PACR22	PACR23
0x2C	PACRD	PACR24	PACR25	PACR26	PACR27	PACR28	PACR29	PACR30	PACR31
0x30	Reserved								
0x34	Reserved								
0x38	Reserved								
0x3C	Reserved								
0x40	PACRE	PACR32	PACR33	PACR34	PACR35	PACR36	PACR37	PACR38	PACR39
0x44	PACRF	PACR40	PACR41	PACR42	PACR43	PACR44	PACR45	PACR46	PACR47
0x48	PACRG	PACR48	PACR49	PACR50	PACR51	PACR52	PACR53	PACR54	PACR55
0x4C	PACRH	PACR56	PACR57	PACR58	PACR59	PACR60	PACR61	PACR62	PACR63
0x50	PACRI	PACR64	PACR65	PACR66	PACR67	PACR68	PACR69	PACR70	PACR71
0x54	PACRJ	PACR72	PACR73	PACR74	PACR75	PACR76	PACR77	PACR78	PACR79
0x58	PACRK	PACR80	PACR81	PACR82	PACR83	PACR84	PACR85	PACR86	PACR87
0x5C	PACRL	PACR88	PACR89	PACR90	PACR91	PACR92	PACR93	PACR94	PACR95
0x60	PACRM	PACR96	PACR97	PACR98	PACR99	PACR100	PACR101	PACR102	PACR103
0x64	PACRN	PACR104	PACR105	PACR106	PACR107	PACR108	PACR109	PACR110	PACR111
0x68	PACRO	PACR112	PACR113	PACR114	PACR115	PACR116	PACR117	PACR118	PACR119
0x6C	PACRP	PACR120	PACR121	PACR122	PACR123	PACR124	PACR125	PACR126	PACR127

NOTE

The register field descriptions for PACR A-D, which control peripheral slots 0-31, are shown below. The following section, Peripheral Access Control Register (AIPS_PACR*n*), shows the register field descriptions for PACR E-P. All PACR registers are identical. They are divided into two sections because they occupy two non-contiguous address spaces.

Address: 0h base + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- The reset value is chip-dependent and can be found in the AIPS chip-specific information.

AIPS_PACR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 SP0	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect Determines whether the peripheral allows write accesss. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
28 TP0	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SP1	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
25 WP1	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
24 TP1	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 SP2	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect Determines whether the peripheral allows write accesss. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
20 TP2	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SP3	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
16 TP3	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 SP4	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect Determines whether the peripheral allows write accesss. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
12 TP4	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 SP5	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
8 TP5	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 SP6	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
4 TP6	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SP7	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
0 TP7	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

21.2.3 Peripheral Access Control Register (AIPS_PACR*n*)

This section describes PACR registers E-P, which control peripheral slots 32-127. See Peripheral Access Control Register (AIPS_PACR*n*) for the description of these registers.

Address: 0h base + 40h offset + (4d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- The reset value is chip-dependent and can be found in the AIPS chip-specific information.

AIPS_PACR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 SP0	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
28 TP0	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SP1	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
25 WP1	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
24 TP1	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 SP2	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
20 TP2	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SP3	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect Determines whether the peripheral allows write accesss. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
16 TP3	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 SP4	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
12 TP4	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 SP5	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
8 TP5	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 SP6	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
4 TP6	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPS_PACR*n* field descriptions (continued)

Field	Description
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SP7	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
0 TP7	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

21.3 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

21.3.1 Access support

All combinations of access size and peripheral data port width are supported. An access that is larger than the target peripheral's data width will be decomposed to multiple, smaller accesses. Bus decomposition is terminated by a transfer error caused by an access to an empty register area.

Chapter 22
Direct Memory Access Multiplexer (DMAMUX)

DMA MUX provides flexibility in the system's use of the available DMA channels. The DMA MUX routes up to 63 DMA sources, which are called slots to be mapped to any of the 4 DMA channels. Functionally, the DMA MUX channels may be divided into two classes: Channels that implement the normal routing functionality plus periodic triggering capability, and channels that implement only the normal routing functionality.

22.1 Introduction

22.1.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the four DMA channels. This process is illustrated in the following figure.

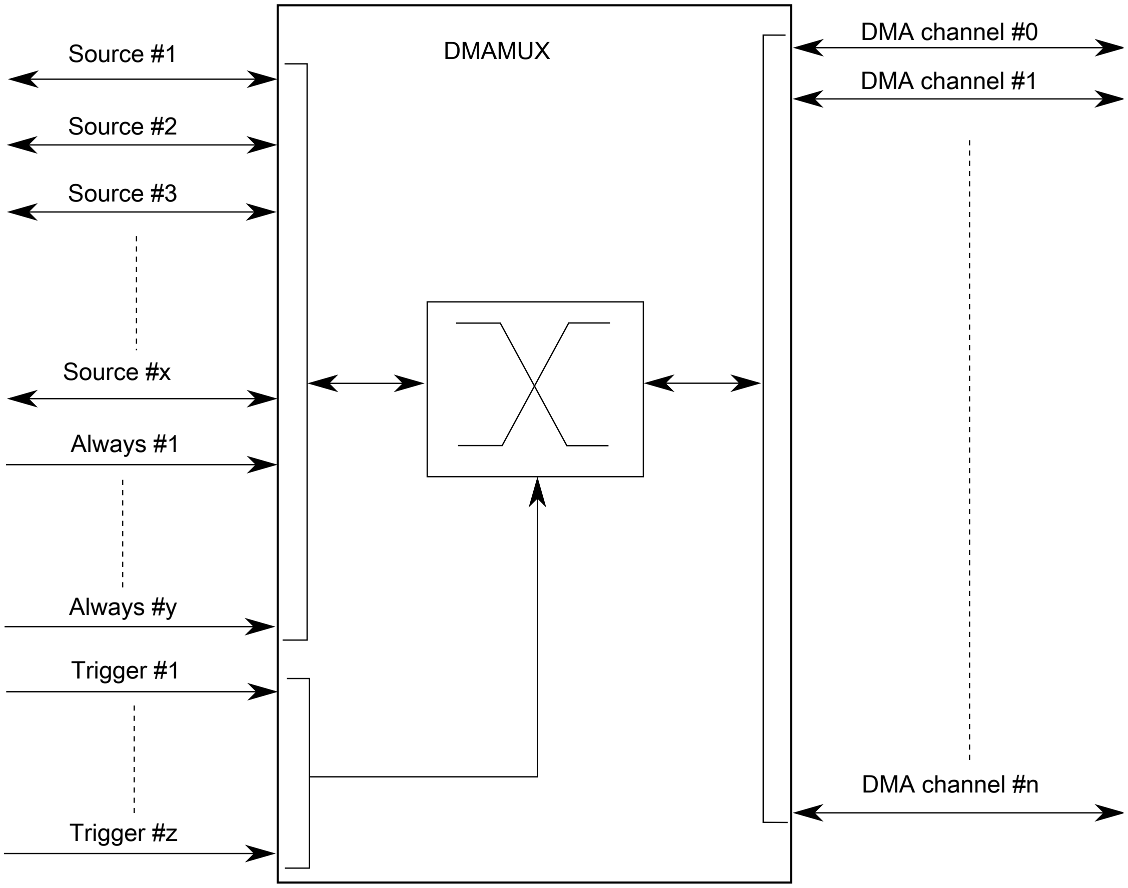


Figure 22-1. DMAMUX block diagram

22.1.2 Features

The DMAMUX module provides these features:

- Up to 63 peripheral slots and up to four always-on slots can be routed to four channels.
- four independently selectable DMA channel routers.
  - The first two channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots.

22.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically.

22.2 External signal description

The DMAMUX has no external pins.

22.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

DMAMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1000	Channel Configuration register (DMAMUX0_CHCFG0)	8	R/W	00h	22.3.1/381
4002_1001	Channel Configuration register (DMAMUX0_CHCFG1)	8	R/W	00h	22.3.1/381
4002_1002	Channel Configuration register (DMAMUX0_CHCFG2)	8	R/W	00h	22.3.1/381
4002_1003	Channel Configuration register (DMAMUX0_CHCFG3)	8	R/W	00h	22.3.1/381

22.3.1 Channel Configuration register (DMAMUX*x*_CHCFG*n*)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

NOTE

Setting multiple CHCFG registers with the same source value will result in unpredictable behavior. This is true, even if a channel is disabled (ENBL==0).

Before changing the trigger or source settings, a DMA channel must be disabled via CHCFGn[ENBL].

Address: 4002_1000h base + 0h offset + (1d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read Write	ENBL	TRIG	SOURCE					
Reset	0	0	0	0	0	0	0	0

DMAMUXx_CHCFG*n* field descriptions

Field	Description
7 ENBL	DMA Channel Enable Enables the DMA channel. 0 DMA channel is disabled. This mode is primarily used during configuration of the DMAMux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 1 DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0 Triggering is disabled. If triggering is disabled and ENBL is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode) 1 Triggering is enabled. If triggering is enabled and ENBL is set, the DMAMUX is in Periodic Trigger mode.
SOURCE	DMA Channel Source (Slot) Specifies which DMA source, if any, is routed to a particular DMA channel. See the chip-specific DMAMUX information for details about the peripherals and their slot numbers.

22.4 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels.

As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

22.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 2 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention.

Note

> Because of the dynamic nature of the system (due to DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

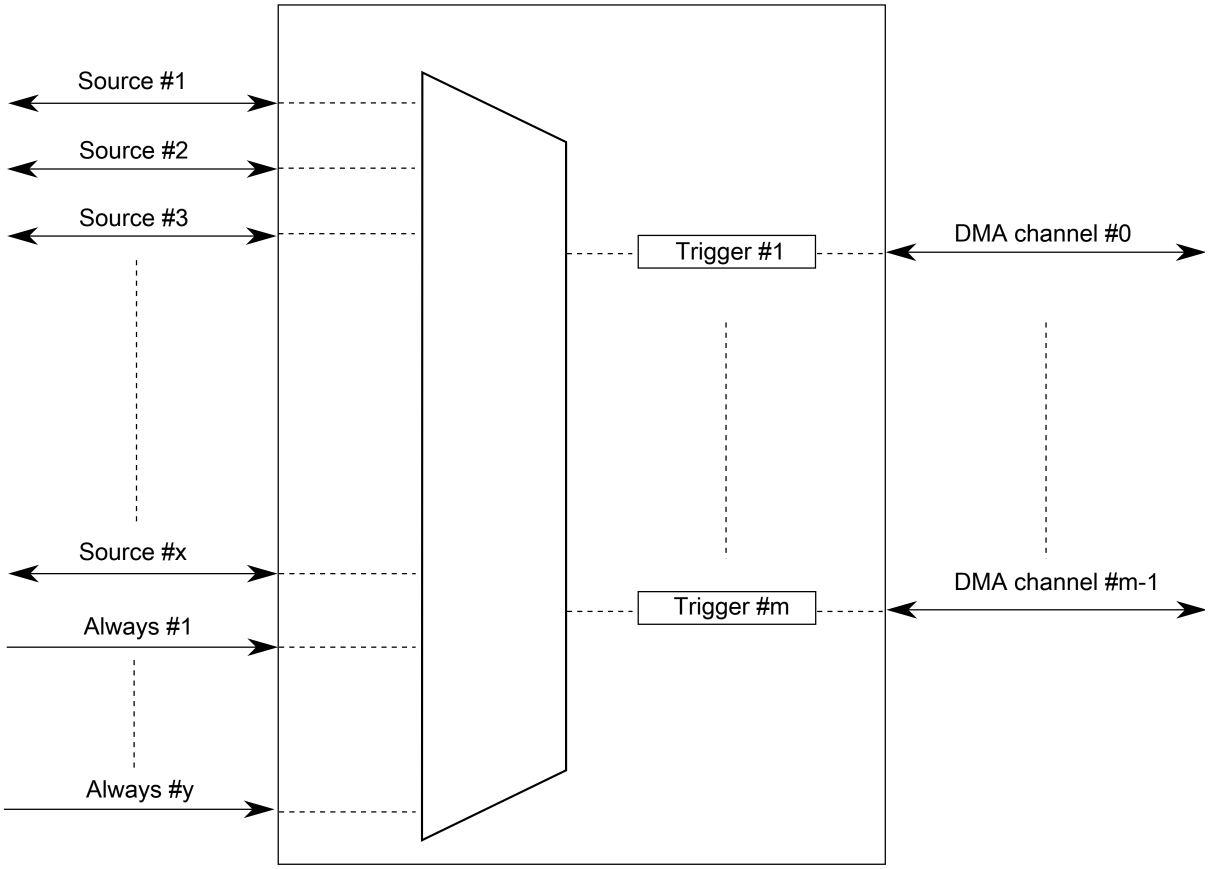


Figure 22-2. DMAMUX triggered channels

The DMA channel triggering capability allows the system to schedule regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

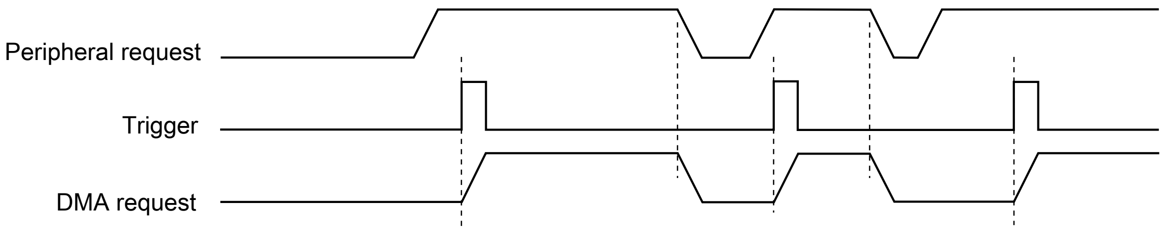


Figure 22-3. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

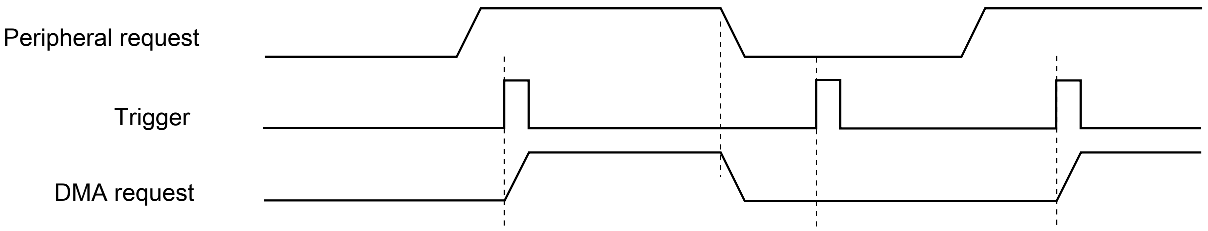


Figure 22-4. DMAMUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

  As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5 μs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

22.4.2 DMA channels with no triggering capability

The other channels of the DMAMUX provide the normal routing functionality as described in Modes of operation.

22.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are four additional DMA sources that are always enabled. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the sources that are always enabled provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use explicit software reactivation.

  In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use an always-enabled DMA source.

  In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

22.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

22.5.1 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

22.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 2 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.

3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

> The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1.
2. Configure channel 1 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC5 to CHCFG1.

The following code example illustrates steps 1 and 4 above:

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

To enable a source, without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 2 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is set while CHCFG[TRIG] is cleared.

NOTE

> The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with no periodic triggering capability:

1. Write 0x00 to CHCFG1.

2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG1.

The following code example illustrates steps 1 and 3 above:

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8.
3. Write 0x87 to CHCFG8. (In this example, setting CHCFG[TRIG] would have no effect due to the assumption that channel 8 does not support the periodic triggering functionality.)

The following code example illustrates steps 2 and 3 above:

Chapter 23
DMA Controller Module

DMA Controller Module describes in detail its signals and programming model. It also explains operations, features, and supported data transfer modes.

23.1 Introduction

Information found here describes the direct memory access (DMA) controller module. It provides an overview of the module and describes in detail its signals and programming model.

The latter sections of this chapter describe operations, features, and supported data transfer modes in detail.

An example of using several features of the DMA module is described in AN4631: Using the Asynchronous DMA features of the Kinetis L Series.

Note

> The designation *n* is used throughout this section to refer to registers or signals associated with one of the four identical DMA channels: DMA0, DMA1, DMA2, or DMA3.

23.1.1 Overview

The DMA controller module enables fast transfers of data, providing an efficient way to move blocks of data with minimal processor interaction. The DMA module, shown in the following figure, has four channels that allow 8-bit, 16-bit, or 32-bit data transfers. Each channel has a dedicated Source Address register (SAR*n*), Destination Address register (DAR*n*), Status register (DSR*n*), Byte Count register (BCR*n*), and Control register (DCR*n*). Collectively, the combined program-visible registers associated with each channel define a transfer control descriptor (TCD). All transfers are dual address, moving data from a source memory location to a destination memory location with the module operating as a 32-bit bus master connected to the system bus. The programming model is accessed through a 32-bit connection with the slave peripheral bus. DMA data transfers may be explicitly initiated by software or by peripheral hardware requests.

The following figure is a simplified block diagram of the 4-channel DMA controller.

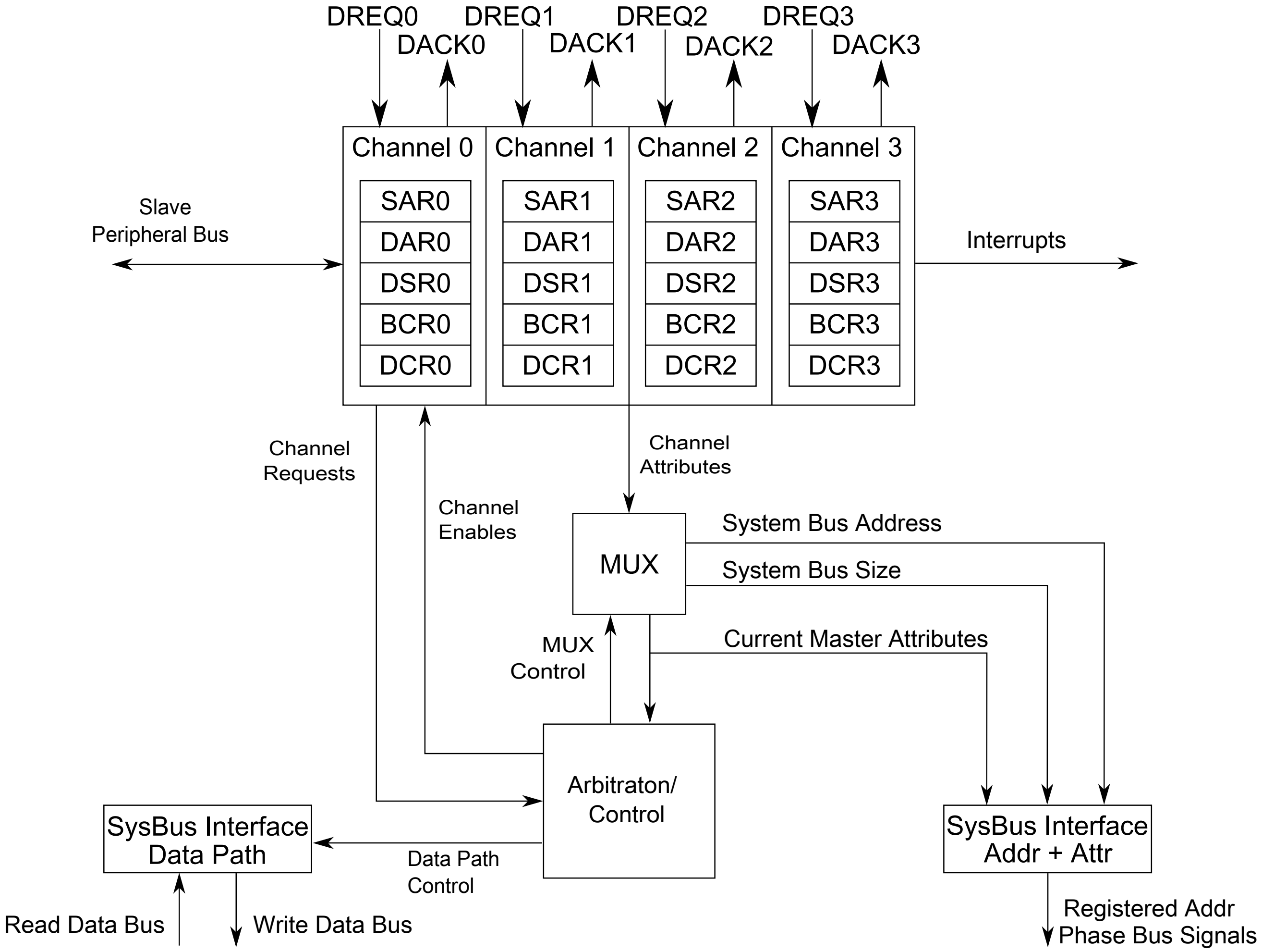


Figure 23-1. 4-Channel DMA Block Diagram

The terms *peripheral request* and *DREQ* refer to a DMA request from one of the on-chip peripherals or package pins. The DMA provides hardware handshake signals: either a DMA acknowledge (DACK) or a done indicator back to the peripheral.

23.1.2 Features

The DMA controller module features:

- Four independently programmable DMA controller channels

- Dual-address transfers via 32-bit master connection to the system bus
- Data transfers in 8-, 16-, or 32-bit blocks
- Continuous-mode or cycle-steal transfers from software or peripheral initiation
- Automatic hardware acknowledge/done indicator from each channel
- Independent source and destination address registers
- Optional modulo addressing and automatic updates of source and destination addresses
- Independent transfer sizes for source and destination
- Optional auto-alignment feature for source or destination accesses
- Optional automatic single or double channel linking
- Programming model accessed via 32-bit slave peripheral bus
- Channel arbitration on transfer boundaries using fixed priority scheme

23.2 DMA Transfer Overview

The DMA module can move data within system memory (including memory and peripheral devices) with minimal processor intervention, greatly improving overall system performance.

The DMA module consists of four independent, functionally equivalent channels, so references to DMA in this chapter apply to any of the channels. It is not possible to address all four channels at once.

As soon as a channel has been initialized, it may be started by setting DCR*n*[START] or a properly-selected peripheral DMA request, depending on the status of DCR*n*[ERQ].

The DMA controller supports dual-address transfers using its bus master connection to the system bus. The DMA channels support transfers up to 32 data bits in size and have the same memory map addressibility as the processor.

- Dual-address transfers—A dual-address transfer consists of a read followed by a write and is initiated by a request using the DCRn[START] bit or by a peripheral DMA request. The read data is temporarily held in the DMA channel hardware until the write operation. Two types of single transfers occur: a read from a source address followed by a write to a destination address. See the following figure.

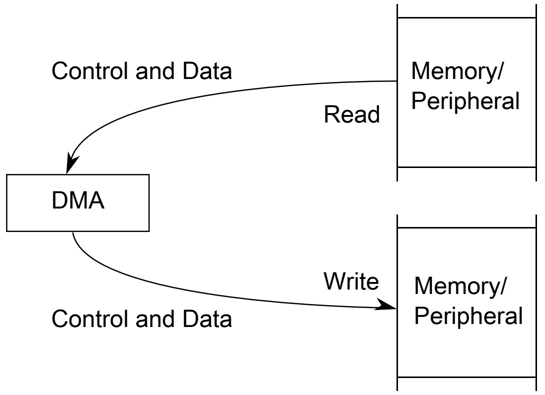


Figure 23-2. Dual-Address Transfer

Any operation involving a DMA channel follows the same three steps:

1. Channel initialization—The transfer control descriptor, contained in the channel registers, is loaded with address pointers, a byte-transfer count, and control information using accesses from the slave peripheral bus.
2. Data transfer—The DMA accepts requests for data transfers. Upon receipt of a request, it provides address and bus control for the transfers via its master connection to the system bus and temporary storage for the read data. The channel performs one or more source read and destination write data transfers.
3. Channel termination—Occurs after the operation is finished successfully or due to an error. The channel indicates the operation status in the channel's DSR, described in the definitions of the DMA Status Registers (DSRn) and Byte Count Registers (BCRn).

23.3 Memory Map/Register Definition

Information about the registers related to the DMA controller module can be found here.

Descriptions of each register and its bit assignments follow. Modifying DMA control registers during a transfer can result in undefined operation. The following table shows the mapping of DMA controller registers. The DMA programming model is accessed via the slave peripheral bus. The concatenation of the source and destination address registers, the status and byte count register, and the control register create a 128-bit transfer control descriptor (TCD) that defines the operation of each DMA channel.

DMA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_8100	Source Address Register (DMA_SAR0)	32	R/W	0000_0000h	23.3.1/393
4000_8104	Destination Address Register (DMA_DAR0)	32	R/W	0000_0000h	23.3.2/394
4000_8108	DMA Status Register / Byte Count Register (DMA_DSR_BCR0)	32	R/W	0000_0000h	23.3.3/395
4000_810C	DMA Control Register (DMA_DCR0)	32	R/W	0000_0000h	23.3.4/397
4000_8110	Source Address Register (DMA_SAR1)	32	R/W	0000_0000h	23.3.1/393
4000_8114	Destination Address Register (DMA_DAR1)	32	R/W	0000_0000h	23.3.2/394
4000_8118	DMA Status Register / Byte Count Register (DMA_DSR_BCR1)	32	R/W	0000_0000h	23.3.3/395
4000_811C	DMA Control Register (DMA_DCR1)	32	R/W	0000_0000h	23.3.4/397
4000_8120	Source Address Register (DMA_SAR2)	32	R/W	0000_0000h	23.3.1/393
4000_8124	Destination Address Register (DMA_DAR2)	32	R/W	0000_0000h	23.3.2/394
4000_8128	DMA Status Register / Byte Count Register (DMA_DSR_BCR2)	32	R/W	0000_0000h	23.3.3/395
4000_812C	DMA Control Register (DMA_DCR2)	32	R/W	0000_0000h	23.3.4/397
4000_8130	Source Address Register (DMA_SAR3)	32	R/W	0000_0000h	23.3.1/393
4000_8134	Destination Address Register (DMA_DAR3)	32	R/W	0000_0000h	23.3.2/394
4000_8138	DMA Status Register / Byte Count Register (DMA_DSR_BCR3)	32	R/W	0000_0000h	23.3.3/395
4000_813C	DMA Control Register (DMA_DCR3)	32	R/W	0000_0000h	23.3.4/397

23.3.1 Source Address Register (DMA_SAR*n*)

Restriction

For this register:

- Only 32-bit writes are allowed. 16-bit and 8-bit writes result in a bus error.
- Only several values are allowed to be written to bits 31-20 of this register, see the value list in the field description. A write of any other value to these bits causes a configuration error when the channel starts to execute. For more information about the configuration error, see the description of the CE field of DSR.

Address: 4000_8000h base + 100h offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_SAR*n* field descriptions

Field	Description
SAR	SAR Each SAR contains the byte address used by the DMA controller to read data. The SARn is typically aligned on a 0-modulo-ssize boundary—that is, on the natural alignment of the source data. **Restriction:** Bits 31-20 of this register must be written with one of only several allowed values. Each of these allowed values corresponds to a valid region of the device's memory map. The allowed values are: • 0x000x_xxxx • 0x1FFx_xxxx • 0x200x_xxxx • 0x400x_xxxx After being written with one of the allowed values, bits 31-20 read back as the written value. After being written with any other value, bits 31-20 read back as an indeterminate value.

23.3.2 Destination Address Register (DMA_DAR*n*)

Restriction

For this register:

- Only 32-bit writes are allowed. 16-bit and 8-bit writes result in a bus error.
- Only several values are allowed to be written to bits 31-20 of this register, see the value list in the field description. A write of any other value to these bits causes a configuration error when the channel starts to execute. For more information about the configuration error, see the description of the CE field of DSR.

Address: 4000_8000h base + 104h offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_DAR*n* field descriptions

Field	Description
DAR	DAR Each DAR contains the byte address used by the DMA controller to write data. The DARn is typically aligned on a 0-modulo-dsize boundary—that is, on the natural alignment of the destination data. **Restriction:** Bits 31-20 of this register must be written with one of only several allowed values. Each of these allowed values corresponds to a valid region of the device's memory map. The allowed values are:

DMA_DAR*n* field descriptions (continued)

Field	Description
	• 0x000x_xxxx • 0x1FFx_xxxx • 0x200x_xxxx • 0x400x_xxxx After being written with one of the allowed values, bits 31-20 read back as the written value. After being written with any other value, bits 31-20 read back as an indeterminate value.

23.3.3 DMA Status Register / Byte Count Register (DMA_DSR_BCR*n*)

DSR and BCR are two logical registers that occupy one 32-bit address. DSRn occupies bits 31–24, and BCRn occupies bits 23–0. DSRn contains flags indicating the channel status, and BCRn contains the number of bytes yet to be transferred for a given block.

On the successful completion of the write transfer, BCRn decrements by 1, 2, or 4 for 8-bit, 16-bit, or 32-bit accesses, respectively. BCRn is cleared if a 1 is written to DSR[DONE].

In response to an event, the DMA controller writes to the appropriate DSRn bit. Only a write to DSRn[DONE] results in action. DSRn[DONE] is set when the block transfer is complete.

When a transfer sequence is initiated and BCRn[BCR] is not a multiple of 4 or 2 when the DMA is configured for 32-bit or 16-bit transfers, respectively, DSRn[CE] is set and no transfer occurs.

Address: 4000_8000h base + 108h offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CE	BES	BED	0	REQ	BSY	DONE	BCR							
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BCR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_DSR_BCR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 CE	Configuration Error Any of the following conditions causes a configuration error: • BCR, SAR, or DAR does not match the requested transfer size. • A value greater than 0F_FFFFh is written to BCR. • Bits 31-20 of SAR or DAR are written with a value other than one of the allowed values. See SAR and DAR. • SSIZE or DSIZE is set to an unsupported value. • BCR equals 0 when the DMA receives a start condition. CE is cleared at hardware reset or by writing a 1 to DONE. 0 No configuration error exists. 1 A configuration error has occurred.
29 BES	Bus Error on Source BES is cleared at hardware reset or by writing a 1 to DONE. 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the read portion of a transfer.
28 BED	Bus Error on Destination BED is cleared at hardware reset or by writing a 1 to DONE. 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the write portion of a transfer.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 REQ	Request 0 No request is pending or the channel is currently active. Cleared when the channel is selected. 1 The DMA channel has a transfer remaining and the channel is not selected.
25 BSY	Busy 0 DMA channel is inactive. Cleared when the DMA has finished the last transaction. 1 BSY is set the first time the channel is enabled after a transfer is initiated.
24 DONE	Transactions Done

Table continues on the next page...

DMA_DSR_BCR*n* field descriptions (continued)

Field	Description
	Set when all DMA controller transactions complete as determined by transfer count, or based on error conditions. When BCR reaches 0, DONE is set when the final transfer completes successfully. DONE can also be used to abort a transfer by resetting the status bits. When a transfer completes, software must clear DONE before reprogramming the DMA. 0 DMA transfer is not yet complete. Writing a 0 has no effect. 1 DMA transfer completed. Writing a 1 to this bit clears all DMA status bits and should be used in an interrupt service routine to clear the DMA interrupt and error bits.
BCR	BCR This field contains the number of bytes yet to be transferred for a given block. **Restriction:** BCR must be written with a value equal to or less than 0F_FFFFh. After being written with a value in this range, bits 23-20 of BCR read back as 0000b. A write to BCR of a value greater than 0F_FFFFh causes a configuration error when the channel starts to execute. After being written with a value in this range, bits 23-20 of BCR read back as 0001b.

23.3.4 DMA Control Register (DMA_DCR*n*)

Address: 4000_8000h base + 10Ch offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EINT	ERQ	CS	AA	0			Reserved	EADREQ	SINC	SSIZE		DINC	DSIZE		0
W																START
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SMOD				DMOD				D_REQ	0	LINKCC		LCH1		LCH2	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_DCR*n* field descriptions

Field	Description
31 EINT	Enable Interrupt on Completion of Transfer

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	Determines whether an interrupt is generated by completing a transfer or by the occurrence of an error condition. 0 No interrupt is generated. 1 Interrupt signal is enabled.
30 ERQ	Enable Peripheral Request **CAUTION:** Be careful: a collision can occur between START and D_REQ when ERQ is 1. 0 Peripheral request is ignored. 1 Enables peripheral request to initiate transfer. A software-initiated request (setting START) is always enabled.
29 CS	Cycle Steal 0 DMA continuously makes read/write transfers until the BCR decrements to 0. 1 Forces a single read/write transfer per request.
28 AA	Auto-align AA and SIZE bits determine whether the source or destination is auto-aligned; that is, transfers are optimized based on the address and size. 0 Auto-align disabled 1 If SSIZE indicates a transfer no smaller than DSIZE, source accesses are auto-aligned; otherwise, destination accesses are auto-aligned. Source alignment takes precedence over destination alignment. If auto-alignment is enabled, the appropriate address register increments, regardless of DINC or SINC.
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 Reserved	This field is reserved. **CAUTION:** Must be written as zero; otherwise, undefined behavior results.
23 EADREQ	Enable asynchronous DMA requests Enables the channel to support asynchronous DREQs while the MCU is in Stop mode. 0 Disabled 1 Enabled
22 SINC	Source Increment Controls whether the source address increments after each successful transfer. 0 No change to SAR after a successful transfer. 1 The SAR increments by 1, 2, 4 as determined by the transfer size.
21–20 SSIZE	Source Size Determines the data size of the source bus cycle for the DMA controller. 00 32-bit 01 8-bit

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	10 16-bit 11 Reserved (generates a configuration error (DSRn[CE]) if incorrectly specified at time of channel activation)
19 DINC	Destination Increment Controls whether the destination address increments after each successful transfer. 0 No change to the DAR after a successful transfer. 1 The DAR increments by 1, 2, 4 depending upon the size of the transfer.
18–17 DSIZE	Destination Size Determines the data size of the destination bus cycle for the DMA controller. 00 32-bit 01 8-bit 10 16-bit 11 Reserved (generates a configuration error (DSRn[CE]) if incorrectly specified at time of channel activation)
16 START	Start Transfer 0 DMA inactive 1 The DMA begins the transfer in accordance to the values in the TCDn. START is cleared automatically after one module clock and always reads as logic 0.
15–12 SMOD	Source Address Modulo Defines the size of the source data circular buffer used by the DMA Controller. If enabled (SMOD is non-zero), the buffer base address is located on a boundary of the buffer size. The value of this boundary is based upon the initial source address (SAR). The base address should be aligned to a 0-modulo-(circular buffer size) boundary. Misaligned buffers are not possible. The boundary is forced to the value determined by the upper address bits in the field selection. 0000 Buffer disabled 0001 Circular buffer size is 16 bytes. 0010 Circular buffer size is 32 bytes. 0011 Circular buffer size is 64 bytes. 0100 Circular buffer size is 128 bytes. 0101 Circular buffer size is 256 bytes. 0110 Circular buffer size is 512 bytes. 0111 Circular buffer size is 1 KB. 1000 Circular buffer size is 2 KB. 1001 Circular buffer size is 4 KB. 1010 Circular buffer size is 8 KB. 1011 Circular buffer size is 16 KB. 1100 Circular buffer size is 32 KB. 1101 Circular buffer size is 64 KB. 1110 Circular buffer size is 128 KB. 1111 Circular buffer size is 256 KB.
11–8 DMOD	Destination Address Modulo Defines the size of the destination data circular buffer used by the DMA Controller. If enabled (DMOD value is non-zero), the buffer base address is located on a boundary of the buffer size. The value of this

Table continues on the next page...

DMA_DCRn field descriptions (continued)

Field	Description
	boundary depends on the initial destination address (DAR). The base address should be aligned to a 0-modulo-(circular buffer size) boundary. Misaligned buffers are not possible. The boundary is forced to the value determined by the upper address bits in the field selection. 0000 Buffer disabled 0001 Circular buffer size is 16 bytes 0010 Circular buffer size is 32 bytes 0011 Circular buffer size is 64 bytes 0100 Circular buffer size is 128 bytes 0101 Circular buffer size is 256 bytes 0110 Circular buffer size is 512 bytes 0111 Circular buffer size is 1 KB 1000 Circular buffer size is 2 KB 1001 Circular buffer size is 4 KB 1010 Circular buffer size is 8 KB 1011 Circular buffer size is 16 KB 1100 Circular buffer size is 32 KB 1101 Circular buffer size is 64 KB 1110 Circular buffer size is 128 KB 1111 Circular buffer size is 256 KB
7 D_REQ	Disable Request DMA hardware automatically clears the corresponding DCRn[ERQ] bit when the byte count register reaches 0. 0 ERQ bit is not affected. 1 ERQ bit is cleared when the BCR is exhausted.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LINKCC	Link Channel Control Allows DMA channels to have their transfers linked. The current DMA channel triggers a DMA request to the linked channels (LCH1 or LCH2) depending on the condition described by the LINKCC bits. If not in cycle steal mode (DCRn[CS]=0) and LINKCC equals 01 or 10, no link to LCH1 occurs. If LINKCC equals 01, a link to LCH1 is created after each cycle-steal transfer performed by the current DMA channel is completed. As the last cycle-steal is performed and the BCR reaches zero, then the link to LCH1 is closed and a link to LCH2 is created. 00 No channel-to-channel linking 01 Perform a link to channel LCH1 after each cycle-steal transfer followed by a link to LCH2 after the BCR decrements to 0. 10 Perform a link to channel LCH1 after each cycle-steal transfer 11 Perform a link to channel LCH1 after the BCR decrements to 0.
3–2 LCH1	Link Channel 1 Indicates the DMA channel assigned as link channel 1. The link channel number cannot be the same as the currently executing channel, and generates a configuration error if this is attempted (DSRn[CE] is set). 00 DMA Channel 0 01 DMA Channel 1

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	10 DMA Channel 2 11 DMA Channel 3
LCH2	Link Channel 2 Indicates the DMA channel assigned as link channel 2. The link channel number cannot be the same as the currently executing channel, and generates a configuration error if this is attempted (DSRn[CE] is set). 00 DMA Channel 0 01 DMA Channel 1 10 DMA Channel 2 11 DMA Channel 3

23.4 Functional Description

In the following discussion, the term DMA request implies that DCR*n*[START] is set, or DCRn[ERQ] is set and then followed by assertion of the properly selected DMA peripheral request. DCR*n*[START] is cleared when the channel is activated.

Before initiating a dual-address access, the DMA module verifies that DCR*n*[SSIZE] and DCR*n*[DSIZE] are consistent with the source and destination addresses. If they are not consistent, the configuration error bit, DSR*n*[CE], is set. If misalignment is detected, no transfer occurs, DSR*n*[CE] is set, and, depending on the DCR configuration, an interrupt event may be issued. If the auto-align bit, DCR*n*[AA], is set, error checking is performed on the appropriate registers.

A read/write transfer sequence reads data from the source address and writes it to the destination address. The number of bytes transferred is the largest of the sizes specified by DCR*n*[SSIZE] and DCR*n*[DSIZE] in the DMA Control Registers (DCRn).

Source and destination address registers (SAR*n* and DAR*n*) can be programmed in the DCR*n* to increment at the completion of a successful transfer.

23.4.1 Transfer requests (Cycle-Steal and Continuous modes)

The DMA channel supports software-initiated or peripheral-initiated requests. A request is issued by setting DCR*n*[START] or when the selected peripheral request asserts and DCRn[ERQ] is set. Setting DCR*n*[ERQ] enables recognition of the peripheral DMA requests. Selecting between cycle-steal and continuous modes minimizes bus usage for either type of request.

- Cycle-steal mode (DCR*n*[CS] = 1)—Only one complete transfer from source to destination occurs for each request. If DCRn[ERQ] is set, the request is peripheral initiated. A software-initiated request is enabled by setting DCR*n*[START].
- Continuous mode (DCR*n*[CS] = 0)—After a software-initiated or peripheral request, the DMA continuously transfers data until BCRn reaches 0. The DMA performs the specified number of transfers, then retires the channel.

In either mode, the crossbar switch performs independent arbitration on each slave port after each transaction.

23.4.2 Channel initialization and startup

Before a data transfer starts, the channel's transfer control descriptor must be initialized with information describing configuration, request-generation method, and pointers to the data to be moved.

23.4.2.1 Channel prioritization

The four DMA channels are prioritized based on number, with channel 0 having highest priority and channel 3 having the lowest, that is, channel 0 > channel 1 > channel 2 > channel 3.

Simultaneous peripheral requests activate the channels based on this priority order. Once activated, a channel runs to completion as defined by DCR*n*[CS] and BCR*n*.

23.4.2.2 Programming the DMA Controller Module

CAUTION

> During a channel's execution, writes to programming model registers can corrupt the data transfer. The DMA module itself does not have a mechanism to prevent writes to registers during a channel's execution.

General guidelines for programming the DMA are:

- TCD*n* is initialized.

    - SAR*n* is loaded with the source (read) address. If the transfer is from a peripheral device to memory or to another peripheral, the source address is the location of the peripheral data register. If the transfer is from memory to a peripheral device or to memory, the source address is the starting address of the data block. This can be any appropriately aligned address.
    - DAR*n* is initialized with the destination (write) address. If the transfer is from a peripheral device to memory, or from memory to memory, DAR*n* is loaded with the starting address of the data block to be written. If the transfer is from memory to a peripheral device, or from a peripheral device to a peripheral device, DAR*n* is loaded with the address of the peripheral data register. This address can be any appropriately aligned address.
    - SAR*n* and DAR*n* change after each data transfer depending on DCR*n*[SSIZE, DSIZE, SINC, DINC, SMOD, DMOD] and the starting addresses. Increment values can be 1, 2, or 4 for 8-bit, 16-bit, or 32-bit transfers, respectively. If the address register is programmed to remain unchanged, the register is not incremented after the data transfer.
    - BCR*n*[BCR] must be loaded with the total number of bytes to be transferred. It is decremented by 1, 2, or 4 at the end of each transfer, depending on the transfer size. DSR*n*[DONE] must be cleared for channel startup.
- After the channel has been initialized, it may be started by setting DCR*n*[START] or a properly selected peripheral DMA request, depending on the status of DCR*n*[ERQ]. For a software-initiated transfer, the channel can be started by setting DCR*n*[START] as part of a single 32-bit write to the last 32 bits of the TCD*n*; that is, it is not required to write the DCR*n* with START cleared and then perform a second write to explicitly set START.
- Programming the channel for a software-initiated request causes the channel to request the system bus and start transferring data immediately. If the channel is programmed for peripheral-initiated request, a properly selected peripheral DMA request must be asserted before the channel begins the system bus transfers.
- The hardware can automatically clear DCR*n*[ERQ], disabling the peripheral request, when BCR*n* reaches zero by setting DCR*n*[D_REQ].
- Changes to DCR*n* are effective immediately while the channel is active. To avoid problems with changing a DMA channel setup, write a one to DSR*n*[DONE] to stop the DMA channel.

23.4.3 Dual-Address Data Transfer Mode

Each channel supports dual-address transfers. Dual-address transfers consist of a source data read and a destination data write. The DMA controller module begins a dual-address transfer sequence after a DMA request. If no error condition exists, DSR*n*[REQ] is set.

- Dual-address read—The DMA controller drives the SAR*n* value onto the system address bus. If DCR*n*[SINC] is set, the SAR*n* increments by the appropriate number of bytes upon a successful read cycle. When the appropriate number of read cycles complete (multiple reads if the destination size is larger than the source), the DMA initiates the write portion of the transfer.

If a termination error occurs, DSR*n*[BES, DONE] are set and DMA transactions stop.

- Dual-address write—The DMA controller drives the DAR*n* value onto the system address bus. When the appropriate number of write cycles complete (multiple writes if the source size is larger than the destination), DAR*n* increments by the appropriate number of bytes if DCRn[DINC] is set. BCRn decrements by the appropriate number of bytes. DSR*n*[DONE] is set when BCR*n* reaches zero. If the BCR*n* is greater than zero, another read/write transfer is initiated if continuous mode is enabled (DCRn[CS] = 0).

  If a termination error occurs, DSR*n*[BED, DONE] are set and DMA transactions stop.

23.4.4 Advanced Data Transfer Controls: Auto-Alignment

Typically, auto-alignment for DMA transfers applies for transfers of large blocks of data. As a result, it does not apply for peripheral-initiated cycle-steal transfers.

Auto-alignment allows block transfers to occur at the optimal size based on the address, byte count, and programmed size. To use this feature, DCR*n*[AA] must be set. The source is auto-aligned if DCR*n*[SSIZE] indicates a transfer size larger than DCR*n*[DSIZE]. Source alignment takes precedence over the destination when the source and destination sizes are equal. Otherwise, the destination is auto-aligned. The address register chosen for alignment increments regardless of the increment value. Configuration error checking is performed on registers not chosen for alignment.

If BCR*n* is greater than 16, the address determines transfer size. Transfers of 8 bits, 16 bits, or 32 bits are transferred until the address is aligned to the programmed size boundary, at which time accesses begin using the programmed size. If BCR*n* is less than 16 at the start of a transfer, the number of bytes remaining dictates transfer size.

Consider this example:

- AA equals 1.
- SAR*n* equals 0x2000_0001.
- BCR*n* equals 0x00_00F0.
- SSIZE equals 00 (32 bits).
- DSIZE equals 01 (8 bits).

Because SSIZE > DSIZE, the source is auto-aligned. Error checking is performed on destination registers. The access sequence is as follows:

1. Read 1 byte from 0x2000_0001, increment SAR*n*, write 1 byte (using DAR*n*).
2. Read 2 bytes from 0x2000_0002, increment SAR*n*, write 2 bytes.
3. Read 4 bytes from 0x2000_0004, increment SAR*n*, write 4 bytes.
4. Repeat 4-byte operations until SAR*n* equals 0x2000_00F0.
5. Read byte from 0x2000_00F0, increment SAR*n*, write byte.

If DSIZE is another size, data writes are optimized to write the largest size allowed based on the address, but not exceeding the configured size.

23.4.5 Termination

An unsuccessful transfer can terminate for one of the following reasons:

- Error conditions—When the DMA encounters a read or write cycle that terminates with an error condition, DSR*n*[BES] is set for a read and DSR*n*[BED] is set for a write before the transfer is halted. If the error occurred in a write cycle, data in the internal holding registers is lost.
- Interrupts—If DCR*n*[EINT] is set, the DMA drives the appropriate interrupt request signal. The processor can read DSR*n* to determine whether the transfer terminated successfully or with an error. DSR*n*[DONE] is then written with a 1 to clear the interrupt, DSR*n*[DONE], and error status bits.

Chapter 24
Multipurpose Clock Generator (MCG)

The Multipurpose Clock Generator (MCG) module provides several clock source choices for the MCU. The MCG operates in conjuction with a crystal oscillator, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock.

24.1 Introduction

The multipurpose clock generator (MCG) module provides several clock source choices for the MCU.

The module contains a frequency-locked loop (FLL). The FLL is controllable by either an internal or an external reference clock. The module can select either an FLL output clock, or a reference clock (internal or external) as a source for the MCU system clock. The MCG operates in conjuction with a crystal oscillator, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock.

24.1.1 Features

Key features of the MCG module are:

- Frequency-locked loop (FLL):
  - Digitally-controlled oscillator (DCO)
  - DCO frequency range is programmable for up to four different frequency ranges.
  - Option to program and maximize DCO output frequency for a low frequency external reference clock source.

    - Option to prevent FLL from resetting its current locked frequency when switching clock modes if FLL reference frequency is not changed.
    - Internal or external reference clock can be used as the FLL source.
    - Can be used as a clock source for other on-chip peripherals.
- Internal reference clock generator:
    - Slow clock with nine trim bits for accuracy
    - Fast clock with four trim bits
    - Can be used as source clock for the FLL. In FEI mode, only the slow Internal Reference Clock (IRC) can be used as the FLL source.
    - Either the slow or the fast clock can be selected as the clock source for the MCU.
    - Can be used as a clock source for other on-chip peripherals.
- External clock from the RF Oscillator :
    - Can be used as a source for the FLL.
    - Can be selected as the clock source for the MCU.
- External clock from the Real Time Counter (RTC):
    - Can be used as a source for the FLL only.
    - Can be selected as the clock source for the MCU.
- External clock monitor with reset and interrupt request capability to check for external clock failure when running in FBE, BLPE, or FEE modes
- Internal Reference Clocks Auto Trim Machine (ATM) capability using an external clock as a reference
- Reference dividers for the FLL are provided
- Reference dividers for the Fast Internal Reference Clock are provided
- MCG FLL Clock (MCGFLLCLK) is provided as a clock source for other on-chip peripherals
- MCG Internal Reference Clock (MCGIRCLK) is provided as a clock source for other on-chip peripherals

This figure presents the block diagram of the MCG module.

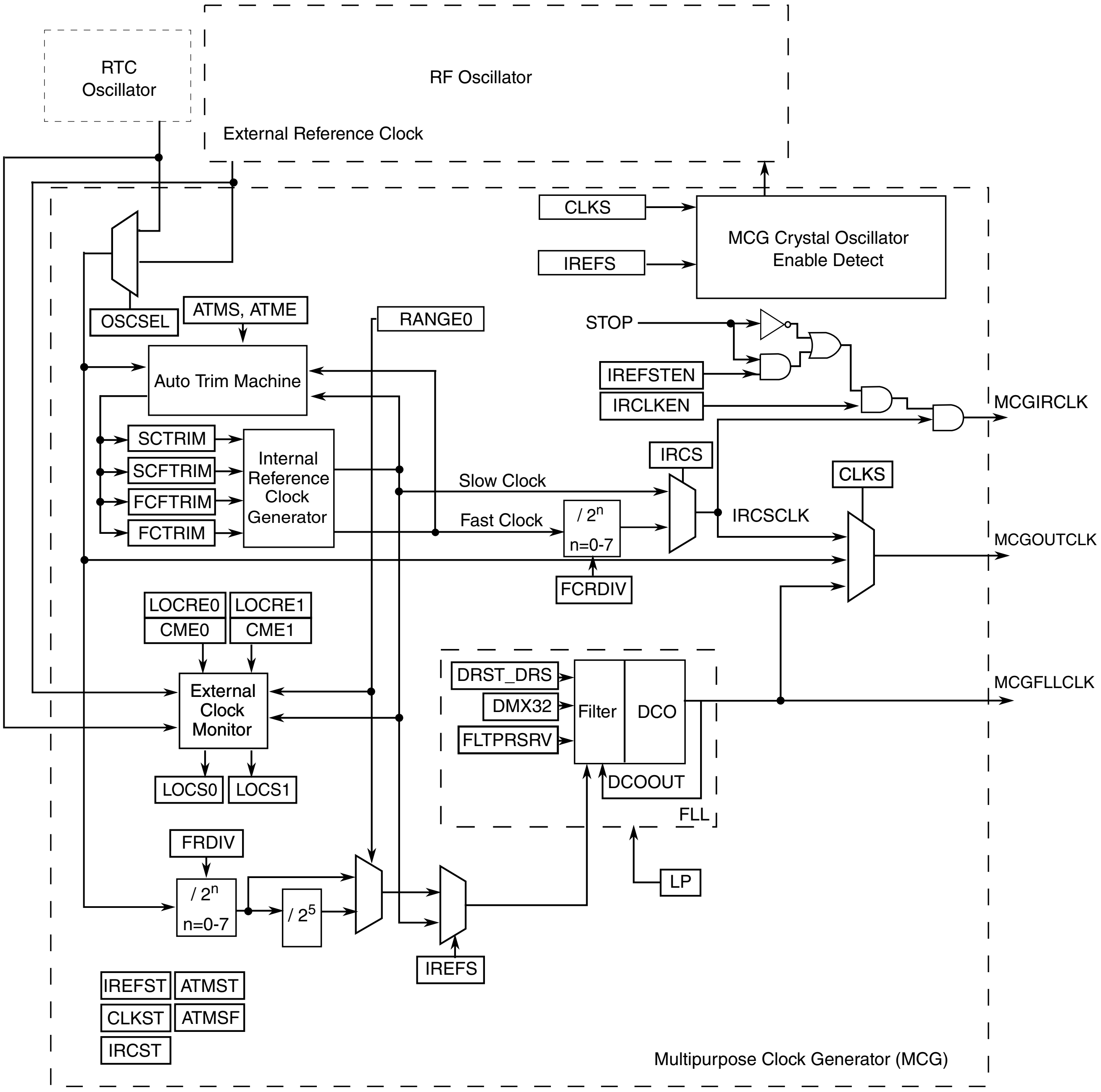


Figure 24-1. Multipurpose Clock Generator (MCG) block diagram

24.1.2 Modes of Operation

The MCG has the following modes of operation: FEI, FEE, FBI, FBE, BLPI, BLPE, and Stop. For details, see MCG modes of operation.

24.2 External Signal Description

There are no MCG signals that connect off chip.

24.3 Memory Map/Register Definition

This section includes the memory map and register definition.

The MCG registers can only be written when in supervisor mode. Write accesses when in user mode will result in a bus error. Read accesses may be performed in both supervisor and user mode.

MCG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_4000	MCG Control 1 Register (MCG_C1)	8	R/W	04h	24.3.1/410
4006_4001	MCG Control 2 Register (MCG_C2)	8	R/W	See section	24.3.2/412
4006_4002	MCG Control 3 Register (MCG_C3)	8	R/W	Undefined	24.3.3/413
4006_4003	MCG Control 4 Register (MCG_C4)	8	R/W	Undefined	24.3.4/414
4006_4004	MCG Control 5 Register (MCG_C5)	8	R/W	00h	24.3.5/415
4006_4005	MCG Control 6 Register (MCG_C6)	8	R/W	00h	24.3.5/415
4006_4006	MCG Status Register (MCG_S)	8	R	10h	24.3.6/416
4006_4008	MCG Status and Control Register (MCG_SC)	8	R/W	02h	24.3.7/417
4006_400A	MCG Auto Trim Compare Value High Register (MCG_ATCVH)	8	R/W	00h	24.3.8/418
4006_400B	MCG Auto Trim Compare Value Low Register (MCG_ATCVL)	8	R/W	00h	24.3.9/419
4006_400C	MCG Control 7 Register (MCG_C7)	8	R/W	00h	24.3.10/419
4006_400D	MCG Control 8 Register (MCG_C8)	8	R/W	See section	24.3.11/420
4006_4011	MCG Control 12 Register (MCG_C12)	8	R/W	00h	24.3.12/421
4006_4012	MCG Status 2 Register (MCG_S2)	8	R/W	00h	24.3.12/421
4006_4013	MCG Test 3 Register (MCG_T3)	8	R/W	00h	24.3.12/421

24.3.1 MCG Control 1 Register (MCG_C1)

Address: 4006_4000h base + 0h offset = 4006_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

MCG_C1 field descriptions

Field	Description
7–6 CLKS	Clock Source Select Selects the clock source for MCGOUTCLK . 00 Encoding 0 — Output of FLL is selected. 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Reserved.
5–3 FRDIV	FLL External Reference Divider Selects the amount to divide down the external reference clock for the FLL. The resulting frequency must be in the range 31.25 kHz to 39.0625 kHz (This is required when FLL/DCO is the clock source for MCGOUTCLK . In FBE mode, it is not required to meet this range, but it is recommended in the cases when trying to enter a FLL mode from FBE). 000 If RANGE = 0 or OSCSEL=1 , Divide Factor is 1; for all other RANGE values, Divide Factor is 32. 001 If RANGE = 0 or OSCSEL=1 , Divide Factor is 2; for all other RANGE values, Divide Factor is 64. 010 If RANGE = 0 or OSCSEL=1 , Divide Factor is 4; for all other RANGE values, Divide Factor is 128. 011 If RANGE = 0 or OSCSEL=1 , Divide Factor is 8; for all other RANGE values, Divide Factor is 256. 100 If RANGE = 0 or OSCSEL=1 , Divide Factor is 16; for all other RANGE values, Divide Factor is 512. 101 If RANGE = 0 or OSCSEL=1 , Divide Factor is 32; for all other RANGE values, Divide Factor is 1024. 110 If RANGE = 0 or OSCSEL=1 , Divide Factor is 64; for all other RANGE values, Divide Factor is 1280 . 111 If RANGE = 0 or OSCSEL=1 , Divide Factor is 128; for all other RANGE values, Divide Factor is 1536 .
2 IREFS	Internal Reference Select Selects the reference clock source for the FLL. 0 External reference clock is selected. 1 The slow internal reference clock is selected.
1 IRCLKEN	Internal Reference Clock Enable Enables the internal reference clock for use as MCGIRCLK. 0 MCGIRCLK inactive. 1 MCGIRCLK active.
0 IREFSTEN	Internal Reference Stop Enable Controls whether or not the internal reference clock remains enabled when the MCG enters Stop mode. 0 Internal reference clock is disabled in Stop mode. 1 Internal reference clock is enabled in Stop mode if IRCLKEN is set or if MCG is in FEI, FBI, or BLPI modes before entering Stop mode.

24.3.2 MCG Control 2 Register (MCG_C2)

Address: 4006_4000h base + 1h offset = 4006_4001h

Bit	7	6	5	4	3	2	1	0
Read Write	LOCRE0	FCFTRIM	RANGE		HGO	EREFS	LP	IRCS
Reset	1	1	0	0	0	0	0	0

MCG_C2 field descriptions

Field	Description
7 LOCRE0	Loss of Clock Reset Enable Determines whether an interrupt or a reset request is made following a loss of OSC0 external reference clock. The LOCRE0 only has an affect when CME0 is set. 0 Interrupt request is generated on a loss of OSC0 external reference clock. 1 Generate a reset request on a loss of OSC0 external reference clock.
6 FCFTRIM	Fast Internal Reference Clock Fine Trim FCFTRIM controls the smallest adjustment of the fast internal reference clock frequency. Setting FCFTRIM increases the period and clearing FCFTRIM decreases the period by the smallest amount possible. If an FCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.
5–4 RANGE	Frequency Range Select Selects the frequency range for the crystal oscillator or external clock source. See the Oscillator (OSC) chapter for more details and the device data sheet for the frequency ranges used. 00 Encoding 0 — Low frequency range selected for the crystal oscillator . 01 Encoding 1 — High frequency range selected for the crystal oscillator . 1X Encoding 2 — Very high frequency range selected for the crystal oscillator .
3 HGO	High Gain Oscillator Select Controls the crystal oscillator mode of operation. See the Oscillator (OSC) chapter for more details. 0 Configure crystal oscillator for low-power operation. 1 Configure crystal oscillator for high-gain operation.
2 EREFS	External Reference Select Selects the source for the external reference clock. See the Oscillator (OSC) chapter for more details. 0 External reference clock requested. 1 Oscillator requested.
1 LP	Low Power Select Controls whether the FLL is disabled in BLPI and BLPE modes. In FBE mode, setting this bit to 1 will transition the MCG into BLPE mode; in FBI mode, setting this bit to 1 will transition the MCG into BLPI mode. In any other MCG mode, LP bit has no affect. 0 FLL is not disabled in bypass modes. 1 FLL is disabled in bypass modes (lower power)

Table continues on the next page...

MCG_C2 field descriptions (continued)

Field	Description
0 IRCS	Internal Reference Clock Select Selects between the fast or slow internal reference clock source. 0 Slow internal reference clock selected. 1 Fast internal reference clock selected.

24.3.3 MCG Control 3 Register (MCG_C3)

Address: 4006_4000h base + 2h offset = 4006_4002h

Bit	7	6	5	4	3	2	1	0
Read Write	SCTRIM							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

MCG_C3 field descriptions

Field	Description
SCTRIM	Slow Internal Reference Clock Trim Setting SCTRIM [1] controls the slow internal reference clock frequency by controlling the slow internal reference clock period. The SCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. An additional fine trim bit is available in C4 register as the SCFTRIM bit. Upon reset, this value is loaded with a factory trim value. If an SCTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.

[1] A value for SCTRIM is loaded during reset from a factory programmed location.

24.3.4 MCG Control 4 Register (MCG_C4)

NOTE

Reset values for DRST and DMX32 bits are 0.

Address: 4006_4000h base + 3h offset = 4006_4003h

Bit	7	6	5	4	3	2	1	0
Read Write	DMX32	DRST_DRS		FCTRIM				SCFTRIM
Reset	0	0	0	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.
- x = Undefined at reset.

MCG_C4 field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>7
DMX32</td><td>DCO Maximum Frequency with 32.768 kHz Reference
The DMX32 bit controls whether the DCO frequency range is narrowed to its maximum frequency with a 32.768 kHz reference.
The following table identifies settings for the DCO frequency range.
<b>NOTE:</b> The system clocks derived from this source should not exceed their specified maximums.
<table>
<tr><th>DRST_DRS</th><th>DMX32</th><th>Reference Range</th><th>FLL Factor</th><th>DCO Range</th></tr>
<tr><td>00</td><td>0</td><td>31.25–39.0625 kHz</td><td>640</td><td>20–25 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>732</td><td>24 MHz</td></tr>
<tr><td>01</td><td>0</td><td>31.25–39.0625 kHz</td><td>1280</td><td>40–50 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>1464</td><td>48 MHz</td></tr>
<tr><td>10</td><td>0</td><td>31.25–39.0625 kHz</td><td>1920</td><td>60–75 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>2197</td><td>72 MHz</td></tr>
<tr><td>11</td><td>0</td><td>31.25–39.0625 kHz</td><td>2560</td><td>80–100 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>2929</td><td>96 MHz</td></tr>
</table>
0 DCO has a default range of 25%.
1 DCO is fine-tuned for maximum frequency with 32.768 kHz reference.</td></tr>
<tr><td>6–5
DRST_DRS</td><td>DCO Range Select
The DRS bits select the frequency range for the FLL output, DCOOUT. When the LP bit is set, writes to the DRS bits are ignored. The DRST read field indicates the current frequency range for DCOOUT. The DRST field does not update immediately after a write to the DRS field due to internal synchronization between clock domains. See the DCO Frequency Range table for more details.</td></tr>
</table>

Table continues on the next page...

MCG_C4 field descriptions (continued)

Field	Description
	00 Encoding 0 — Low range (reset default). 01 Encoding 1 — Mid range. 10 Encoding 2 — Mid-high range. 11 Encoding 3 — High range.
4–1 FCTRIM	Fast Internal Reference Clock Trim Setting FCTRIM [1] controls the fast internal reference clock frequency by controlling the fast internal reference clock period. The FCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. If an FCTRIM[3:0] value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.
0 SCFTRIM	Slow Internal Reference Clock Fine Trim SCFTRIM [2] controls the smallest adjustment of the slow internal reference clock frequency. Setting SCFTRIM increases the period and clearing SCFTRIM decreases the period by the smallest amount possible. If an SCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.

1. A value for FCTRIM is loaded during reset from a factory programmed location.
2. A value for SCFTRIM is loaded during reset from a factory programmed location .

24.3.5 MCG Control 5 Register (MCG_C5)

Address: 4006_4000h base + 4h offset = 4006_4004h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C5 field descriptions

Field	Description
Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.

24.3.5 MCG Control 6 Register (MCG_C6)

Address: 4006_4000h base + 5h offset = 4006_4005h

Bit	7	6	5	4	3	2	1	0
Read	0		CME0	0				
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C6 field descriptions

Field	Description
7–6 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
5 CME0	Clock Monitor Enable Determines if an interrupt or a reset request (see MCG_C2[LOCRE0]) is made following a loss of external clock indication. The CME0 bit should only be set to a logic 1 when the MCG is in an operational mode that uses the external clock (FEE, FBE, or BLPE). Whenever the CME0 bit is set to a logic 1, the value of the RANGE bits in the C2 register should not be changed. CME0 bit should be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur when in Stop mode. CME0 should also be set to a logic 0 before entering VLPR or VLPW power modes if the MCG is in BLPE mode. 0 External clock monitor is disabled. 1 Generate an interrupt or a reset request (see MCG_C2[LOCRE0]) on loss of external clock.
Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.

24.3.6 MCG Status Register (MCG_S)

Address: 4006_4000h base + 6h offset = 4006_4006h

Bit	7	6	5	4	3	2	1	0
Read	0			IREFST	CLKST		OSCINIT0	IRCST
Write								
Reset	0	0	0	1	0	0	0	0

MCG_S field descriptions

Field	Description
7–5 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
4 IREFST	Internal Reference Status This bit indicates the current source for the FLL reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains. 0 Source of FLL reference clock is the external reference clock. 1 Source of FLL reference clock is the internal reference clock.
3–2 CLKST	Clock Mode Status These bits indicate the current clock mode. The CLKST bits do not update immediately after a write to the CLKS bits due to internal synchronization between clock domains.

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
	00 Encoding 0 — Output of the FLL is selected (reset default). 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Reserved.
1 OSCINIT0	OSC Initialization This bit, which resets to 0, is set to 1 after the initialization cycles of the RF oscillator clock have completed. After being set, the bit is cleared to 0 if the OSC is subsequently disabled. See the OSC module's detailed description for more information.
0 IRCST	Internal Reference Clock Status The IRCST bit indicates the current source for the internal reference clock select clock (IRCSCLK). The IRCST bit does not update immediately after a write to the IRCS bit due to internal synchronization between clock domains. The IRCST bit will only be updated if the internal reference clock is enabled, either by the MCG being in a mode that uses the IRC or by setting the C1[IRCLKEN] bit . 0 Source of internal reference clock is the slow clock (32 kHz IRC). 1 Source of internal reference clock is the fast clock (4 MHz IRC).

24.3.7 MCG Status and Control Register (MCG_SC)

Address: 4006_4000h base + 8h offset = 4006_4008h

Bit	7	6	5	4	3	2	1	0
Read	ATME	ATMS	ATMF	FLTPRSRV	FCRDIV			LOCS0
Write			w1c					w1c
Reset	0	0	0	0	0	0	1	0

MCG_SC field descriptions

Field	Description
7 ATME	Automatic Trim Machine Enable Enables the Auto Trim Machine to start automatically trimming the selected Internal Reference Clock. **NOTE:** ATME deasserts after the Auto Trim Machine has completed trimming all trim bits of the IRCS clock selected by the ATMS bit. Writing to C1, C3, C4, and SC registers or entering Stop mode aborts the auto trim operation and clears this bit. 0 Auto Trim Machine disabled. 1 Auto Trim Machine enabled.
6 ATMS	Automatic Trim Machine Select Selects the IRCS clock for Auto Trim Test. 0 32 kHz Internal Reference Clock selected. 1 4 MHz Internal Reference Clock selected.

Table continues on the next page...

MCG_SC field descriptions (continued)

Field	Description
5 ATMF	Automatic Trim Machine Fail Flag Fail flag for the Automatic Trim Machine (ATM). This bit asserts when the Automatic Trim Machine is enabled, ATME=1, and a write to the C1, C3, C4, and SC registers is detected or the MCG enters into any Stop mode. A write to ATMF clears the flag. 0 Automatic Trim Machine completed normally. 1 Automatic Trim Machine failed.
4 FLTPRSRV	FLL Filter Preserve Enable This bit will prevent the FLL filter values from resetting allowing the FLL output frequency to remain the same during clock mode changes where the FLL/DCO output is still valid. (Note: This requires that the FLL reference frequency to remain the same as what it was prior to the new clock mode switch. Otherwise FLL filter and frequency values will change.) 0 FLL filter and FLL frequency will reset on changes to currect clock mode. 1 Fll filter and FLL frequency retain their previous values during new clock mode change.
3–1 FCRDIV	Fast Clock Internal Reference Divider Selects the amount to divide down the fast internal reference clock. The resulting frequency will be in the range 31.25 kHz to 4 MHz (Note: Changing the divider when the Fast IRC is enabled is not supported). 000 Divide Factor is 1 001 Divide Factor is 2. 010 Divide Factor is 4. 011 Divide Factor is 8. 100 Divide Factor is 16 101 Divide Factor is 32 110 Divide Factor is 64 111 Divide Factor is 128.
0 LOCS0	OSC0 Loss of Clock Status The LOCS0 indicates when a loss of OSC0 reference clock has occurred. The LOCS0 bit only has an effect when CME0 is set. This bit is cleared by writing a logic 1 to it when set. 0 Loss of OSC0 has not occurred. 1 Loss of OSC0 has occurred.

24.3.8 MCG Auto Trim Compare Value High Register (MCG_ATCVH)

Address: 4006_4000h base + Ah offset = 4006_400Ah

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVH							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVH field descriptions

Field	Description
ATCVH	ATM Compare Value High Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

24.3.9 MCG Auto Trim Compare Value Low Register (MCG_ATCVL)

Address: 4006_4000h base + Bh offset = 4006_400Bh

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVL							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVL field descriptions

Field	Description
ATCVL	ATM Compare Value Low Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

24.3.10 MCG Control 7 Register (MCG_C7)

Address: 4006_4000h base + Ch offset = 4006_400Ch

Bit	7	6	5	4	3	2	1	0
Read	0		0				0	OSCSEL
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C7 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–2 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCG_C7 field descriptions (continued)

Field	Description
0 OSCSEL	MCG OSC Clock Select Selects the MCG FLL external reference clock 0 Selects Oscillator (OSCCLK). 1 Selects 32 kHz RTC Oscillator.

24.3.11 MCG Control 8 Register (MCG_C8)

Address: 4006_4000h base + Dh offset = 4006_400Dh

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td rowspan="2">LOCRE1</td><td>0</td><td rowspan="2">CME1</td><td colspan="4">0</td><td>LOCS1</td></tr>
<tr><td>Write</td><td></td><td colspan="4"></td><td>w1c</td></tr>
<tr><td>Reset</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

MCG_C8 field descriptions

Field	Description
7 LOCRE1	Loss of Clock Reset Enable Determines if a interrupt or a reset request is made following a loss of RTC external reference clock. The LOCRE1 only has an affect when CME1 is set. 0 Interrupt request is generated on a loss of RTC external reference clock. 1 Generate a reset request on a loss of RTC external reference clock
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 CME1	Clock Monitor Enable1 Enables the loss of clock monitoring circuit for the output of the RTC external reference clock. The LOCRE1 bit will determine whether an interrupt or a reset request is generated following a loss of RTC clock indication. The CME1 bit should be set to a logic 1 when the MCG is in an operational mode that uses the RTC as its external reference clock or if the RTC is operational. CME1 bit must be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur when in Stop mode. CME1 should also be set to a logic 0 before entering VLPR or VLPW power modes. 0 External clock monitor is disabled for RTC clock. 1 External clock monitor is enabled for RTC clock.
4–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LOCS1	RTC Loss of Clock Status This bit indicates when a loss of clock has occurred. This bit is cleared by writing a logic 1 to it when set. 0 Loss of RTC has not occur. 1 Loss of RTC has occur

24.3.12 MCG Control 12 Register (MCG_C12)

Address: 4006_4000h base + 11h offset = 4006_4011h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C12 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

24.3.12 MCG Status 2 Register (MCG_S2)

Address: 4006_4000h base + 12h offset = 4006_4012h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_S2 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

24.3.12 MCG Test 3 Register (MCG_T3)

Address: 4006_4000h base + 13h offset = 4006_4013h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_T3 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

24.4 Functional description

24.4.1 MCG mode state diagram

The seven states of the MCG are shown in the following figure and are described in Table 24-1. The arrows indicate the permitted MCG mode transitions.

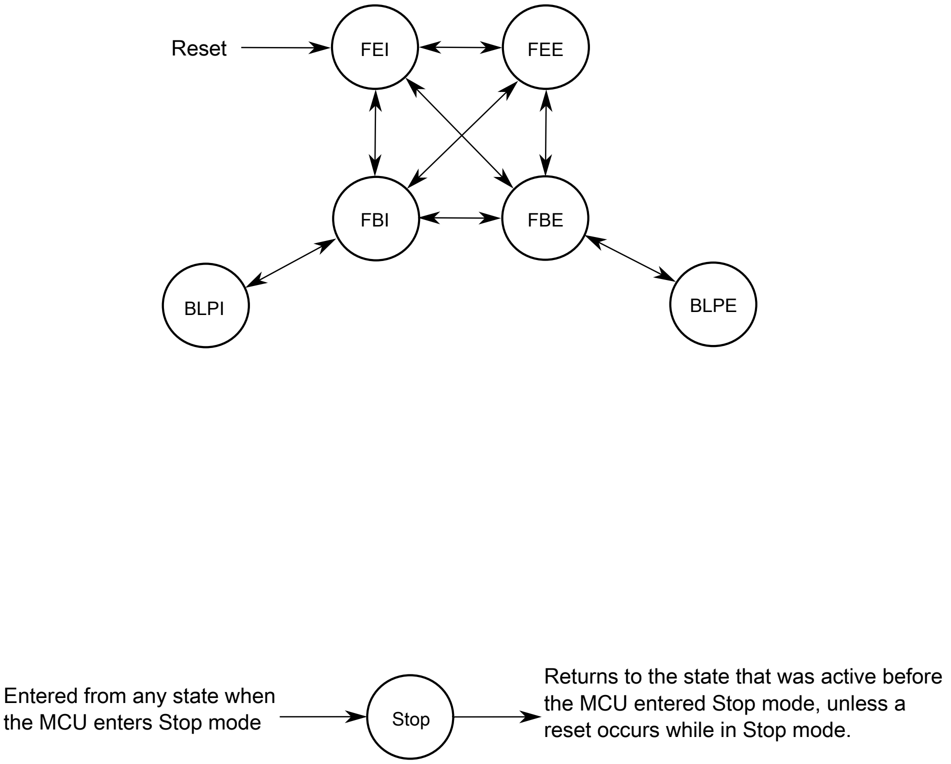

Reset
FEI
FEE
FBI
FBE
BLPI
BLPE
Entered from any state when the MCU enters Stop mode
Stop
Returns to the state that was active before the MCU entered Stop mode, unless a reset occurs while in Stop mode.

Figure 24-2. MCG mode state diagram

24.4.1.1 MCG modes of operation

The MCG operates in one of the following modes.

Note

> The MCG restricts transitions between modes. For the permitted transitions, see Figure 24-2.

Table 24-1. MCG modes of operation

Mode	Description
FLL Engaged Internal (FEI)	FLL engaged internal (FEI) is the default mode of operation and is entered when all the following condtions occur: • 00 is written to C1[CLKS]. • 1 is written to C1[IREFS]. In FEI mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the 32 kHz Internal Reference Clock (IRC). The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details.
FLL Engaged External (FEE)	FLL engaged external (FEE) mode is entered when all the following conditions occur: • 00 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz In FEE mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the external reference clock. The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the external reference frequency, as specified by C1[FRDIV] and C2[RANGE]. See the C4[DMX32] bit description for more details.
FLL Bypassed Internal (FBI)	FLL bypassed internal (FBI) mode is entered when all the following conditions occur: • 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C2[LP]. In FBI mode, the MCGOUTCLK is derived either from the slow (32 kHz IRC) or fast (4 MHz IRC) internal reference clock, as selected by the C2[IRCS] bit. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the C2[IRCS] selected internal reference clock. The FLL clock (DCOCLK) is controlled by the slow internal reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details.
FLL Bypassed External (FBE)	FLL bypassed external (FBE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz. • 0 is written to C2[LP]. In FBE mode, the MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the external reference clock. The FLL clock (DCOCLK) is controlled by the external reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the divided external reference frequency. See the C4[DMX32] bit description for more details.
Bypassed Low Power Internal (BLPI)	Bypassed Low Power Internal (BLPI) mode is entered when all the following conditions occur:

Table continues on the next page...

Table 24-1. MCG modes of operation (continued)

Mode	Description
	• 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 1 is written to C2[LP]. In BLPI mode, MCGOUTCLK is derived from the internal reference clock. The FLL is disabled
Bypassed Low Power External (BLPE)	Bypassed Low Power External (BLPE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C2[LP]. In BLPE mode, MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is disabled
Stop	Entered whenever the MCU enters a Stop state. The power modes are chip specific. For power mode assignments, see the chapter that describes how modules are configured and MCG behavior during Stop recovery. Entering Stop mode, the FLL is disabled, and all MCG clock signals are static except in the following case: MCGIRCLK is active in Normal Stop mode when all the following conditions become true: • C1[IRCLKEN] = 1 • C1[IREFSTEN] = 1 **NOTE:** • In VLPS Stop Mode, the MCGIRCLK can be programmed to stay enabled and continue running if C1[IRCLKEN] = 1, C1[IREFSTEN]=1, and Fast IRC clock is selected (C2[IRCS] = 1)

NOTE

For the chip-specific modes of operation, see the power management chapter of this MCU.

24.4.1.2 MCG mode switching

C1[IREFS] can be changed at any time, but the actual switch to the newly selected reference clocks is shown by S[IREFST]. When switching between engaged internal and engaged external modes, the FLL will begin locking again after the switch is completed.

C1[CLKS] can also be changed at any time, but the actual switch to the newly selected clock is shown by S[CLKST]. If the newly selected clock is not available, the previous clock will remain selected.

The C4[DRST_DRS] write bits can be changed at any time except when C2[LP] bit is 1. If C4[DRST_DRS] write bits are changed while in FLL engaged internal (FEI) or FLL engaged external (FEE) mode, the MCGOUTCLK switches to the new selected DCO range within three clocks of the selected DCO clock. After switching to the new DCO (indicated by the updated C4[DRST_DRS] read bits), the FLL remains unlocked for several reference cycles. The FLL lock time is provided in the device data sheet as $t_{fll\_acquire}$.

24.4.2 Low-power bit usage

C2[LP] is provided to allow the FLL to be disabled and thus conserve power when these systems are not being used. C4[DRST_DRS] can not be written while C2[LP] is 1. However, in some applications, it may be desirable to enable the FLL and allow it to lock for maximum accuracy before switching to an engaged mode. Do this by writing 0 to C2[LP].

24.4.3 MCG Internal Reference Clocks

This module supports two internal reference clocks with nominal frequencies of 32 kHz (slow IRC) and 4 MHz (fast IRC). The fast IRC frequency can be divided down by programming of the FCRDIV to produce a frequency range of 32 kHz to 4 MHz.

24.4.3.1 MCG Internal Reference Clock

The MCG Internal Reference Clock (MCGIRCLK) provides a clock source for other on-chip peripherals and is enabled when C1[IRCLKEN]=1. When enabled, MCGIRCLK is driven by either the fast internal reference clock (4 MHz IRC which can be divided down by the FRDIV factors) or the slow internal reference clock (32 kHz IRC). The IRCS clock frequency can be re-targeted by trimming the period of its IRCS selected internal reference clock. This can be done by writing a new trim value to the C3[SCTRIM]:C4[SCFTRIM] bits when the slow IRC clock is selected or by writing a new trim value to C4[FCTRIM]:C2[FCFTRIM] when the fast IRC clock is selected. The internal reference clock period is proportional to the trim value written. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) and C4[FCTRIM]:C2[FCFTRIM] (if C2[IRCS]=1) bits affect the MCGOUTCLK frequency if the MCG is in FBI or BLPI modes. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) bits also affect the MCGOUTCLK frequency if the MCG is in FEI mode.

Additionally, this clock can be enabled in Stop mode by setting C1[IRCLKEN] and C1[IREFSTEN], otherwise this clock is disabled in Stop mode.

24.4.4 External Reference Clock

The MCG module can support an external reference clock in all modes. See the device datasheet for external reference frequency range. When C1[IREFS] is set, the external reference clock will not be used by the FLL. In these mode, the frequency can be equal to the maximum frequency the chip-level timing specifications will support.

If any of the CME bits are asserted the slow internal reference clock is enabled along with the enabled external clock monitor. For the case when C6[CME0]=1, a loss of clock is detected if the OSC0 external reference falls below a minimum frequency (f_{loc_high} or f_{loc_low} depending on C2[RANGE0]). For the case when C8[CME1]=1, a loss of clock is detected if the RTC external reference falls below a minimum frequency (f_{loc_low}).

NOTE

All clock monitors must be disabled before entering these low-power modes: Stop, VLPS, VLPR, VLPW, LLS, and VLLSx.

On detecting a loss-of-clock event, the MCU generates a system reset if the respective LOCRE bit is set. Otherwise the MCG sets the respective LOCS bit and the MCG generates a LOCS interrupt request.

24.4.5 MCG Auto TRIM (ATM)

The MCG Auto Trim (ATM) is a MCG feature that when enabled, it configures the MCG hardware to automatically trim the MCG Internal Reference Clocks using an external clock as a reference. The selection between which MCG IRC clock gets tested and enabled is controlled by the ATC[ATMS] control bit (ATC[ATMS]=0 selects the 32 kHz IRC and ATC[ATMS]=1 selects the 4 MHz IRC). If 4 MHz IRC is selected for the ATM, a divide by 128 is enabled to divide down the 4 MHz IRC to a range of 31.250 kHz.

When MCG ATM is enabled by writing ATC[ATME] bit to 1, The ATM machine will start auto trimming the selected IRC clock. During the autotrim process, ATC[ATME] will remain asserted and will deassert after ATM is completed or an abort occurs. The MCG ATM is aborted if a write to any of the following control registers is detected : C1, C3, C4, or ATC or if Stop mode is entered. If an abort occurs, ATC[ATMF] fail flag is asserted.

The ATM machine uses the bus clock as the external reference clock to perform the IRC auto-trim. Therefore, it is required that the MCG is configured in a clock mode where the reference clock used to generate the system clock is the external reference clock such as FBE clock mode. The MCG must not be configured in a clock mode where selected IRC ATM clock is used to generate the system clock. The bus clock is also required to be running with in the range of 8–16 MHz.

To perform the ATM on the selected IRC, the ATM machine uses the successive approximation technique to adjust the IRC trim bits to generate the desired IRC trimmed frequency. The ATM SARs each of the ATM IRC trim bits starting with the MSB. For each trim bit test, the ATM uses a pulse that is generated by the ATM selected IRC clock to enable a counter that counts number of ATM external clocks. At end of each trim bit, the ATM external counter value is compared to the ATCV[15:0] register value. Based on the comparison result, the ATM trim bit under test will get cleared or stay asserted. This is done until all trim bits have been tested by ATM SAR machine.

Before the ATM can be enabled, the ATM expected count needs to be derived and stored into the ATCV register. The ATCV expected count is derived based on the required target Internal Reference Clock (IRC) frequency, and the frequency of the external reference clock using the following formula:

$$\text{ATCV Expected Count Value} = 21 * (Fe / Fr)$$

- Fr = Target Internal Reference Clock (IRC) Trimmed Frequency
- Fe = External Clock Frequency

If the auto trim is being performed on the 4 MHz IRC, the calculated expected count value must be multiplied by 128 before storing it in the ATCV register. Therefore, the ATCV Expected Count Value for trimming the 4 MHz IRC is calculated using the following formula.

$$\text{Expected Count Value} = (Fe / Fr) * 21 * (128)$$

24.5 Initialization / Application information

This section describes how to initialize and configure the MCG module in an application.

The following sections include examples on how to initialize the MCG and properly switch between the various available modes.

24.5.1 MCG module initialization sequence

The MCG comes out of reset configured for FEI mode.

The internal reference will stabilize in $t_{irefsts}$ microseconds before the FLL can acquire lock. As soon as the internal reference is stable, the FLL will acquire lock in $t_{fll\_acquire}$ milliseconds.

24.5.1.1 Initializing the MCG

Because the MCG comes out of reset in FEI mode, the only MCG modes that can be directly switched to upon reset are FEE, FBE, and FBI modes (see Figure 24-2). Reaching any of the other modes requires first configuring the MCG for one of these three intermediate modes. Care must be taken to check relevant status bits in the MCG status register reflecting all configuration changes within each mode.

To change from FEI mode to FEE or FBE modes, follow this procedure:

1. Enable the external clock source by setting the appropriate bits in C2 register.
2. Write to C1 register to select the clock mode.
   - If entering FEE mode, set C1[FRDIV] appropriately, clear C1[IREFS] bit to switch to the external reference, and leave C1[CLKS] at 2'b00 so that the output of the FLL is selected as the system clock source.
   - If entering FBE, clear C1[IREFS] to switch to the external reference and change C1[CLKS] to 2'b10 so that the external reference clock is selected as the system clock source. The C1[FRDIV] bits should also be set appropriately here according to the external reference frequency to keep the FLL reference clock in the range of 31.25 kHz to 39.0625 kHz. Although the FLL is bypassed, it is still on in FBE mode.
   - The internal reference can optionally be kept running by setting C1[IRCLKEN]. This is useful if the application will switch back and forth between internal and external modes. For minimum power consumption, leave the internal reference disabled while in an external clock mode.
3. Once the proper configuration bits have been set, wait for the affected bits in the MCG status register to be changed appropriately, reflecting that the MCG has moved into the proper mode.
   - If the MCG is in FEE, FBE, or BLPE mode, and C2[EREFS] was also set in step 1, wait here for S[OSCINIT0] bit to become set indicating that the external clock source has finished its initialization cycles and stabilized.
   - If in FEE mode, check to make sure S[IREFST] is cleared before moving on.
   - If in FBE mode, check to make sure S[IREFST] is cleared and S[CLKST] bits have changed to 2'b10 indicating the external reference clock has been appropriately selected. Although the FLL is bypassed, it is still on in FBE mode.

4. Write to the C4 register to determine the DCO output (MCGFLLCLK) frequency range.
   - By default, with C4[DMX32] cleared to 0, the FLL multiplier for the DCO output is 640. For greater flexibility, if a mid-low-range FLL multiplier of 1280 is desired instead, set C4[DRST_DRS] bits to 2'b01 for a DCO output frequency of 40 MHz. If a mid high-range FLL multiplier of 1920 is desired instead, set the C4[DRST_DRS] bits to 2'b10 for a DCO output frequency of 60 MHz. If a high-range FLL multiplier of 2560 is desired instead, set the C4[DRST_DRS] bits to 2'b11 for a DCO output frequency of 80 MHz.
   - When using a 32.768 kHz external reference, if the maximum low-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b00 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 732 will be 24 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b01 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 1464 will be 48 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b10 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2197 will be 72 MHz.
   - When using a 32.768 kHz external reference, if the maximum high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b11 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2929 will be 96 MHz.
5. Wait for the FLL lock time to guarantee FLL is running at new C4[DRST_DRS] and C4[DMX32] programmed frequency.

To change from FEI clock mode to FBI clock mode, follow this procedure:

1. Change C1[CLKS] bits in C1 register to 2'b01 so that the internal reference clock is selected as the system clock source.

2. Wait for S[CLKST] bits in the MCG status register to change to 2'b01, indicating that the internal reference clock has been appropriately selected.
3. Write to the C2 register to determine the IRCS output (IRCSCLK) frequency range.
    - By default, with C2[IRCS] cleared to 0, the IRCS selected output clock is the slow internal reference clock (32 kHz IRC). If the faster IRC is desired, set C2[IRCS] to 1 for a IRCS clock derived from the 4 MHz IRC source.

24.5.2 Using a 32.768 kHz reference

In FEE and FBE modes, if using a 32.768 kHz external reference, at the default FLL multiplication factor of 640, the DCO output (MCGFLLCLK) frequency is 20.97 MHz at low-range.

If C4[DRST_DRS] bits are set to 2'b01, the multiplication factor is doubled to 1280, and the resulting DCO output frequency is 41.94 MHz at mid-low-range. If C4[DRST_DRS] bits are set to 2'b10, the multiplication factor is set to 1920, and the resulting DCO output frequency is 62.91 MHz at mid high-range. If C4[DRST_DRS] bits are set to 2'b11, the multiplication factor is set to 2560, and the resulting DCO output frequency is 83.89 MHz at high-range.

In FBI and FEI modes, setting C4[DMX32] bit is not recommended. If the internal reference is trimmed to a frequency above 32.768 kHz, the greater FLL multiplication factor could potentially push the microcontroller system clock out of specification and damage the part.

24.5.3 MCG mode switching

When switching between operational modes of the MCG, certain configuration bits must be changed in order to properly move from one mode to another.

Each time any of these bits are changed (C1[IREFS], C1[CLKS], C2[IRCS], or C2[EREFS], the corresponding bits in the MCG status register (IREFST, CLKST, IRCST, or OSCINIT) must be checked before moving on in the application software.

Additionally, care must be taken to ensure that the reference clock divider (C1[FRDIV]) is set properly for the mode being switched to. For instance, in FEE mode, if using a 4MHz crystal, C1[FRDIV] must be set to 3'b010 (divide-by-128) to devide the external frequency down to the required frequency between 31.25 and 39.0625 kHz.

In FBE, FEE, FBI, and FEI modes, at any time, the application can switch the FLL multiplication factor between 640, 1280, 1920, and 2560 with C4[DRST_DRS] bits. Writes to C4[DRST_DRS] bits will be ignored if C2[LP]=1.

The table below shows MCGOUTCLK frequency calculations using C1[FRDIV]settings for each clock mode.

Table 24-2. MCGOUTCLK Frequency Calculation Options

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	f_{int} × F	Typical $f_{MCGOUTCLK}$ = 21 MHz immediately after reset.
FEE (FLL engaged external)	(f_{ext} / FLL_R) × F	f_{ext} / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	OSCCLK	OSCCLK / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	MCGIRCLK	Selectable between slow and fast IRC
BLPI (Bypassed low power internal)	MCGIRCLK	Selectable between slow and fast IRC
BLPE (Bypassed low power external)	OSCCLK	

1. FLL_R is the reference divider selected by the C1[FRDIV] bits, F is the FLL factor selected by C4[DRST_DRS] and C4[DMX32] bits .

This section will include several mode switching examples, using an MHz external crystal..

Chapter 25
32 kHz Oscillator (32kRTC)

The 32 kHz RTC module is a crystal oscillator. The module, in conjunction with an external crystal, generates a 32kHz clock for the MCU with very low power.

25.1 Introduction

The RTC oscillator module provides the clock source for the RTC. The RTC oscillator module, in conjunction with an external crystal, generates a reference clock for the RTC.

25.1.1 Features and Modes

The key features of the RTC oscillator are as follows:

- Supports 32 kHz crystals with very low power
- Consists of internal feed back resistor
- Consists of internal programmable capacitors as the C_{load} of the oscillator
- Automatic Gain Control (AGC) to optimize power consumption

The RTC oscillator operations are described in detail in Functional Description .

25.1.2 Block Diagram

The following is the block diagram of the RTC oscillator.

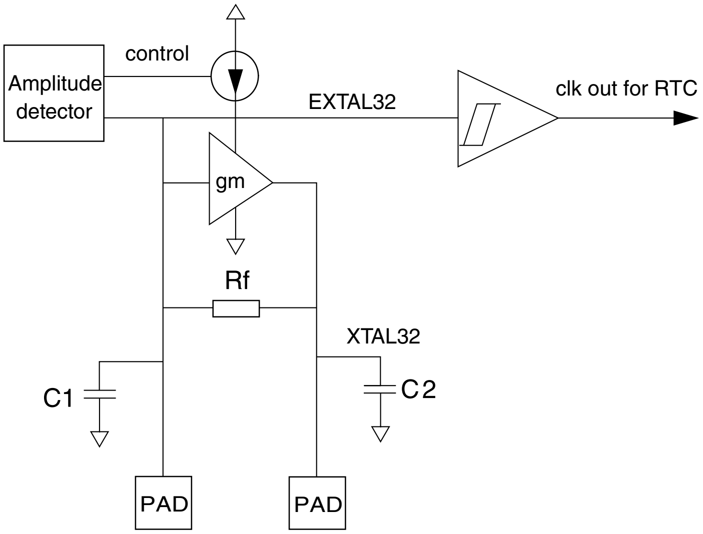


Figure 25-1. RTC Oscillator Block Diagram

25.2 RTC Signal Descriptions

The following table shows the user-accessible signals available for the RTC oscillator. See the chip-level specification to find out which signals are actually connected to the external pins.

Table 25-1. RTC Signal Descriptions

Signal	Description	I/O
EXTAL32	Oscillator Input	I
XTAL32	Oscillator Output	O

25.2.1 EXTAL32 — Oscillator Input

This signal is the analog input of the RTC oscillator.

25.2.2 XTAL32 — Oscillator Output

This signal is the analog output of the RTC oscillator module.

25.3 External Crystal Connections

The connections with a crystal is shown in the following figure. External load capacitors and feedback resistor are not required.

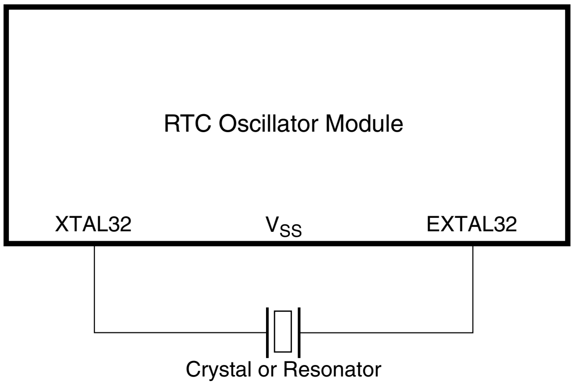


Figure 25-2. Crystal Connections

25.4 Memory Map/Register Descriptions

RTC oscillator control bits are part of the RTC registers. Refer to RTC Control Register (RTC_CR) , or RTC_GP_DATA_REG in the chip-specific information section, for more details.

25.5 Functional Description

As shown in Figure 25-1, the module includes an amplifier which supplies the negative resistor for the RTC oscillator. The gain of the amplifier is controlled by the amplitude detector, which optimizes the power consumption. A schmitt trigger is used to translate the sine-wave generated by this oscillator to a pulse clock out, which is a reference clock for the RTC digital core.

The oscillator includes an internal feedback resistor of approximately 100 MΩ between EXTAL32 and XTAL32.

In addition, there are two programmable capacitors with this oscillator, which can be used as the Cload of the oscillator. The programmable range is from 0pF to 30pF.

25.6 Reset Overview

There is no reset state associated with the RTC oscillator.

25.7 Interrupts

The RTC oscillator does not generate any interrupts.

Chapter 26
Flash Memory Controller (FMC)

26.1 Introduction

The Flash Memory Controller (FMC) is a memory acceleration unit. A list of features provided by the FMC can be found here.

- an interface between bus masters and the 32-bit program flash memory.
- a buffer and a cache that can accelerate program flash memory data transfers.

26.1.1 Overview

The Flash Memory Controller manages the interface between bus masters and the 32-bit program flash memory. The FMC receives status information detailing the configuration of the flash memory and uses this information to ensure a proper interface. The FMC supports 8-bit, 16-bit, and 32-bit read operations from the program flash memory. A write operation to program flash memory results in a bus error.

In addition, the FMC provides two separate mechanisms for accelerating the interface between bus masters and program flash memory. A 32-bit speculation buffer can prefetch the next 32-bit flash memory location, and a 4-way, 4-set program flash memory cache can store previously accessed program flash memory data for quick access times.

26.1.2 Features

The features of FMC module include:

- Interface between bus masters and the 32-bit program flash memory:
  - 8-bit, 16-bit, and 32-bit read operations to nonvolatile flash memory.
- Acceleration of data transfer from the program flash memory to the device:

- 32-bit prefetch speculation buffer for program flash accesses with controls for instruction/data access
- 4-way, 4-set, 32-bit line size program flash memory cache for a total of sixteen 32-bit entries with invalidation control

26.2 Modes of operation

The FMC operates only when a bus master accesses the program flash memory.

In terms of chip power modes:

- The FMC operates only in Run and Wait modes, including VLPR and VLPW modes.
- For any power mode where the program flash memory cannot be accessed, the FMC is disabled.

26.3 External signal description

The FMC has no external (off-chip) signals.

26.4 Memory map and register descriptions

The MCM's programming model provides control and configuration of the FMC's features.

For details, see the description of the MCM's Platform Control Register (PLACR).

26.5 Functional description

The FMC is a flash acceleration unit with flexible buffers for user configuration.

Besides managing the interface between bus masters and the program flash memory, the FMC can be used to customize the program flash memory cache and buffer to provide single-cycle system clock data access times. Whenever a hit occurs for the prefetch speculation buffer or the cache (when enabled), the requested data is transferred within a single system clock.

Upon system reset, the FMC is configured as follows:

- Flash cache is enabled.
- Instruction speculation and caching are enabled.

- Data speculation is disabled.
- Data caching is enabled.

Though the default configuration provides flash acceleration, advanced users may desire to customize the FMC buffer configurations to maximize throughput for their use cases. For example, the user may adjust the controls to enable buffering per access type (data or instruction).

NOTE

When reconfiguring the FMC, do not program the control and configuration inputs to the FMC while the program flash memory is being accessed. Instead, change them with a routine executing from RAM in supervisor mode.

26.5.1 Flash Access Control (FAC) Function

The Flash Access Control (FAC) is a configurable memory protection scheme optimized to allow end users to use software libraries while offering programmable restrictions to these libraries. The flash memory is divided into *equal size segments* that provide protection to proprietary software libraries. The protection of these segments is controlled: the FAC provides a cycle-by-cycle evaluation of the access rights for each transaction routed to the on-chip flash memory. Two levels of vendors can add their proprietary software to a device; FAC protection of segments for each level are defined once, using the PGMONCE command.

Flash access control aligns to the 3 privilege levels supported by ARM Cortex-M family products:

- Most secure state is supervisor/privileged secure: allows execute-only and provides supervisor-only access control.
- Mid-level state is execute-only.
- Unsecure state is where no access control states are set.

Features:

- Lightweight access control logic for on-chip flash memory
- Flash address space divided into (32 or 64) equal-sized segments (segment size is defined as flash_size [bytes]/(32 or 64))
- Separate control bits for supervisor-only access and execute-only access per segment
- Access control evaluated on each bus cycle routed to the flash
- Access violation errors terminate the bus cycle and return zeroes for read data
- Programming model allows 2 levels of protected segments

26.5.1.1 Memory map and register definitions

The following table shows the mapping of FAC registers. Descriptions of each register and its bit assignments follow.

- The Flash Management Unit (FMU) supports access to its FAC programming model via a 32-bit slave peripheral bus connection.
- Unimplemented register bits read as zero.
- For implementations supporting only 32 segments, only the 32-bit "low" register is implemented.
- Writes to any read-only or reserved registers are ignored; attempts to access flash register space above offset '2B' will generate a transfer error.
- The terms *supervisor* and *user* modes are equivalent to *privileged* and *unprivileged* modes.
- In this FAC section, *n* refers to the segment number, and *x* is the acronym of the module that the registers are in (which sometimes varies from one device to another).

x memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1C	Execute-only Access Register (x_XACC)	32	R	See section	26.5.1.1.1/ 440
24	Supervisor-only Access Register (x_SACC)	32	R	See section	26.5.1.1.2/ 441
28	Configuration Register (x_CR)	32	R	See section	26.5.1.1.3/ 442

26.5.1.1.1 Execute-only Access Register (x_XACC)

The XACC register provides a bit map for the flash segments to allow execute only or both data and instruction fetches for each associated segment. By definition, execute-only accesses include instruction fetches or PC-relative data loads from the processor.

During the reset sequence the XACC register is loaded with a pre-programmed value from non-volatile space in flash. For more about NVM characteristics, see the functional description. Any change made to an NVM location takes effect on the next system reset. The flash basis for the values is signified by * in the reset value.

Address: 0h base + 1Ch offset = 1Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XA[31:0]																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Pre-programmed flash valuex = Undefined at reset.

x_XACC field descriptions

Field	Description
XA[31:0]	Execute-only Access Control for segments 31-0 0 Associated segment is accessible in execute mode only (as an instruction fetch) 1 Associated segment is accessible as data or in execute mode

26.5.1.1.2 Supervisor-only Access Register (x_SACC)

The SACC register provides a bit map for the flash segments to allow *supervisor only* or *user* and *supervisor* access to the associated segment.

During the reset sequence the SACC register is loaded with a pre-programmed value from non-volatile space in flash. For more about NVM characteristics, see the functional description. Any change made to an NVM location takes effect on the next system reset. The flash basis for the values is signified by * in the reset value.

Address: 0h base + 24h offset = 24h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SA[31:0]																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Pre-programmed flash valuex = Undefined at reset.

x_SACC field descriptions

Field	Description
SA[31:0]	Supervisor Access for segments 31-0 0 Associated segment is accessible in supervisor mode only 1 Associated segment is accessible in user or supervisor mode

26.5.1.1.3 Configuration Register (x_CR)

The FAC Configuration Register provides basic configuration information including the flash segment size and an indicator of segment divisions.

The NUMSG and SGSIZE values are fixed for a device. The chip-specific basis for the values is signified by * in the reset value.

Address: 0h base + 28h offset = 28h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NUMSG								0																SGSIZE							
W																																
Reset	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*

* Notes:
- NUMSG field: Device specific value
- SGSIZE field: Device specific value

x_CR field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–24
NUMSG</td><td>Number of Segments Indicator
The NUMSG field indicates the number of equal-sized segments in the flash.
0x20 Flash is divided into 32 segments
0x40 Flash is divided into 64 segments</td></tr>
<tr><td>23–8
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>SGSIZE</td><td>Segment Size
The segment size is a fixed value calculated from the available flash size (rounded up to nearest power of 2) divided by 32 or 64, depending on the amount of available program flash. This field determines which bits in the address are used to index into the <i>x</i>_SACC and <i>x</i>_XACC bitmaps to get the appropriate permission flags. The segment size is defined by the equation 2^(8 + SGSIZE[7:0]). The tables below show a sampling of possible settings.
<table>
<tr><th>Flash Size</th><th>Segment Size</th><th>Segment Size Encoding</th></tr>
<tr><td colspan="3"><b>32 Segment Encodings</b></td></tr>
<tr><td>16 KBytes</td><td>16 KBytes/32 = 512 Bytes</td><td>0x1</td></tr>
<tr><td>32 KBytes</td><td>32 KBytes/32 = 1 KBytes</td><td>0x2</td></tr>
<tr><td>64 KBytes</td><td>64 KBytes/32 = 2 KBytes</td><td>0x3</td></tr>
<tr><td>128 KBytes</td><td>128 KBytes/32 = 4 KBytes</td><td>0x4</td></tr>
<tr><td colspan="3"><b>64 Segment Encodings</b></td></tr>
<tr><td>256 KBytes</td><td>256 KBytes/64 = 4 KBytes</td><td>0x4</td></tr>
<tr><td>512 KBytes</td><td>512 KBytes/64 = 8 KBytes</td><td>0x5</td></tr>
<tr><td>1 MBytes</td><td>1 MBytes/64 = 16 KBytes</td><td>0x6</td></tr>
</table>
</td></tr>
</table>

x_CR field descriptions (continued)

Field	Description

26.5.1.2 FAC functional description

The access control functionality is implemented in 2 separate blocks within the SoC. The Flash Management Unit (FMU) includes non-volatile configuration information that is retrieved during reset and and sent to the platform to control access to the flash array during normal operation.

There are (4) 32-bit NVM storage locations to support access control features. These NVM locations are summarized in the table below.

Table 26-1. NVM Locations

NVM location	Description	
NVSACC1, NVSACC2	Two locations are ANDed together and loaded during reset into the *x*_SACC register to provide access configuration.	Segment-wise control for supervisor-only access vs. supervisor and user access
NVXACC1, NVXACC2	Two locations are ANDed together and loaded during reset into the *x*_XACC register to provide access configuration.	Segment-wise control for execute-only vs. data and execute

Each of these NVM locations is programmable through a Program Once flash command and can be programmed one time. These NVM locations are unaffected by Erase All Blocks flash command and debug interface initiated mass erase operations. Since the 2 NVXACCx fields are ANDed, the access protection can only be increased. A segment's access controls can be changed from data read and execute (XA*n* =1) to execute-only (XA*n* =0), or from supervisor and user mode (SAn = 1) to supervisor-only mode (SAn = 0).

The flash is released from reset early while the core continues to be held in reset. The FMU captures the NVM access control information in internal registers. The FMU ANDs the multiple execute-only fields to create a single execute-only field. This execute-only field driven to the platform is static prior to the core being released from reset. The supervisor-only field is handled in the same manner.

The FMU includes the FAC registers that provide control access to the flash address space. During the address phase of every attempted flash transfer, the supervisor access (SA*n*) and execute access (XA*n*) bits are examined to either allow or deny access. If access is denied, then the access is aborted and terminates with a bus error; the read data is also zeroed.

The next table shows segment assignments relative to the flash location.

Table 26-2. Flash Protection Ranges

SA*n* and XA*n* Bit	Protected Segment Address Range	Segment Size (Fraction of total Flash)
64 Segment Encodings		
0	0x0_0000_0000 – (Flash_size/64-1)	1/64
1	(Flash_size/64) – 2*(Flash_size/64-1)	1/64
....		
63	63*(Flash_size/64) – 62*(Flash_size/64-1)	1/64
32 Segment Encodings		
0	0x0_0000_0000 – (Flash_size/32-1)	1/32
1	(Flash_size/32) – 2*(Flash_size/32-1)	1/32
....		
31	31*(Flash_size/32) – 30*(Flash_size/32-1)	1/32

Individual segments within the flash memory can be independently protected from user access and data access. Protection is controlled by the individual bits within the *x*_SACC and *x*_XACC registers, as shown in the next figure.

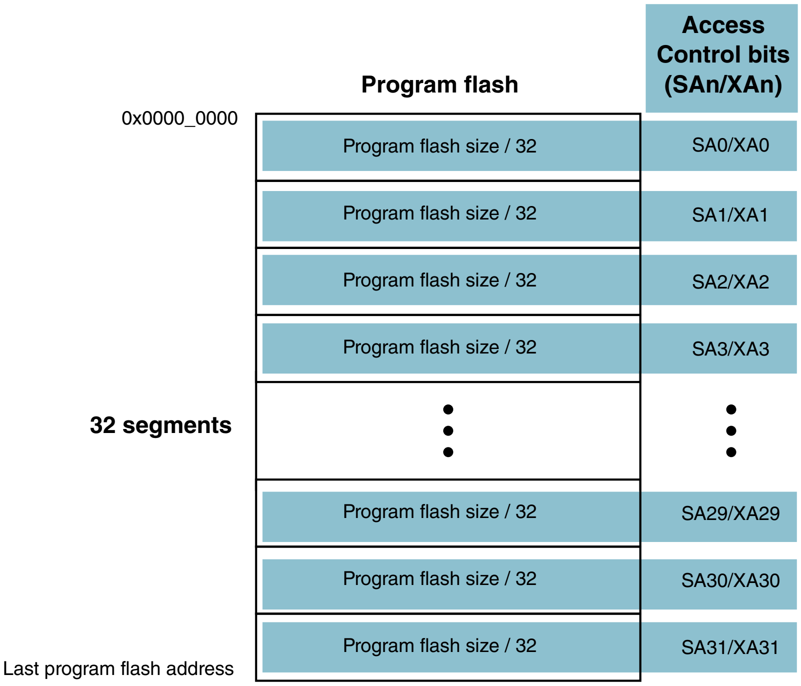


Figure 26-1. Program flash protection (32 segments)

26.5.1.2.1 Interface Signals

Table 26-3. Interface Signals

Signal	Width	From	To	Description
xacc	64 or 32	FMU	Platform	Direct xacc (execute-only access control) register
sacc	64 or 32	FMU	Platform	Direct sacc (supervisor access control) register
numsg	8	FMU	Platform	NUMSG bit field - Binary encoded number of segments 0x40 for 64 segments 0x20 for 32 segments
fac_enable	1	SIM	FMU	SIM Option bit - derived from an IFR bit and captured in SIM_SOPTx. A way to disable the flash access control for phantom devices without this feature. fac_enable==1 - Access Control feature is enabled fac_enable==0 - Access Control feature is disabled • During the reset sequence, XACC registers are written to all "1"s. • During the reset sequence, SACC registers are written to all 1"s. • Implied protection based on XACC registers is turned off.

26.5.1.2.2 Flash Command Impact

Program Longword/Phrase/ Section	If the targeted flash location is in an execute-only protected segment, then these program commands are not allowed unless a Read 1s All Blocks command is executed and returns with a pass code (which means the part has been fully erased). After the Read 1s All Blocks command is executed with a pass code returned, then the protected segment is open to program commands. To close off programmability to execute-only spaces once again, the device must be reset or a Read 1s All Blocks command is executed with a fail result. Attempts to program in a protected segment *when not open to program commands* causes a Protection Violation flag.
PGMCHK	The FMU will not execute the PGMCHK on a segment that has been configured as execute-only. The Flash Protection Violation flag is set if an attempt is made to execute PGMCHK command on an execute-only address.
Erase Flash Sector	If the targeted flash sector is in an execute-only protected segment, then the Erase Flash Sector command is not allowed, and sets the Protection Violation flag. The only means of erasing protected space is by an Erase All operation.
ERSALL	The Erase All Blocks command is not affected by Access Control. An Erase All Blocks command will erase any libraries that have been programmed in any execute-only segment. The programmed execute-only assignment is not erased as part of the Erase All Blocks command, and access control regions remain as previously programmed. **NOTE:** The ERSALL command may be used for field upgrades. Access control states remain programmed. Software must plan accordingly, possibly making extra space available for future use.
SWAP	A new control has been added to the SWAP command to disable the SWAP feature. The IFR SWAP Field and the SWAP indicator are erased during the Erase All Blocks command operation, resulting in the SWAP system being uninitialized. The SWAP command must be run with the initialization code to set the SWAP indicator address and initialize the SWAP system.

	After being disabled, SWAP cannot be enabled without doing an Erase All. An Erase All erases preloaded code and libraries in the flash array and resets the SWAP system back to uninitialized, but leaves the access controls as previously programmed. If SWAP is intended to be disabled as part of the access control protection, then the disabled setting must be restored after an Erase All Blocks operation.

26.5.1.2.3 Core Platform Impact

Platform core caches (Flash and LMEM caches)	If any segment is marked as *execute-only*, then the caches are hidden from the user. The tag is read-only and cannot be written, and the data caches cannot be read or written. Writes to the tag and data arrays are ignored, and reads of the data array return 0's. This will impact debug breakpoints. See the debug section for details.
Debug	The debugger is a non-processor bus master and cannot step, trace or break in execute-only regions. In supervisor-only mode, the debugger is restricted from changing modes. Debug accesses to any segment of flash space marked as execute-only also terminate with a bus error.
PC-relative addressing	The PC-relative addressing issue is still being understood and this section will be updated in the future. PC relative re-entry to execute-only segments will be allowed......... Restrictions will be placed on software for PC relative addressing, because hardware cannot determine if PC relative data references are crossing segment boundaries. • If ifetch is executing in a protected segment, then data references will be allowed. • Hardware cannot track speculative ifetches across boundaries.
Interrupts	If function calls are used to move into an execute-only segment, then this can be tracked by hardware when typical software controls are used (i.e., saving registers and states before executing new code).
Reset Vector	In the ARM core, the reset vector fetch is supervisor data, which poses issues if the reset vector is located in a segment marked execute-only. Additional logic has been implemented to allow supervisor data fetches to execute-only spaces, after reset until the first valid instruction fetch. After the first valid instruction fetch, the FAC logic follows normal checks.

26.5.1.2.4 Software Impact

As implementation, verification and validation continue, there will be more details on software impact that will need to be communicated to tool and library vendors. The hardware cannot see all states of the ARM core and cannot track the software flow requiring software restrictions to work with the hardware for a robust solution.

Any segment marked as execute-only can see all code in the system. This means one execute-only segment can read the execute-only code in another segment. Therefore, if Freescale is sending pre-loaded code to another vendor, then that vendor will have access to Freescale code. Possibly use NDAs and legal agreements to deal with this issue.

For single pre-loads (for example, if Freescale is pre-loading for a GP market or if a vendor with a blank part is pre-loading proprietary code), then both levels of access control must be programmed, to protect the pre-loaded code.

If any portion of a protected segment is not used by pre-loaded code, then it (the portion of a protected segment that is not used by pre-loaded code) should be programmed with NOPs, to prevent additional code from being programmed in that segment by hackers.

26.5.1.2.5 Access Check Evaluation

The flash controller FAC provides a cycle-by-cycle evaluation of the access rights for each data transaction routed to the on-chip flash memory.

The entire flash storage capacity is partitioned into equal sized segments. Two registers include a supervisor-only access control indicator and a execute-only access control indicator for each segment.

The FAC logic performs the required access control evaluation using the reference address and a 2-bit attribute (or "protection" field) as inputs from the bus cycle plus the contents of the programming model registers.

The following code example illustrates C code for FAC evaluation:

```
unsigned long long sacc;  // supervisor-only map
unsigned long long xacc;  // execute-only map
unsigned int seg_size;    // 8-bit segment size
unsigned int fac_error;

fac_evaluation (addr, prot)
  unsigned int addr;  // access address
  unsigned int hprot; // encoded 2-bit "protection" field {supv, data}
{
  unsigned int sacc_flag; // sacc flag for this segment
  unsigned int xacc_flag; // xacc flag for this segment
  unsigned int i;         // segment index

  i         = (addr >> (8 + seg_size & 0x0f)) & 0x3f; // form 6-bit segment index
  sacc_flag = (sacc >> i) & 1;              // extract sacc bit for this segment
  xacc_flag = (xacc >> i) & 1;              // extract xacc bit for this segment

  // create a 4-tuple concatenating the 2-bit protection field + {sacc, xacc} flags

  switch ((hprot & 3) << 2 | (sacc_flag << 1) | xacc_flag) {
  // all these combinations are allowed accesses
    case 0x2: // {user, ifetch} && {supv+user, ifetch-only}
    case 0x3: // {user, ifetch} && {supv+user, ifetch+data}
    case 0x7: // {user, data}   && {supv+user, ifetch+data}
    case 0x8: // {supv, ifetch} && {supv-only, ifetch-only}
    case 0x9: // {supv, ifetch} && {supv-only, ifetch+data}
    case 0xa: // {supv, ifetch} && {supv+user, ifetch-only}
    case 0xb: // {supv, ifetch} && {supv+user, ifetch+data}
    case 0xd: // {supv, data}   && {supv-only, ifetch+data}
    case 0xf: // {supv, data}   && {supv+user, ifetch+data}
      fac_error = 00;
      break;

  // all these combinations are unallowed, that is, errored accesses
    case 0x0: // {user, ifetch} && {supv-only, ifetch-only}
    case 0x1: // {user, ifetch} && {supv-only, ifetch+data}
    case 0x4: // {user, data}   && {supv-only, ifetch-only}
    case 0x5: // {user, data}   && {supv-only, ifetch+data}
    case 0x6: // {user, data}   && {supv+user, ifetch-only}
    case 0xc: // {supv, data}   && {supv-only, ifetch-only}
    case 0xe: // {supv, data}   && {supv+user, ifetch-only}
      fac_error = 1;
      break;

   } // switch()

} // fac_evaluation()
```

26.5.1.2.6 FAC application tips

In one use case, the NVSACC1 and NVXACC1 locations are programmed by Freescale and they protect Freescale libraries that have been programmed into associated flash segments in a device. Later, the NVSACC2 and NVXACC2 NVM locations can optionally be programmed by a third-party vendor who wants to program their proprietary software and to extend the protection of protected flash segments to include their software libraries before supplying it all to their customers.

Their customer would then develop their own code to use the available libraries, and program their code into the remaining available on-chip flash. The device continues to support the end user with standard security features that further limit external access to flash resources.

SWAP: If execute-only code is mirrored in both halves of the flash array, then SWAP can be enabled without any issues; otherwise SWAP should be disabled, because hardware does not track access control addressing during SWAP.

Chapter 27
Flash Memory Module (FTFA)

27.1 Introduction

The flash memory module includes the following accessible memory regions:

- Program flash memory for vector space and code store

Flash memory is ideal for single-supply applications, permitting in-the-field erase and reprogramming operations without the need for any external high voltage power sources.

The flash memory module includes a memory controller that executes commands to modify flash memory contents. An erased bit reads '1' and a programmed bit reads '0'. The programming operation is unidirectional; it can only move bits from the '1' state (erased) to the '0' state (programmed). Only the erase operation restores bits from '0' to '1'; bits cannot be programmed from a '0' to a '1'.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

The standard shipping condition for flash memory is erased with security disabled. Data loss over time may occur due to degradation of the erased ('1') states and/or programmed ('0') states. Therefore, it is recommended that each flash block or sector be re-erased immediately prior to factory programming to ensure that the full data retention capability is achieved.

27.1.1 Features

The flash memory module includes the following features.

NOTE

> See the device's Chip Configuration details for the exact amount of flash memory available on your device.

27.1.1.1 Program Flash Memory Features

- Sector size of 1 KB
- Program flash protection scheme prevents accidental program or erase of stored data
- Program flash access control scheme prevents unauthorized access to selected code segments
- Automated, built-in, program and erase algorithms with verify
- Read access to one program flash block is possible while programming or erasing data in the other program flash block

27.1.1.2 Other Flash Memory Module Features

- Internal high-voltage supply generator for flash memory program and erase operations
- Optional interrupt generation upon flash command completion
- Supports MCU security mechanisms which prevent unauthorized access to the flash memory contents

27.1.2 Block Diagram

The block diagram of the flash memory module is shown in the following figure.

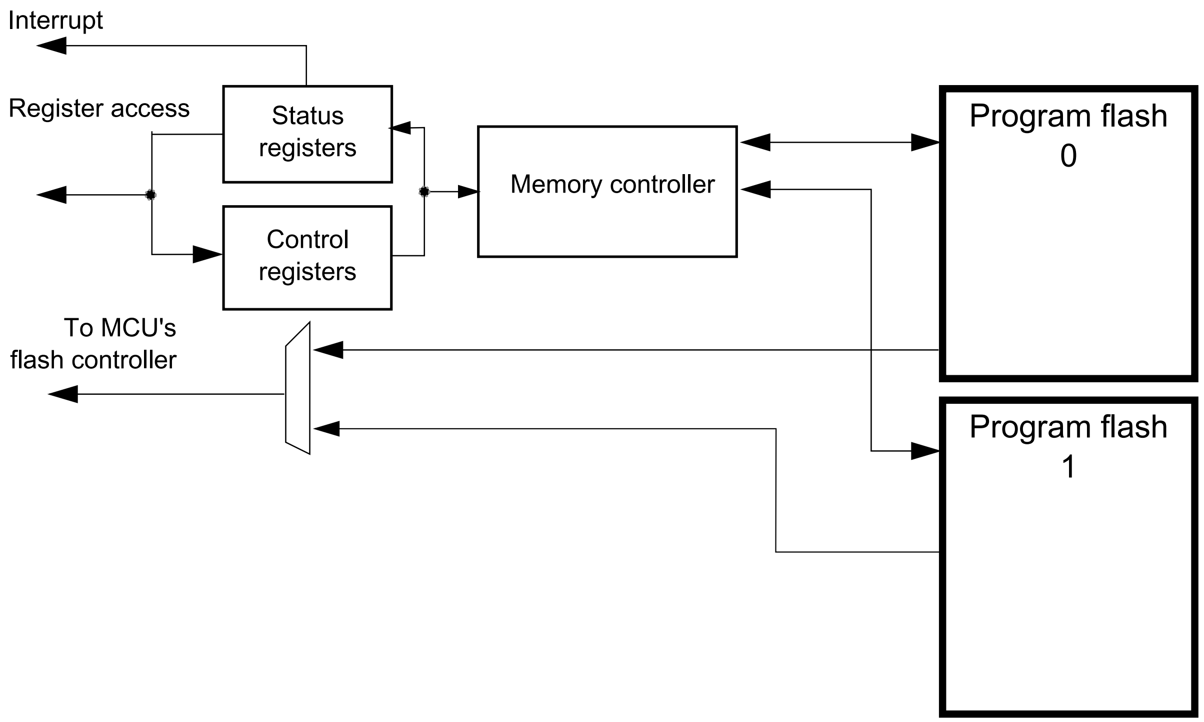


Figure 27-1. Flash Block Diagram

27.1.3 Glossary

Command write sequence — A series of MCU writes to the flash FCCOB register group that initiates and controls the execution of flash algorithms that are built into the flash memory module.

Endurance — The number of times that a flash memory location can be erased and reprogrammed.

FCCOB (Flash Common Command Object) — A group of flash registers that are used to pass command, address, data, and any associated parameters to the memory controller in the flash memory module.

Flash block — A macro within the flash memory module which provides the nonvolatile memory storage.

Flash Memory Module — All flash blocks plus a flash management unit providing high-level control and an interface to MCU buses.

IFR — Nonvolatile information register found in each flash block, separate from the main memory array.

Longword — 32 bits of data with an aligned longword having byte-address[1:0] = 00.

NVM — Nonvolatile memory. A memory technology that maintains stored data during power-off. The flash array is an NVM using NOR-type flash memory technology.

NVM Normal Mode — An NVM mode that provides basic user access to flash memory module resources. The CPU or other bus masters initiate flash program and erase operations (or other flash commands) using writes to the FCCOB register group in the flash memory module.

Program flash — The program flash memory provides nonvolatile storage for vectors and code store.

Program flash Sector — The smallest portion of the program flash memory (consecutive addresses) that can be erased.

Retention — The length of time that data can be kept in the NVM without experiencing errors upon readout. Since erased (1) states are subject to degradation just like programmed (0) states, the data retention limit may be reached from the last erase operation (not from the programming time).

RWW— Read-While-Write. The ability to simultaneously read from one memory resource while commanded operations are active in another memory resource.

Secure — An MCU state conveyed to the flash memory module as described in the Chip Configuration details for this device. In the secure state, reading and changing NVM contents is restricted.

Word — 16 bits of data with an aligned word having byte-address[0] = 0.

27.2 External Signal Description

The flash memory module contains no signals that connect off-chip.

27.3 Memory Map and Registers

This section describes the memory map and registers for the flash memory module.

Data read from unimplemented memory space in the flash memory module is undefined. Writes to unimplemented or reserved memory space (registers) in the flash memory module are ignored.

27.3.1 Flash Configuration Field Description

The program flash memory contains a 16-byte flash configuration field that stores default protection settings (loaded on reset) and security information that allows the MCU to restrict access to the flash memory module.

Flash Configuration Field Offset Address	Size (Bytes)	Field Description
0x0_0400–0x0_0407	8	Backdoor Comparison Key. Refer to Verify Backdoor Access Key Command and Unsecuring the Chip Using Backdoor Key Access.
0x0_0408–0x0_040B	4	Program flash protection bytes. Refer to the description of the Program Flash Protection Registers (FPROT0-3).
0x0_040F	1	Reserved
0x0_040E	1	Reserved
0x0_040D	1	Flash nonvolatile option byte. Refer to the description of the Flash Option Register (FOPT).
0x0_040C	1	Flash security byte. Refer to the description of the Flash Security Register (FSEC).

27.3.2 Program Flash IFR Map

The program flash IFR is nonvolatile information memory that can be read freely, but the user has no erase and limited program capabilities (see the Read Once, Program Once, and Read Resource commands in Read Once Command, Program Once Command and Read Resource Command).

The contents of the program flash IFR are summarized in the table found here and further described in the subsequent paragraphs.

The program flash IFR is located within the program flash 0 memory block .

Address Range	Size (Bytes)	Field Description
0x00 – 0x9F	160	Reserved
0xA0 – 0xA3	4	Program Once XACCH-1 Field (index = 0x10)
0xA4 – 0xA7	4	Program Once XACCL-1 Field (index = 0x10)
0xA8 – 0xAB	4	Program Once XACCH-2 Field (index = 0x11)

Table continues on the next page...

Address Range	Size (Bytes)	Field Description
0xAC – 0xAF	4	Program Once XACCL-2 Field (index = 0x11)
0xB0 – 0xB3	4	Program Once SACCH-1 Field (index = 0x12)
0xB4 – 0xB7	4	Program Once SACCL-1 Field (index = 0x12)
0xB8 – 0xBB	4	Program Once SACCH-2 Field (index = 0x13)
0xBC – 0xBF	4	Program Once SACCL-2 Field (index = 0x13)
0xC0 – 0xFF	64	Program Once ID Field (index = 0x00 - 0x0F)

27.3.2.1 Program Once Field

The Program Once Field in the program flash IFR provides 96 bytes of user data storage separate from the program flash main array. The user can program the Program Once Field one time only as there is no program flash IFR erase mechanism available to the user. The Program Once Field can be read any number of times. This section of the program flash IFR is accessed in 4-byte or 8-Byte records using the Read Once and Program Once commands (see Read Once Command and Program Once Command).

27.3.3 Register Descriptions

The flash memory module contains a set of memory-mapped control and status registers.

NOTE

While a command is running (FSTAT[CCIF]=0), register writes are not accepted to any register except FCNFG and FSTAT. The no-write rule is relaxed during the start-up reset sequence, prior to the initial rise of CCIF. During this initialization period the user may write any register. All register writes are also disabled (except for registers FCNFG and FSTAT) whenever an erase suspend request is active (FCNFG[ERSSUSP]=1).

FTFA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Flash Status Register (FTFA_FSTAT)	8	R/W	00h	27.3.3.1/ 456
4002_0001	Flash Configuration Register (FTFA_FCNFG)	8	R/W	00h	27.3.3.2/ 458
4002_0002	Flash Security Register (FTFA_FSEC)	8	R	Undefined	27.3.3.3/ 459
4002_0003	Flash Option Register (FTFA_FOPT)	8	R	Undefined	27.3.3.4/ 460
4002_0004	Flash Common Command Object Registers (FTFA_FCCOB3)	8	R/W	00h	27.3.3.5/ 461
4002_0005	Flash Common Command Object Registers (FTFA_FCCOB2)	8	R/W	00h	27.3.3.5/ 461
4002_0006	Flash Common Command Object Registers (FTFA_FCCOB1)	8	R/W	00h	27.3.3.5/ 461
4002_0007	Flash Common Command Object Registers (FTFA_FCCOB0)	8	R/W	00h	27.3.3.5/ 461
4002_0008	Flash Common Command Object Registers (FTFA_FCCOB7)	8	R/W	00h	27.3.3.5/ 461
4002_0009	Flash Common Command Object Registers (FTFA_FCCOB6)	8	R/W	00h	27.3.3.5/ 461
4002_000A	Flash Common Command Object Registers (FTFA_FCCOB5)	8	R/W	00h	27.3.3.5/ 461
4002_000B	Flash Common Command Object Registers (FTFA_FCCOB4)	8	R/W	00h	27.3.3.5/ 461
4002_000C	Flash Common Command Object Registers (FTFA_FCCOBB)	8	R/W	00h	27.3.3.5/ 461
4002_000D	Flash Common Command Object Registers (FTFA_FCCOBA)	8	R/W	00h	27.3.3.5/ 461
4002_000E	Flash Common Command Object Registers (FTFA_FCCOB9)	8	R/W	00h	27.3.3.5/ 461
4002_000F	Flash Common Command Object Registers (FTFA_FCCOB8)	8	R/W	00h	27.3.3.5/ 461
4002_0010	Program Flash Protection Registers (FTFA_FPROT3)	8	R/W	Undefined	27.3.3.6/ 462
4002_0011	Program Flash Protection Registers (FTFA_FPROT2)	8	R/W	Undefined	27.3.3.6/ 462
4002_0012	Program Flash Protection Registers (FTFA_FPROT1)	8	R/W	Undefined	27.3.3.6/ 462
4002_0013	Program Flash Protection Registers (FTFA_FPROT0)	8	R/W	Undefined	27.3.3.6/ 462
4002_0018	Execute-only Access Registers (FTFA_XACCH3)	8	R	Undefined	27.3.3.7/ 464
4002_0019	Execute-only Access Registers (FTFA_XACCH2)	8	R	Undefined	27.3.3.7/ 464

Table continues on the next page...

FTFA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_001A	Execute-only Access Registers (FTFA_XACCH1)	8	R	Undefined	27.3.3.7/ 464
4002_001B	Execute-only Access Registers (FTFA_XACCH0)	8	R	Undefined	27.3.3.7/ 464
4002_001C	Execute-only Access Registers (FTFA_XACCL3)	8	R	Undefined	27.3.3.7/ 464
4002_001D	Execute-only Access Registers (FTFA_XACCL2)	8	R	Undefined	27.3.3.7/ 464
4002_001E	Execute-only Access Registers (FTFA_XACCL1)	8	R	Undefined	27.3.3.7/ 464
4002_001F	Execute-only Access Registers (FTFA_XACCL0)	8	R	Undefined	27.3.3.7/ 464
4002_0020	Supervisor-only Access Registers (FTFA_SACCH3)	8	R	Undefined	27.3.3.8/ 465
4002_0021	Supervisor-only Access Registers (FTFA_SACCH2)	8	R	Undefined	27.3.3.8/ 465
4002_0022	Supervisor-only Access Registers (FTFA_SACCH1)	8	R	Undefined	27.3.3.8/ 465
4002_0023	Supervisor-only Access Registers (FTFA_SACCH0)	8	R	Undefined	27.3.3.8/ 465
4002_0024	Supervisor-only Access Registers (FTFA_SACCL3)	8	R	Undefined	27.3.3.8/ 465
4002_0025	Supervisor-only Access Registers (FTFA_SACCL2)	8	R	Undefined	27.3.3.8/ 465
4002_0026	Supervisor-only Access Registers (FTFA_SACCL1)	8	R	Undefined	27.3.3.8/ 465
4002_0027	Supervisor-only Access Registers (FTFA_SACCL0)	8	R	Undefined	27.3.3.8/ 465
4002_0028	Flash Access Segment Size Register (FTFA_FACSS)	8	R	Undefined	27.3.3.9/ 466
4002_002B	Flash Access Segment Number Register (FTFA_FACSN)	8	R	Undefined	27.3.3.10/ 467

27.3.3.1 Flash Status Register (FTFA_FSTAT)

The FSTAT register reports the operational status of the flash memory module.

The CCIF, RDCOLERR, ACCERR, and FPVIOL bits are readable and writable. The MGSTAT0 bit is read only. The unassigned bits read 0 and are not writable.

NOTE

When set, the Access Error (ACCERR) and Flash Protection Violation (FPVIOL) bits in this register prevent the launch of any more commands until the flag is cleared (by writing a one to it).

Address: 4002_0000h base + 0h offset = 4002_0000h

Bit	7	6	5	4	3	2	1	0
Read	CCIF	RDCOLERR	ACCERR	FPVIOL	0			MGSTAT0
Write	w1c	w1c	w1c	w1c				
Reset	0	0	0	0	0	0	0	0

FTFA_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag Indicates that a flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command, and CCIF stays low until command completion or command violation. CCIF is reset to 0 but is set to 1 by the memory controller at the end of the reset initialization sequence. Depending on how quickly the read occurs after reset release, the user may or may not see the 0 hardware reset value. 0 Flash command in progress 1 Flash command has completed
6 RDCOLERR	Flash Read Collision Error Flag Indicates that the MCU attempted a read from a flash memory resource that was being manipulated by a flash command (CCIF=0). Any simultaneous access is detected as a collision error by the block arbitration logic. The read data in this case cannot be guaranteed. The RDCOLERR bit is cleared by writing a 1 to it. Writing a 0 to RDCOLERR has no effect. 0 No collision error detected 1 Collision error detected
5 ACCERR	Flash Access Error Flag Indicates an illegal access has occurred to a flash memory resource caused by a violation of the command write sequence or issuing an illegal flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR while CCIF is set. Writing a 0 to the ACCERR bit has no effect. 0 No access error detected 1 Access error detected
4 FPVIOL	Flash Protection Violation Flag Indicates an attempt was made to program or erase an address in a protected area of program flash memory during a command write sequence . While FPVIOL is set, the CCIF flag cannot be cleared to launch a command. The FPVIOL bit is cleared by writing a 1 to FPVIOL while CCIF is set. Writing a 0 to the FPVIOL bit has no effect. 0 No protection violation detected 1 Protection violation detected
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 MGSTAT0	Memory Controller Command Completion Status Flag

Table continues on the next page...

FTFA_FSTAT field descriptions (continued)

Field	Description
	The MGSTAT0 status flag is set if an error is detected during execution of a flash command or during the flash reset sequence. As a status flag, this field cannot (and need not) be cleared by the user like the other error flags in this register. The value of the MGSTAT0 bit for "command-N" is valid only at the end of the "command-N" execution when CCIF=1 and before the next command has been launched. At some point during the execution of "command-N+1," the previous result is discarded and any previous error is cleared.

27.3.3.2 Flash Configuration Register (FTFA_FCNFG)

This register provides information on the current functional state of the flash memory module.

The erase control bits (ERSAREQ and ERSSUSP) have write restrictions. The unassigned bits read as noted and are not writable.

Address: 4002_0000h base + 1h offset = 4002_0001h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	0	0	0	0
Write								
Reset	0	0	0	0	0	0	0	0

FTFA_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable Controls interrupt generation when a flash command completes. 0 Command complete interrupt disabled 1 Command complete interrupt enabled. An interrupt request is generated whenever the FSTAT[CCIF] flag is set.
6 RDCOLLIE	Read Collision Error Interrupt Enable Controls interrupt generation when a flash memory read collision error occurs. 0 Read collision error interrupt disabled 1 Read collision error interrupt enabled. An interrupt request is generated whenever a flash memory read collision error is detected (see the description of FSTAT[RDCOLERR]).
5 ERSAREQ	Erase All Request Issues a request to the memory controller to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. Refer to the device's Chip Configuration details on how to request this command. ERSAREQ sets when an erase all request is triggered external to the flash memory module and CCIF is set (no command is currently being executed). ERSAREQ is cleared by the flash memory module when the operation completes.

Table continues on the next page...

FTFA_FCNFG field descriptions (continued)

Field	Description
	0 No request or request complete 1 Request to: 1. run the Erase All Blocks command, 2. verify the erased state, 3. program the security byte in the Flash Configuration Field to the unsecure state, and 4. release MCU security by setting the FSEC[SEC] field to the unsecure state.
4 ERSSUSP	Erase Suspend Allows the user to suspend (interrupt) the Erase Flash Sector command while it is executing. 0 No suspend requested 1 Suspend the current Erase Flash Sector command execution.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

27.3.3.3 Flash Security Register (FTFA_FSEC)

This read-only register holds all bits associated with the security of the MCU and flash memory module.

During the reset sequence, the register is loaded with the contents of the flash security byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 2h offset = 4002_0002h

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable Enables or disables backdoor key access to the flash memory module.

Table continues on the next page...

FTFA_FSEC field descriptions (continued)

Field	Description
	00 Backdoor key access disabled 01 Backdoor key access disabled (preferred KEYEN state to disable backdoor key access) 10 Backdoor key access enabled 11 Backdoor key access disabled
5–4 MEEN	Mass Erase Enable Enables and disables mass erase capability of the flash memory module. When SEC is set to unsecure, the MEEN setting does not matter. 00 Mass erase is enabled 01 Mass erase is enabled 10 Mass erase is disabled 11 Mass erase is enabled
3–2 FSLACC	Freescale Failure Analysis Access Code Enables or disables access to the flash memory contents during returned part failure analysis at Freescale. When SEC is secure and FSLACC is denied, access to the program flash contents is denied and any failure analysis performed by Freescale factory test must begin with a full erase to unsecure the part. When access is granted (SEC is unsecure, or SEC is secure and FSLACC is granted), Freescale factory testing has visibility of the current flash contents. The state of the FSLACC bits is only relevant when SEC is set to secure. When SEC is set to unsecure, the FSLACC setting does not matter. 00 Freescale factory access granted 01 Freescale factory access denied 10 Freescale factory access denied 11 Freescale factory access granted
SEC	Flash Security Defines the security state of the MCU. In the secure state, the MCU limits access to flash memory module resources. The limitations are defined per device and are detailed in the Chip Configuration details. If the flash memory module is unsecured using backdoor key access, SEC is forced to 10b. 00 MCU security status is secure. 01 MCU security status is secure. 10 MCU security status is unsecure. (The standard shipping condition of the flash memory module is unsecure.) 11 MCU security status is secure.

27.3.3.4 Flash Option Register (FTFA_FOPT)

The flash option register allows the MCU to customize its operations by examining the state of these read-only bits, which are loaded from NVM at reset. The function of the bits is defined in the device's Chip Configuration details.

All bits in the register are read-only .

During the reset sequence, the register is loaded from the flash nonvolatile option byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 3h offset = 4002_0003h

Bit	7	6	5	4	3	2	1	0
Read	OPT							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_FOPT field descriptions

Field	Description
OPT	Nonvolatile Option These bits are loaded from flash to this register at reset. Refer to the device's Chip Configuration details for the definition and use of these bits.

27.3.3.5 Flash Common Command Object Registers (FTFA_FCCOB*n*)

The FCCOB register group provides 12 bytes for command codes and parameters. The individual bytes within the set append a 0-B hex identifier to the FCCOB register name: FCCOB0, FCCOB1, ..., FCCOBB.

Address: 4002_0000h base + 4h offset + (1d × i), where i=0d to 11d

Bit	7	6	5	4	3	2	1	0
Read Write	CCOBn							
Reset	0	0	0	0	0	0	0	0

FTFA_FCCOB*n* field descriptions

Field	Description
CCOBn	The FCCOB register provides a command code and relevant parameters to the memory controller. The individual registers that compose the FCCOB data set can be written in any order, but you must provide all needed values, which vary from command to command. First, set up all required FCCOB fields and then initiate the command's execution by writing a 1 to the FSTAT[CCIF] bit. This clears the CCIF bit, which locks all FCCOB parameter fields and they cannot be changed by the user until the command completes (CCIF returns to 1). No command buffering or queueing is provided; the next command can be loaded only after the current command completes. Some commands return information to the FCCOB registers. Any values returned to FCCOB are available for reading after the FSTAT[CCIF] flag returns to 1 by the memory controller.

FTFA_FCCOB*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<p>The following table shows a generic flash command format. The first FCCOB register, FCCOB0, always contains the command code. This 8-bit value defines the command to be executed. The command code is followed by the parameters required for this specific flash command, typically an address and/or data values.</p>
<p><b>NOTE:</b> The command parameter table is written in terms of FCCOB Number (which is equivalent to the byte number). This number is a reference to the FCCOB register name and is not the register address.</p>
<table>
<tr><th>FCCOB Number</th><th>Typical Command Parameter Contents [7:0]</th></tr>
<tr><td>0</td><td>FCMD (a code that defines the flash command)</td></tr>
<tr><td>1</td><td>Flash address [23:16]</td></tr>
<tr><td>2</td><td>Flash address [15:8]</td></tr>
<tr><td>3</td><td>Flash address [7:0]</td></tr>
<tr><td>4</td><td>Data Byte 0</td></tr>
<tr><td>5</td><td>Data Byte 1</td></tr>
<tr><td>6</td><td>Data Byte 2</td></tr>
<tr><td>7</td><td>Data Byte 3</td></tr>
<tr><td>8</td><td>Data Byte 4</td></tr>
<tr><td>9</td><td>Data Byte 5</td></tr>
<tr><td>A</td><td>Data Byte 6</td></tr>
<tr><td>B</td><td>Data Byte 7</td></tr>
</table>
<p><b>FCCOB Endianness and Multi-Byte Access :</b></p>
<p>The FCCOB register group uses a big endian addressing convention. For all command parameter fields larger than 1 byte, the most significant data resides in the lowest FCCOB register number. The FCCOB register group may be read and written as individual bytes, aligned words (2 bytes) or aligned longwords (4 bytes).</p>
</td></tr>
</table>

27.3.3.6 Program Flash Protection Registers (FTFA_FPROT*n*)

The FPROT registers define which program flash regions are protected from program and erase operations. Protected flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by program and erase operations.

The four FPROT registers allow up to 32 protectable regions. Each bit protects a 1/32 region of the program flash memory except for memory configurations with less than 32 KB of program flash where each assigned bit protects 1 KB . For configurations with 24 KB of program flash memory or less, FPROT0 is not used. For configurations with 16 KB of program flash memory or less, FPROT1 is not used. For configurations with 8 KB of program flash memory, FPROT2 is not used.For configurations with 24 KB of program flash memory or less, FPROT0 is not used. For configurations with 16 KB of program flash memory or less, FPROT1 is not used. For configurations with 8 KB of program flash memory, FPROT2 is not used. The bitfields are defined in each register as follows:

Program flash protection register	Program flash protection bits
FPROT0	PROT[31:24]
FPROT1	PROT[23:16]
FPROT2	PROT[15:8]
FPROT3	PROT[7:0]

During the reset sequence, the FPROT registers are loaded with the contents of the program flash protection bytes in the Flash Configuration Field as indicated in the following table.

Program flash protection register	Flash Configuration Field offset address
FPROT0	0x000B
FPROT1	0x000A
FPROT2	0x0009
FPROT3	0x0008

To change the program flash protection that is loaded during the reset sequence, unprotect the sector of program flash memory that contains the Flash Configuration Field. Then, reprogram the program flash protection byte.

Address: 4002_0000h base + 10h offset + (1d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read Write	PROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_FPROT*n* field descriptions

Field	Description
PROT	Program Flash Region Protect Each program flash region can be protected from program and erase operations by setting the associated PROT bit. The protection can only be increased, meaning that currently unprotected memory can be protected, but currently protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored.

FTFA_FPROT*n* field descriptions (continued)

Field	Description
	Restriction: The user must never write to any FPROT register while a command is running (CCIF=0). Trying to alter data in any protected area in the program flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A full block erase of a program flash block is not possible if it contains any protected region. Each bit in the 32-bit protection register represents 1/32 of the total program flash except for memory configurations with less than 32 KB of program flash where each assigned bit protects 1 KB . 0 Program flash region is protected. 1 Program flash region is not protected

27.3.3.7 Execute-only Access Registers (FTFA_XACC*n*)

The XACC registers define which program flash segments are restricted to data read or execute only or both data and instruction fetches.

The eight XACC registers allow up to 64 restricted segments of equal memory size.

Execute-only access register	Program flash execute-only access bits
XACCH0	XA[63:56]
XACCH1	XA[55:48]
XACCH2	XA[47:40]
XACCH3	XA[39:32]
XACCL0	XA[31:24]
XACCL1	XA[23:16]
XACCL2	XA[15:8]
XACCL3	XA[7:0]

During the reset sequence, the XACC registers are loaded with the logical AND of Program Flash IFR addresses A and B as indicated in the following table.

Execute-only access register	Program Flash IFR address A	Program Flash IFR address B
XACCH0	0xA3	0xAB
XACCH1	0xA2	0xAA
XACCH2	0xA1	0xA9
XACCH3	0xA0	0xA8
XACCL0	0xA7	0xAF
XACCL1	0xA6	0xAE
XACCL2	0xA5	0xAD
XACCL3	0xA4	0xAC

Use the Program Once command to program the execute-only access control fields that are loaded during the reset sequence.

Address: 4002_0000h base + 18h offset + (1d × i), where i=0d to 7d

Bit	7	6	5	4	3	2	1	0
Read	XA							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_XACC*n* field descriptions

Field	Description
XA	Execute-only access control 0 Associated segment is accessible in execute mode only (as an instruction fetch) 1 Associated segment is accessible as data or in execute mode

27.3.3.8 Supervisor-only Access Registers (FTFA_SACC*n*)

The SACC registers define which program flash segments are restricted to supervisor only or user and supervisor access.

The eight SACC registers allow up to 64 restricted segments of equal memory size.

Supervisor-only access register	Program flash supervisor-only access bits
SACCH0	SA[63:56]
SACCH1	SA[55:48]
SACCH2	SA[47:40]
SACCH3	SA[39:32]
SACCL0	SA[31:24]
SACCL1	SA[23:16]
SACCL2	SA[15:8]
SACCL3	SA[7:0]

During the reset sequence, the SACC registers are loaded with the logical AND of Program Flash IFR addresses A and B as indicated in the following table.

Supervisor-only access register	Program Flash IFR address A	Program Flash IFR address B
SACCH0	0xB3	0xBB
SACCH1	0xB2	0xBA

Table continues on the next page...

Supervisor-only access register	Program Flash IFR address A	Program Flash IFR address B
SACCH2	0xB1	0xB9
SACCH3	0xB0	0xB8
SACCL0	0xB7	0xBF
SACCL1	0xB6	0xBE
SACCL2	0xB5	0xBD
SACCL3	0xB4	0xBC

Use the Program Once command to program the supervisor-only access control fields that are loaded during the reset sequence.

Address: 4002_0000h base + 20h offset + (1d × i), where i=0d to 7d

Bit	7	6	5	4	3	2	1	0
Read	SA							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_SACC*n* field descriptions

Field	Description
SA	Supervisor-only access control 0 Associated segment is accessible in supervisor mode only 1 Associated segment is accessible in user or supervisor mode

27.3.3.9 Flash Access Segment Size Register (FTFA_FACSS)

The flash access segment size register determines which bits in the address are used to index into the SACC and XACC bitmaps to get the appropriate permission flags.

All bits in the register are read-only.

The contents of this register are loaded during the reset sequence.

Address: 4002_0000h base + 28h offset = 4002_0028h

Bit	7	6	5	4	3	2	1	0
Read	SGSIZE							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_FACSS field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>SGSIZE</td><td>Segment Size
The segment size is a fixed value based on the available program flash size divided by NUMSG.
<table>
<tr><td>Program Flash Size</td><td>Segment Size</td><td>Segment Size Encoding</td></tr>
<tr><td>64 KBytes</td><td>2 KBytes</td><td>0x3</td></tr>
<tr><td>128 KBytes</td><td>4 KBytes</td><td>0x4</td></tr>
<tr><td>160 KBytes</td><td>4 KBytes</td><td>0x4</td></tr>
<tr><td>256 KBytes</td><td>4 KBytes</td><td>0x4</td></tr>
<tr><td>512 KBytes</td><td>8 KBytes</td><td>0x5</td></tr>
</table></td></tr>
</table>

27.3.3.10 Flash Access Segment Number Register (FTFA_FACSN)

The flash access segment number register provides the number of program flash segments that are available for XACC and SACC permissions.

All bits in the register are read-only.

The contents of this register are loaded during the reset sequence.

Address: 4002_0000h base + 2Bh offset = 4002_002Bh

Bit	7	6	5	4	3	2	1	0
Read	NUMSG							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFA_FACSN field descriptions

Field	Description
NUMSG	Number of Segments Indicator The NUMSG field indicates the number of equal-sized segments in the program flash. 0x20 Program flash memory is divided into 32 segments (64 Kbytes, 128 Kbytes) 0x28 Program flash memory is divided into 40 segments (160 Kbytes) 0x40 Program flash memory is divided into 64 segments (256 Kbytes, 512 Kbytes)

27.4 Functional Description

The information found here describes functional details of the flash memory module.

27.4.1 Flash Protection

Individual regions within the flash memory can be protected from program and erase operations.

Protection is controlled by the following registers:

- FPROT*n* —
  - For 2^n program flash sizes, four registers typically protect 32 regions of the program flash memory as shown in the following figure

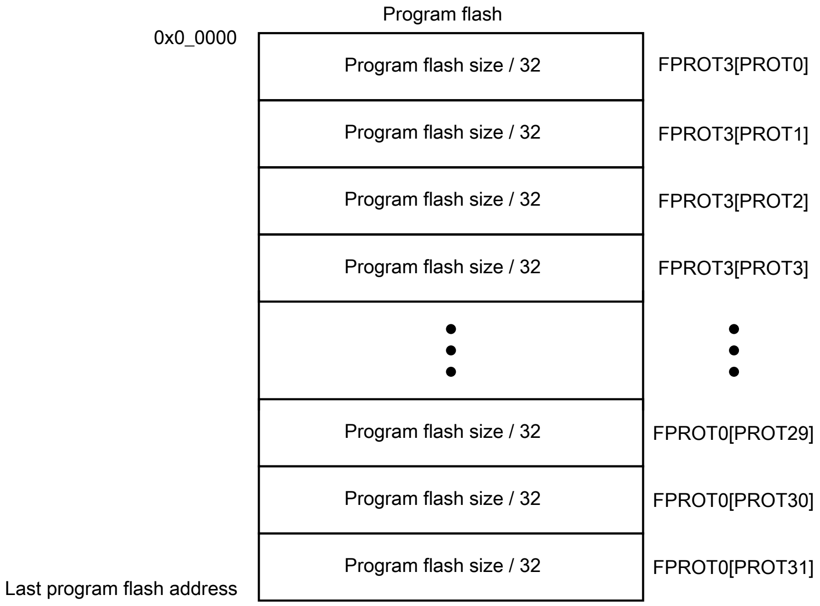


Figure 27-2. Program flash protection

  - For the non-2^n program flash size of 160KB, the protection granularity is 8KB. Therefore, only the PROT[19:0] bits are used.

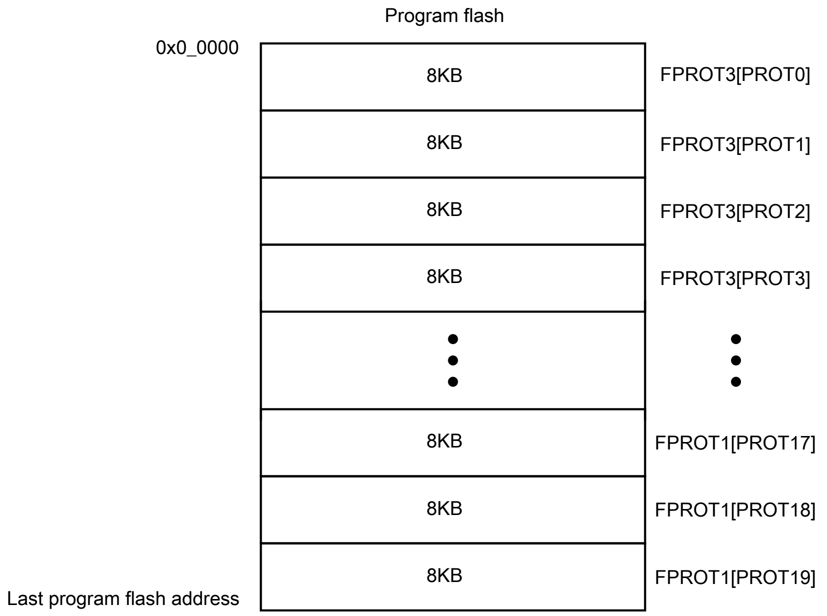


Figure 27-3. Program flash protection (160KB program flash size)

NOTE

Flash protection features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Not all features described in the application note are available on this device.

27.4.2 Flash Access Protection

Individual segments within the program flash memory can be designated for restricted access. Specific flash commands (Program Check, Program Longword, Erase Flash Block, Erase Flash Sector) monitor FXACC contents to protect flash memory but the FSACC contents do not impact flash command operation. Access is controlled by the following registers:

- FTFA_XACC —
  - For the non-2^n program flash size of 160KB, five registers control 40 segments of the program flash memory as shown in the following figure

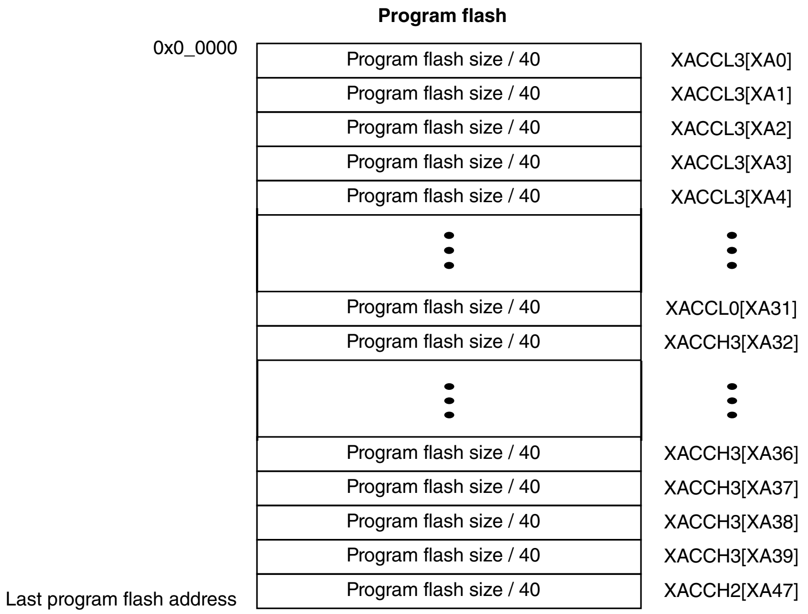


Figure 27-4. Program flash access control (160KB of program flash)

- FTFA_SACC —
  - For the non-2^n program flash size of 160KB, five registers control 40 segments of the program flash memory as shown in the following figure

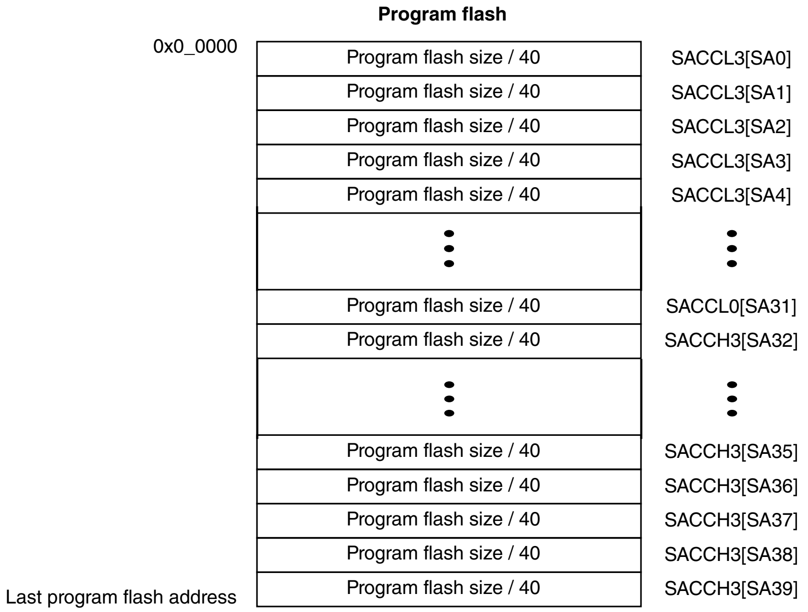


Figure 27-5. Program flash access control (160KB of program flash)

27.4.3 Interrupts

The flash memory module can generate interrupt requests to the MCU upon the occurrence of various flash events.

These interrupt events and their associated status and control bits are shown in the following table.

Table 27-1. Flash Interrupt Sources

Flash Event	Readable Status Bit	Interrupt Enable Bit
Flash Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
Flash Read Collision Error	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

Some devices also generate a bus error response as a result of a Read Collision Error event. See the chip configuration information to determine if a bus error response is also supported.

27.4.4 Flash Operation in Low-Power Modes

27.4.4.1 Wait Mode

When the MCU enters wait mode, the flash memory module is not affected. The flash memory module can recover the MCU from wait via the command complete interrupt (see Interrupts).

27.4.4.2 Stop Mode

When the MCU requests stop mode, if a flash command is active (CCIF = 0) the command execution completes before the MCU is allowed to enter stop mode.

CAUTION

The MCU should never enter stop mode while any flash command is running (CCIF = 0).

NOTE

While the MCU is in very-low-power modes (VLPR, VLPW, VLPS), the flash memory module does not accept flash commands.

27.4.5 Flash Reads and Ignored Writes

The flash memory module requires only the flash address to execute a flash memory read.

The MCU must not read from the flash memory while commands are running (as evidenced by CCIF=0) on that block. Read data cannot be guaranteed from a flash block while any command is processing within that block. The block arbitration logic detects any simultaneous access and reports this as a read collision error (see the FSTAT[RDCOLERR] bit).

27.4.6 Read While Write (RWW)

The following simultaneous accesses are allowed:

- The user may read from one logical program flash memory space while flash commands are active in the other logical program flash memory space.

Simultaneous operations are further discussed in Allowed Simultaneous Flash Operations.

27.4.7 Flash Program and Erase

All flash functions except read require the user to setup and launch a flash command through a series of peripheral bus writes.

The user cannot initiate any further flash commands until notified that the current command has completed. The flash command structure and operation are detailed in Flash Command Operations.

27.4.8 Flash Command Operations

Flash command operations are typically used to modify flash memory contents.

The next sections describe:

- The command write sequence used to set flash command parameters and launch execution
- A description of all flash commands available

27.4.8.1 Command Write Sequence

Flash commands are specified using a command write sequence illustrated in Figure 27-6. The flash memory module performs various checks on the command (FCCOB) content and continues with command execution if all requirements are fulfilled.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be zero and the CCIF flag must read 1 to verify that any previous command has completed. If CCIF is zero, the previous command execution is still active, a new command write sequence cannot be started, and all writes to the FCCOB registers are ignored.

Attempts to launch a flash command in VLP mode will be ignored. Attempts to launch a flash command while the BLE radio is active are ignored.

27.4.8.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the desired flash command. The individual registers that make up the FCCOB data set can be written in any order.

27.4.8.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing FSTAT[CCIF] by writing a '1' to it. FSTAT[CCIF] remains 0 until the flash command completes.

The FSTAT register contains a blocking mechanism that prevents a new command from launching (can't clear FSTAT[CCIF]) if the previous command resulted in an access error (FSTAT[ACCERR]=1) or a protection violation (FSTAT[FPVIOL]=1). In error scenarios, two writes to FSTAT are required to initiate the next command: the first write clears the error flags, the second write clears CCIF.

27.4.8.1.3 Command Execution and Error Reporting

The command processing has several steps:

1. The flash memory module reads the command code and performs a series of parameter checks and protection checks, if applicable, which are unique to each command.

If the parameter check fails, the FSTAT[ACCERR] (access error) flag is set. FSTAT[ACCERR] reports invalid instruction codes and out-of bounds addresses. Usually, access errors suggest that the command was not set-up with valid parameters in the FCCOB register group.

Program and erase commands also check the address to determine if the operation is requested to execute on protected areas. If the protection check fails, FSTAT[FPVIOL] (protection error) flag is set.

Command processing never proceeds to execution when the parameter or protection step fails. Instead, command processing is terminated after setting FSTAT[CCIF].

2. If the parameter and protection checks pass, the command proceeds to execution. Run-time errors, such as failure to erase verify, may occur during the execution phase. Run-time errors are reported in FSTAT[MGSTAT0]. A command may have access errors, protection errors, and run-time errors, but the run-time errors are not seen until all access and protection errors have been corrected.
3. Command execution results, if applicable, are reported back to the user via the FCCOB and FSTAT registers.
4. The flash memory module sets FSTAT[CCIF] signifying that the command has completed.

The flow for a generic command write sequence is illustrated in the following figure.

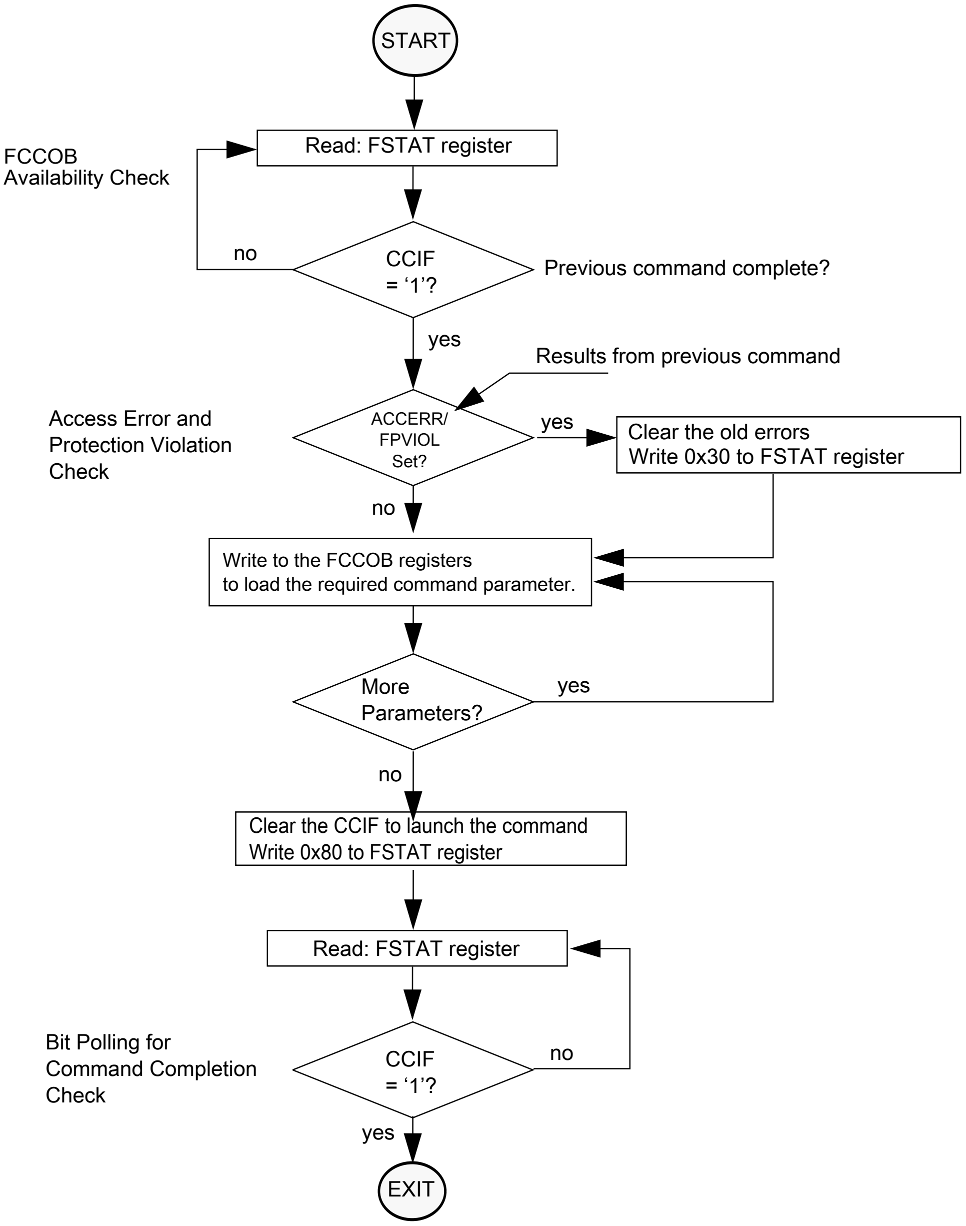


Figure 27-6. Generic flash command write sequence flowchart

27.4.8.2 Flash Commands

The following table summarizes the function of all flash commands.

FCMD	Command	Program flash 0	Program flash 1	Function
0x00	Read 1s Block	×	×	Verify that a program flash block is erased.

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1	Function
0x01	Read 1s Section	×	×	Verify that a given number of program flash locations from a starting address are erased.
0x02	Program Check	×	×	Tests previously-programmed locations at margin read levels.
0x03	Read Resource	IFR, ID	IFR	Read 4 bytes from program flash IFR or version ID.
0x06	Program Longword	×	×	Program 4 bytes in a program flash block.
0x08	Erase Flash Block	×	×	Erase a program flash block. An erase of any flash block is only possible when unprotected.
0x09	Erase Flash Sector	×	×	Erase all bytes in a program flash sector.
0x40	Read 1s All Blocks	×	×	Verify that all program flash blocks are erased then release MCU security.
0x41	Read Once	IFR		Read 4 bytes of a dedicated 64 byte field in the program flash 0 IFR.
0x43	Program Once	IFR		One-time program of 4 bytes of a dedicated 64-byte field in the program flash 0 IFR.
0x44	Erase All Blocks	×	×	Erase all program flash blocks. Then, verify-erase and release MCU security. **NOTE:** An erase is only possible when all memory locations are unprotected.
0x45	Verify Backdoor Access Key	×	×	Release MCU security after comparing a set of user-supplied security keys to those stored in the program flash.

27.4.8.3 Allowed Simultaneous Flash Operations

Only the operations marked 'OK' in the following table are permitted to run simultaneously on the program flash memories. Some operations cannot be executed simultaneously because certain hardware resources are shared by the memories.

Table 27-2. Allowed Simultaneous Memory Operations

		Program Flash 0			Program Flash 1		
		Read	Program	Sector Erase	Read	Program	Sector Erase
Program flash 0	Read	—				OK	OK
	Program		—		OK		
	Sector Erase			—	OK		
Program flash 1	Read		OK	OK	—		
	Program	OK				—	
	Sector Erase	OK					—

27.4.9 Margin Read Commands

The Read-1s commands (Read 1s All Blocks, Read 1s Block, Read 1s Section) and the Program Check command have a margin choice parameter that allows the user to apply non-standard read reference levels to the program flash array reads performed by these commands. Using the preset 'user' and 'factory' margin levels, these commands perform their associated read operations at tighter tolerances than a 'normal' read. These non-standard read levels are applied only during the command execution. Basic flash array reads use the standard, un-margined, read reference level.

Only the 'normal' read level should be employed during normal flash usage. The non-standard, 'user' and 'factory' margin levels should be employed only in special cases. They can be used during special diagnostic routines to gain confidence that the device is not suffering from the end-of-life data loss customary of flash memory devices.

Erased ('1') and programmed ('0') bit states can degrade due to elapsed time and data cycling (number of times a bit is erased and re-programmed). The lifetime of the erased states is relative to the last erase operation. The lifetime of the programmed states is measured from the last program time.

The 'user' and 'factory' levels become, in effect, a minimum safety margin; i.e. if the reads pass at the tighter tolerances of the 'user' and 'factory' margins, then the 'normal' reads have at least this much safety margin before they experience data loss.

The 'user' margin is a small delta to the normal read reference level. 'User' margin levels can be employed to check that flash memory contents have adequate margin for normal level read operations. If unexpected read results are encountered when checking flash memory contents at the 'user' margin levels, loss of information might soon occur during 'normal' readout.

The 'factory' margin is a bigger deviation from the norm, a more stringent read criteria that should only be attempted immediately (or very soon) after completion of an erase or program command, early in the cycling life. 'Factory' margin levels can be used to check that flash memory contents have adequate margin for long-term data retention at the normal level setting. If unexpected results are encountered when checking flash memory contents at 'factory' margin levels, the flash memory contents should be erased and reprogrammed.

CAUTION

Factory margin levels must only be used during verify of the initial factory programming.

27.4.10 Flash Command Description

This section describes all flash commands that can be launched by a command write sequence.

The flash memory module sets the FSTAT[ACCERR] bit and aborts the command execution if any of the following illegal conditions occur:

- There is an unrecognized command code in the FCCOB FCMD field.
- There is an error in a FCCOB field for the specific commands. Refer to the error handling table provided for each command.

Ensure that FSTAT[ACCERR] and FSTAT[FPVIOL] are cleared prior to starting the command write sequence. As described in Launch the Command by Clearing CCIF, a new command cannot be launched while these error flags are set.

Do not attempt to read a flash block while the flash memory module is running a command (FSTAT[CCIF] = 0) on that same block. The flash memory module may return invalid data to the MCU with the collision error flag (FSTAT[RDCOLERR]) set.

CAUTION

Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

27.4.10.1 Read 1s Block Command

The Read 1s Block command checks to see if an entire program flash block has been erased to the specified margin level. The FCCOB flash address bits determine which flash block is erase-verified.

Table 27-3. Read 1s Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x00 (RD1BLK)
1	Flash address [23:16] in the flash block to be verified
2	Flash address [15:8] in the flash block to be verified
3	Flash address [7:0][1] in the flash block to be verified
4	Read-1 Margin Choice

[1] Must be longword aligned (Flash address [1:0] = 00).

After clearing CCIF to launch the Read 1s Block command, the flash memory module sets the read margin for 1s according to Table 27-4 and then reads all locations within the selected program flash block.

Table 27-4. Margin Level Choices for Read 1s Block

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 27-5. Read 1s Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

27.4.10.2 Read 1s Section Command

The Read 1s Section command checks if a section of program flash memory is erased to the specified read margin level. The Read 1s Section command defines the starting address and the number of to be verified.

Upon clearing CCIF to launch the Read 1s Section command, the flash memory module sets the read margin for 1s according to Table 27-6 and then reads all locations within the specified section of flash memory. If the flash memory module fails to read all 1s (that is, the flash section is not erased), FSTAT[MGSTAT0] is set. FSTAT[CCIF] sets after the Read 1s Section operation completes.

Table 27-6. Margin Level Choices for Read 1s Section

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 27-7. Read 1s Section Command Error Handling

Error condition	Error bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin code is supplied.	FSTAT[ACCERR]
An invalid flash address is supplied.	FSTAT[ACCERR]
Flash address is not aligned.	FSTAT[ACCERR]
The requested section crosses a Flash block boundary.	FSTAT[ACCERR]
The requested number of is 0.	FSTAT[ACCERR]
Read-1s fails.	FSTAT[MGSTAT0]

27.4.10.3 Program Check Command

The Program Check command tests a previously programmed program flash longword to see if it reads correctly at the specified margin level.

Table 27-8. Program Check Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x02 (PGMCHK)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Margin Choice
8	Byte 0 expected data
9	Byte 1 expected data
A	Byte 2 expected data
B	Byte 3 expected data

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Check command, the flash memory module sets the read margin for 1s according to Table 27-9, reads the specified longword, and compares the actual read data to the expected data provided by the FCCOB. If the comparison at margin-1 fails, FSTAT[MGSTAT0] is set.

The flash memory module then sets the read margin for 0s, re-reads, and compares again. If the comparison at margin-0 fails, FSTAT[MGSTAT0] is set. FSTAT[CCIF] is set after the Program Check operation completes.

The supplied address must be longword aligned (the lowest two bits of the byte address must be 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10,
- Byte 0 data is programmed to byte address start+0b11.

NOTE

See the description of margin reads, Margin Read Commands

Table 27-9. Margin Level Choices for Program Check

Read Margin Choice	Margin Level Description
0x01	Read at 'User' margin-1 and 'User' margin-0
0x02	Read at 'Factory' margin-1 and 'Factory' margin-0

Table 27-10. Program Check Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
An invalid margin choice is supplied	FSTAT[ACCERR]
Flash address is located in an XA controlled segment and the Erase All Blocks or the Read 1s All Blocks command has not successfully completed since the last reset	FSTAT[FPVIOL]
Either of the margin reads does not match the expected data	FSTAT[MGSTAT0]

27.4.10.4 Read Resource Command

The Read Resource command allows the user to read data from special-purpose memory resources located within the flash memory module. The special-purpose memory resources available include program flash IFR space and the Version ID field. Each resource is assigned a select code as shown in Table 27-12.

Table 27-11. Read Resource Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x03 (RDRSRC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
Returned Values	
4	Read Data [31:24]
5	Read Data [23:16]
6	Read Data [15:8]
7	Read Data [7:0]
User-provided values	
8	Resource Select Code (see Table 27-12)

[1] Must be longword aligned (Flash address [1:0] = 00).

Table 27-12. Read Resource Select Codes

Resource Select Code	Description	Resource Size	Local Address Range
0x00	Program Flash 0 IFR	256 Bytes	0x00_0000–0x00_00FF
0x01[1]	Version ID	8 Bytes	0x00_0000–0x00_0007

[1] Located in program flash 0 reserved space.

After clearing CCIF to launch the Read Resource command, four consecutive bytes are read from the selected resource at the provided relative address and stored in the FCCOB register. The CCIF flag sets after the Read Resource operation completes. The Read Resource command exits with an access error if an invalid resource code is provided or if the address for the applicable area is out-of-range.

Table 27-13. Read Resource Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid resource code is entered	FSTAT[ACCERR]
Flash address is out-of-range for the targeted resource.	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]

27.4.10.5 Program Longword Command

The Program Longword command programs four previously-erased bytes in the program flash memory using an embedded algorithm.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 27-14. Program Longword Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x06 (PGM4)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Byte 0 program value
5	Byte 1 program value
6	Byte 2 program value
7	Byte 3 program value

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Longword command, the flash memory module programs the data bytes into the flash using the supplied address. The targeted flash locations must be currently unprotected (see the description of the FPROT registers) to permit execution of the Program Longword operation.

The programming operation is unidirectional. It can only move NVM bits from the erased state ('1') to the programmed state ('0'). Erased bits that fail to program to the '0' state are flagged as errors in FSTAT[MGSTAT0]. The CCIF flag is set after the Program Longword operation completes.

The supplied address must be longword aligned (flash address [1:0] = 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10, and
- Byte 0 data is programmed to byte address start+0b11.

Table 27-15. Program Longword Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Flash address points to a protected area	FSTAT[FPVIOL]

Table continues on the next page...

Table 27-15. Program Longword Command Error Handling (continued)

Error Condition	Error Bit
Flash address is located in an XA controlled segment and the Erase All Blocks or the Read 1s All Blocks command has not successfully completed since the last reset	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

27.4.10.6 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a single program flash.

Table 27-16. Erase Flash Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x08 (ERSBLK)
1	Flash address [23:16] in the flash block to be erased
2	Flash address [15:8] in the flash block to be erased
3	Flash address [7:0][1] in the flash block to be erased

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Erase Flash Block command, the flash memory module erases the main array of the selected flash block and verifies that it is erased. The Erase Flash Block command aborts and sets the FSTAT[FPVIOL] bit if any region within the block is protected (see the description of the FPROT registers). If the erase verify fails, FSTAT[MGSTAT0] is set. The CCIF flag will set after the Erase Flash Block operation has completed.

CAUTION

The Erase Flash Block operation will not react to the early indicator for BLE radio activity. Therefore, the Erase Flash Block command must not be launched if there is a concern about inadequate power available to support both BLE radio activity and the Erase Flash Block operation.

Table 27-17. Erase Flash Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Any area of the selected flash block is protected	FSTAT[FPVIOL]

Table continues on the next page...

Table 27-17. Erase Flash Block Command Error Handling (continued)

Error Condition	Error Bit
The selected program flash block contains an XA controlled segment and the Erase All Blocks or the Read 1s All Blocks command has not successfully completed since the last reset	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s Block command to verify all bits are erased.

27.4.10.7 Erase Flash Sector Command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 27-18. Erase Flash Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x09 (ERSSCR)
1	Flash address [23:16] in the flash sector to be erased
2	Flash address [15:8] in the flash sector to be erased
3	Flash address [7:0] in the flash sector to be erased

After clearing CCIF to launch the Erase Flash Sector command, the flash memory module erases the selected program flash sector and then verifies that it is erased. The Erase Flash Sector command aborts if the selected sector is protected (see the description of the FPROT registers). If the erase-verify fails the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase Flash Sector operation completes. The Erase Flash Sector command is suspendable (see the FCNFG[ERSSUSP] bit and Figure 27-7).

Table 27-19. Erase Flash Sector Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid Flash address is supplied	FSTAT[ACCERR]
Flash address is not aligned	FSTAT[ACCERR]
The selected program flash sector is protected	FSTAT[FPVIOL]
The selected program flash sector is located in an XA controlled segment and the Erase All Blocks or the Read 1s All Blocks command has not successfully completed since the last reset	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s Section command to verify all bits are erased.

27.4.10.7.1 Suspending an Erase Flash Sector Operation

To suspend an Erase Flash Sector operation set the FCNFG[ERSSUSP] bit when CCIF, ACCERR, and FPVIOL are clear and the CCOB command field holds the code for the Erase Flash Sector command. During the Erase Flash Sector operation (see Erase Flash Sector Command), the flash memory module samples the state of the ERSSUSP bit at convenient points. If the flash memory module detects that the ERSSUSP bit is set, the Erase Flash Sector operation is suspended and the flash memory module sets CCIF. While ERSSUSP is set, all writes to flash registers are ignored except for writes to the FSTAT and FCNFG registers.

If an Erase Flash Sector operation effectively completes before the flash memory module detects that a suspend request has been made, the flash memory module clears the ERSSUSP bit prior to setting CCIF. When an Erase Flash Sector operation has been successfully suspended, the flash memory module sets CCIF and leaves the ERSSUSP bit set. While CCIF is set, the ERSSUSP bit can only be cleared to prevent the withdrawal of a suspend request before the flash memory module has acknowledged it.

27.4.10.7.2 Resuming a Suspended Erase Flash Sector Operation

If the ERSSUSP bit is still set when CCIF is cleared to launch the next command, the previous Erase Flash Sector operation resumes. The flash memory module acknowledges the request to resume a suspended operation by clearing the ERSSUSP bit. A new suspend request can then be made by setting ERSSUSP. A single Erase Flash Sector operation can be suspended and resumed multiple times.

There is a minimum elapsed time limit of 4.3 msec between the request to resume the Erase Flash Sector operation (CCIF is cleared) and the request to suspend the operation again (ERSSUSP is set). This minimum time period is required to ensure that the Erase Flash Sector operation will eventually complete. If the minimum period is continually violated, i.e. the suspend requests come repeatedly and too quickly, no forward progress is made by the Erase Flash Sector algorithm. The resume/suspend sequence runs indefinitely without completing the erase.

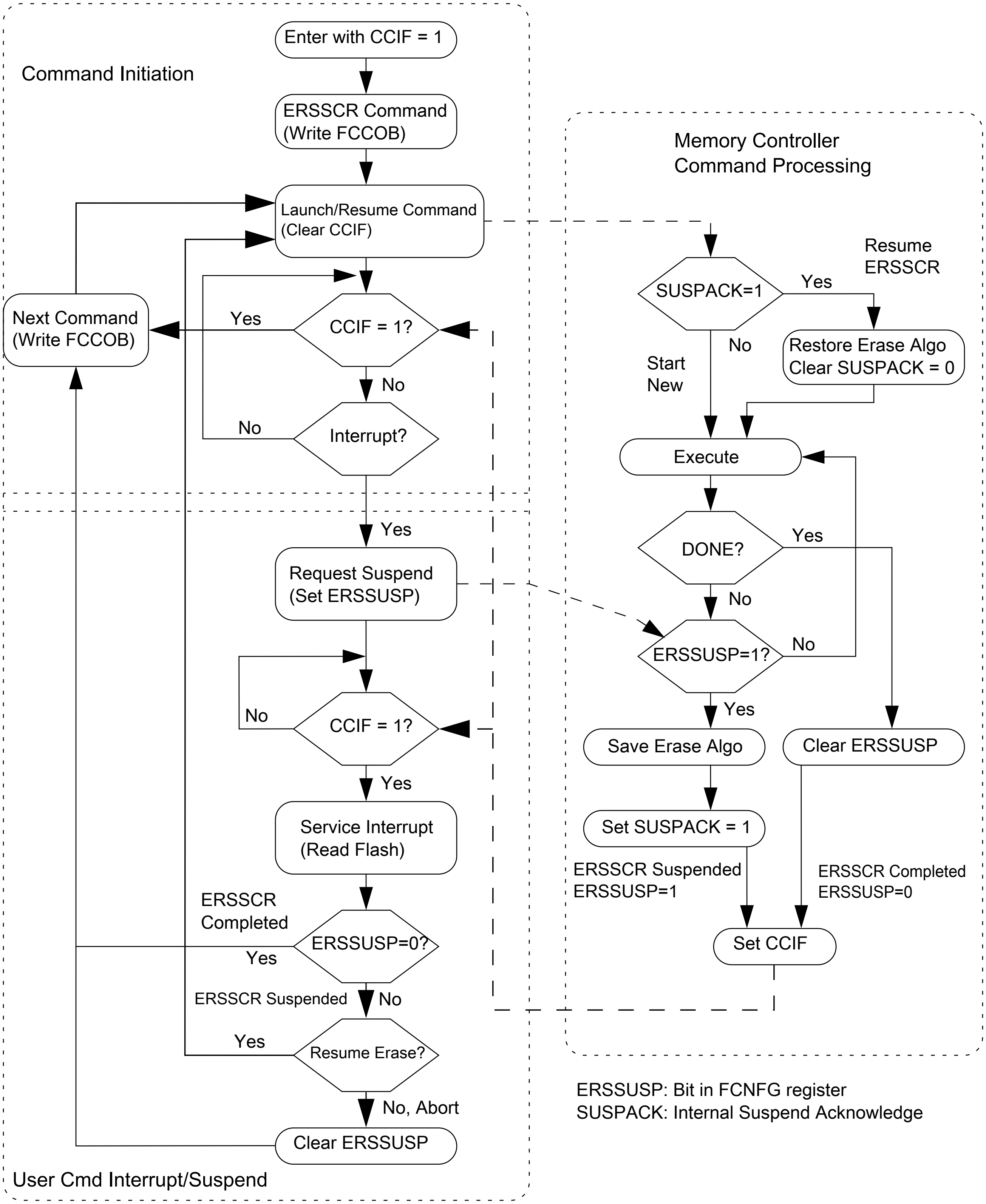
27.4.10.7.3 Aborting a Suspended Erase Flash Sector Operation

The user may choose to abort a suspended Erase Flash Sector operation by clearing the ERSSUSP bit prior to clearing CCIF for the next command launch. When a suspended operation is aborted, the flash memory module starts the new command using the new FCCOB contents.

Note

Aborting the erase leaves the bitcells in an indeterminate, partially-erased state. Data in this sector is not reliable until a new erase command fully completes.

The following figure shows how to suspend and resume the Erase Flash Sector operation.


Command Initiation
Enter with CCIF = 1
ERSSCR Command (Write FCCOB)
Launch/Resume Command (Clear CCIF)
Next Command (Write FCCOB)
Yes
CCIF = 1?
No
No
Interrupt?
Yes
Request Suspend (Set ERSSUSP)
No
CCIF = 1?
Yes
Service Interrupt (Read Flash)
ERSSCR Completed
ERSSUSP=0?
Yes
ERSSCR Suspended
No
Yes
Resume Erase?
No, Abort
Clear ERSSUSP
User Cmd Interrupt/Suspend
Memory Controller Command Processing
SUSPACK=1
Yes
Resume ERSSCR
Restore Erase Algo Clear SUSPACK = 0
No
Start New
Execute
DONE?
Yes
No
ERSSUSP=1?
No
Yes
Save Erase Algo
Clear ERSSUSP
Set SUSPACK = 1
ERSSCR Suspended ERSSUSP=1
ERSSCR Completed ERSSUSP=0
Set CCIF

ERSSUSP: Bit in FCNFG register
SUSPACK: Internal Suspend Acknowledge

Figure 27-7. Suspend and Resume of Erase Flash Sector Operation

27.4.10.7.4 Impact of BLE radio activity on Erase Flash Sector Operation

The RSIM module provides an early indicator to the flash module that the BLE radio is going active and also indicates when the BLE radio goes inactive. If the BLE active indicator is asserted while the Erase Flash Sector operation is active, the operation will stall to reduce power consumption before the BLE radio goes active. The Erase Flash Sector operation will resume after the BLE active indicator negates. FSTAT[CCIF] remains clear during the stall to prevent disruption of the Erase Flash Sector operation while the block containing the sector being erased remains unavailable for read operations.

The following figure shows how the Erase Flash Sector operation stalls and resumes based on BLE radio activity.

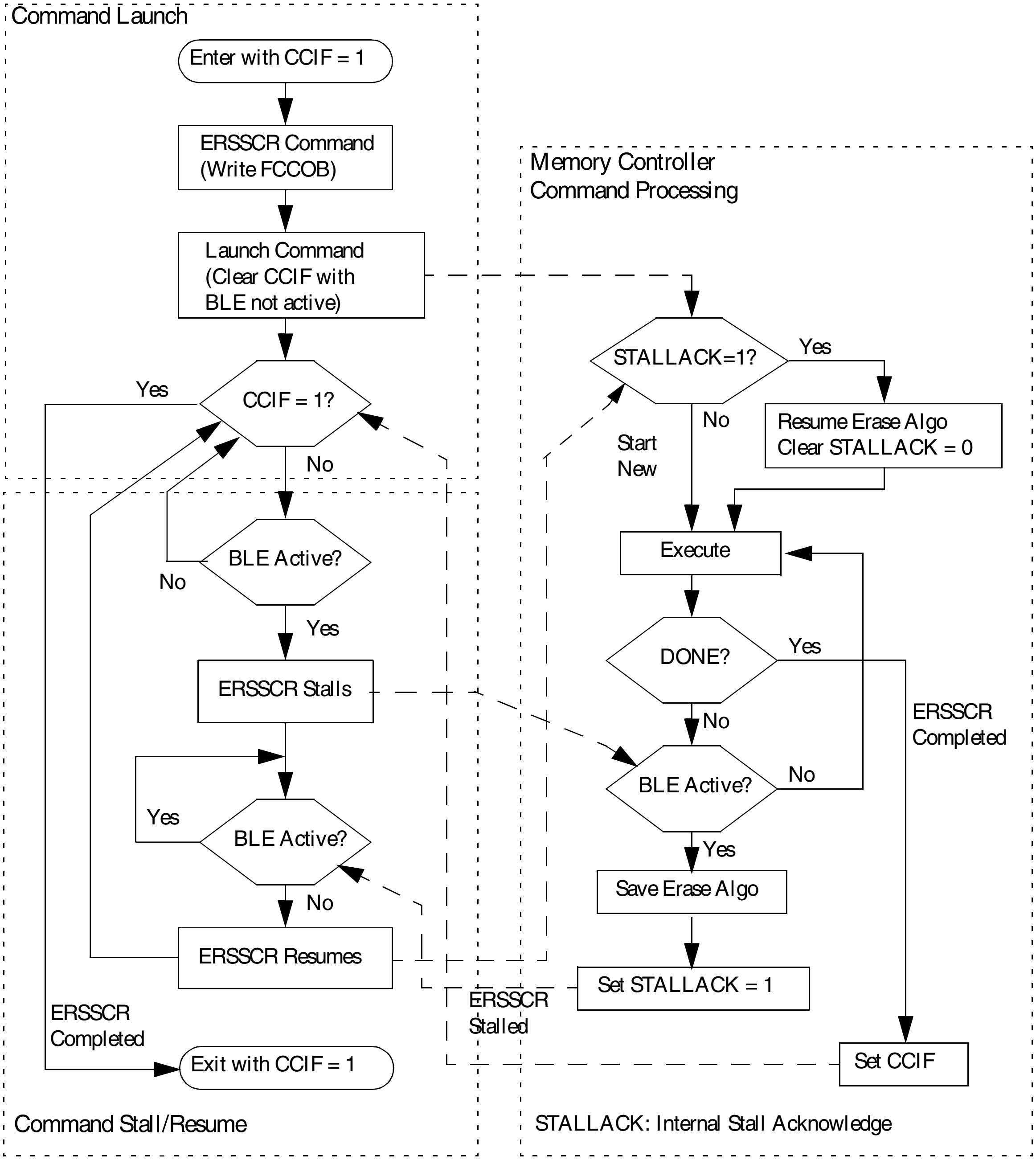


Figure 27-8. BLE Active Impact on Erase Flash Sector Operation

27.4.10.8 Read 1s All Blocks Command

The Read 1s All Blocks command checks if the program flash blocks have been erased to the specified read margin level, if applicable, and releases security if the readout passes, i.e. all data reads as '1'.

Table 27-20. Read 1s All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x40 (RD1ALL)
1	Read-1 Margin Choice

After clearing CCIF to launch the Read 1s All Blocks command, the flash memory module :

- sets the read margin for 1s according to Table 27-21,
- checks the contents of the program flash are in the erased state.

If the flash memory module confirms that these memory resources are erased, access control is disabled and security is released by setting the FSEC[SEC] field to the unsecure state. The security byte in the flash configuration field (see Flash Configuration Field Description) remains unaffected by the Read 1s All Blocks command. If the read fails, i.e. all memory resources are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The CCIF flag sets after the Read 1s All Blocks operation has completed.

Table 27-21. Margin Level Choices for Read 1s All Blocks

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 27-22. Read 1s All Blocks Command Error Handling

Error Condition	Error Bit
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

27.4.10.9 Read Once Command

The Read Once command provides read access to special 96-byte fields located in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once ID field is via 16 records (index values 0x00 - 0x0F), each 4 bytes long. Access to the Program Once XACC and SACC fields are via 4 records (index values 0x10 - 0x13), each of which is 8 bytes long. These fields are programmed using the Program Once command described in Program Once Command.

Table 27-23. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Program Once record index (0x00 - 0x13)
2	Not used
3	Not used
Returned Values	
4	Program Once byte 0 value
5	Program Once byte 1 value
6	Program Once byte 2 value
7	Program Once byte 3 value
8	Program Once byte 4 value (index 0x10 - 0x13)
9	Program Once byte 5 value (index 0x10 - 0x13)
10	Program Once byte 6 value (index 0x10 - 0x13)
11	Program Once byte 7 value (index 0x10 - 0x13)

After clearing CCIF to launch the Read Once command, a 4-byte or 8-byte Program Once record is read and stored in the FCCOB register. The CCIF flag is set after the Read Once operation completes. Valid record index values for the Read Once command range from 0x00 - 0x13. During execution of the Read Once command, any attempt to read addresses within the program flash block containing the selected record index returns invalid data. The Read Once command can be executed any number of times.

Table 27-24. Read Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]

27.4.10.10 Program Once Command

The Program Once command enables programming to special 96-byte fields in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once ID field is via 16 records (index values 0x00 - 0x0F), each 4 bytes long. Access to the Program Once XACC and SACC fields are via 4 records (index values 0x10 - 0x13), each of which is 8 bytes long. These records can be read using the Read Once command (see Read Once Command) or using the Read Resource command (see Read Resource Command). These records can be programmed only once since the program flash 0 IFR cannot be erased.

Table 27-25. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x13)
2	Not Used
3	Not Used
4	Program Once byte 0 value
5	Program Once byte 1 value
6	Program Once byte 2 value
7	Program Once byte 3 value
8	Program Once byte 4 value (index 0x10 - 0x13)
9	Program Once byte 5 value (index 0x10 - 0x13)
10	Program Once byte 6 value (index 0x10 - 0x13)
11	Program Once byte 7 value (index 0x10 - 0x13)

After clearing CCIF to launch the Program Once command, the flash memory module first verifies that the selected record is erased. If erased, then the selected record is programmed using the values provided. The Program Once command also verifies that the programmed values read back correctly. The CCIF flag is set after the Program Once operation has completed.

Any attempt to program one of these records when the existing value is not Fs (erased) is not allowed. Valid record index values for the Program Once command range from 0x00 - 0x13. During execution of the Program Once command, any attempt to read addresses within the program flash block containing the selected record index returns invalid data.

Table 27-26. Program Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]
The requested record has already been programmed to a non-FFFF value[1]	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

[1] If a Program Once record is initially programmed to 0xFFFF_FFFF (0xFFFF_FFFF_FFFF_FFFF for index 0x10 - 0x13), the Program Once command is allowed to execute again on that same record.

27.4.10.11 Erase All Blocks Command

The Erase All Blocks operation erases all flash memory, verifies all memory contents, and releases MCU security.

Table 27-27. Erase All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x44 (ERSALL)

After clearing CCIF to launch the Erase All Blocks command, the flash memory module erases all program flash memory, then verifies that all are erased.

If the flash memory module verifies that all flash memories were properly erased, access control is disabled and security is released by setting the FSEC[SEC] field to the unsecure state. The Erase All Blocks command aborts if any flash region is protected. The security byte and all other contents of the flash configuration field (see Flash Configuration Field Description) are erased by the Erase All Blocks command. If the erase-verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Blocks operation completes.

Access control determined by the contents of the FXACC registers will not block execution of the Erase All Blocks command. While most Flash memory will be erased, the program flash IFR space containing the Program Once XACC and SACC fields will not be erased and, therefore, the contents of the Program Once XACC and SACC fields will not change. The contents of the FXACC and FSACC registers will not be impacted by the execution of the Erase All Blocks command. After completion of the Erase All Blocks command, access control is disabled until the next reset of the flash module or the Read 1s All Blocks command is executed and fails (FSTAT[MGSTAT0] is set).

CAUTION

The Erase All Blocks operation will not react to the early indicator for BLE radio activity. Therefore, the Erase All Blocks command must not be launched if there is a concern about inadequate power available to support both BLE radio activity and the Erase All Blocks operation.

Table 27-28. Erase All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any region of the program flash memory is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s All Blocks command to verify all bits are erased.

27.4.10.11.1 Triggering an Erase All External to the Flash Memory Module

The functionality of the Erase All Blocks command is also available in an uncommanded fashion outside of the flash memory. Refer to the device's Chip Configuration details for information on this functionality.

Before invoking the external erase all function, the FSTAT[ACCERR and PVIOL] flags must be cleared and the FCCOB0 register must not contain 0x44. When invoked, the erase-all function erases all program flash memory regardless of the protection settings. If the post-erase verify passes, access control determined by the contents of the FXACC registers is disabled and the routine then releases security by setting the FSEC[SEC] field register to the unsecure state. The security byte in the Flash Configuration Field is also programmed to the unsecure state. The status of the erase-all request is reflected in the FCNFG[ERSAREQ] bit. The FCNFG[ERSAREQ] bit is cleared once the operation completes and the normal FSTAT error reporting is available, except FPVIOL, as described in Erase All Blocks Command.

CAUTION

> The Erase All Blocks operation will not react to the early indicator for BLE radio activity. Therefore, the Erase All pin must not be asserted if there is a concern about inadequate power available to support both BLE radio activity and the Erase All Blocks operation.

27.4.10.12 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command only executes if the mode and security conditions are satisfied (see Flash Commands by Mode). Execution of the Verify Backdoor Access Key command is further qualified by the FSEC[KEYEN] bits. The Verify Backdoor Access Key command releases security if user-supplied keys in the FCCOB match those stored in the Backdoor Comparison Key bytes of the Flash Configuration Field (see Flash Configuration Field Description). The column labelled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 27-29. Verify Backdoor Access Key Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]	Flash Configuration Field Offset Address
0	0x45 (VFYKEY)	
1-3	Not Used	
4	Key Byte 0	0x0_0003
5	Key Byte 1	0x0_0002

Table continues on the next page...

Table 27-29. Verify Backdoor Access Key Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]	Flash Configuration Field Offset Address
6	Key Byte 2	0x0_0001
7	Key Byte 3	0x0_0000
8	Key Byte 4	0x0_0007
9	Key Byte 5	0x0_0006
A	Key Byte 6	0x0_0005
B	Key Byte 7	0x0_0004

After clearing CCIF to launch the Verify Backdoor Access Key command, the flash memory module checks the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the flash memory module sets the FSTAT[ACCERR] bit and terminates. If the command is enabled, the flash memory module compares the key provided in FCCOB to the backdoor comparison key in the Flash Configuration Field. If the backdoor keys match, the FSEC[SEC] field is changed to the unsecure state and security is released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are immediately aborted and the FSTAT[ACCERR] bit is (again) set to 1 until a reset of the flash memory module module occurs. If the entire 8-byte key is all zeros or all ones, the Verify Backdoor Access Key command fails with an access error. The CCIF flag is set after the Verify Backdoor Access Key operation completes.

Table 27-30. Verify Backdoor Access Key Command Error Handling

Error Condition	Error Bit
The supplied key is all-0s or all-Fs	FSTAT[ACCERR]
An incorrect backdoor key is supplied	FSTAT[ACCERR]
Backdoor key access has not been enabled (see the description of the FSEC register)	FSTAT[ACCERR]
This command is launched and the backdoor key has mismatched since the last power down reset	FSTAT[ACCERR]

27.4.11 Security

The flash memory module provides security information to the MCU based on contents of the FSEC security register.

The MCU then limits access to flash memory resources as defined in the device's Chip Configuration details. During reset, the flash memory module initializes the FSEC register using data read from the security byte of the Flash Configuration Field (see Flash Configuration Field Description).

The following fields are available in the FSEC register. The settings are described in the Flash Security Register (FTFA_FSEC) details.

Flash security features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Note that not all features described in the application note are available on this device.

Table 27-31. FSEC register fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Freescale Factory Access
SEC	MCU security

27.4.11.1 Changing the Security State

The security state out of reset can be permanently changed by programming the security byte of the flash configuration field. This assumes that you are starting from a mode where the necessary program flash erase and program commands are available and that the region of the program flash containing the flash configuration field is unprotected. If the flash security byte is successfully programmed, its new value takes affect after the next chip reset.

27.4.11.1.1 Unsecuring the Chip Using Backdoor Key Access

The chip can be unsecured by using the backdoor key access feature, which requires knowledge of the contents of the 8-byte backdoor key value stored in the Flash Configuration Field (see Flash Configuration Field Description). If the FSEC[KEYEN] bits are in the enabled state, the Verify Backdoor Access Key command (see Verify Backdoor Access Key Command) can be run; it allows the user to present prospective keys for comparison to the stored keys. If the keys match, the FSEC[SEC] bits are changed to unsecure the chip. The entire 8-byte key cannot be all 0s or all 1s; that is, 0000_0000_0000_0000h and FFFF_FFFF_FFFF_FFFFh are not accepted by the Verify Backdoor Access Key command as valid comparison values. While the Verify Backdoor Access Key command is active, program flash memory is not available for read access and returns invalid data.

The user code stored in the program flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN bits are in the enabled state, the chip can be unsecured by the following backdoor key access sequence:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Verify Backdoor Access Key Command
2. If the Verify Backdoor Access Key command is successful, the chip is unsecured and the FSEC[SEC] bits are forced to the unsecure state

An illegal key provided to the Verify Backdoor Access Key command prohibits further use of the Verify Backdoor Access Key command. A reset of the chip is the only method to re-enable the Verify Backdoor Access Key command when a comparison fails.

After the backdoor keys have been correctly matched, the chip is unsecured by changing the FSEC[SEC] bits. A successful execution of the Verify Backdoor Access Key command changes the security in the FSEC register only. It does not alter the security byte or the keys stored in the Flash Configuration Field (Flash Configuration Field Description). After the next reset of the chip, the security state of the flash memory module reverts back to the flash security byte in the Flash Configuration Field. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the program flash protection registers.

If the backdoor keys successfully match, the unsecured chip has full control of the contents of the Flash Configuration Field. The chip may erase the sector containing the Flash Configuration Field and reprogram the flash security byte to the unsecure state and change the backdoor keys to any desired value.

27.4.12 Reset Sequence

On each system reset the flash memory module executes a sequence which establishes initial values for the flash block configuration parameters, FPROT, FOPT, FSEC, FXACC, FSACC, and FACNFG registers.

FSTAT[CCIF] is cleared throughout the reset sequence. The flash memory module holds off CPU access during the reset sequence. Flash reads are possible when the hold is removed. Completion of the reset sequence is marked by setting CCIF which enables flash user commands.

If a reset occurs while any flash command is in progress, that command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed. Commands and operations do not automatically resume after exiting reset.

Chapter 28
Analog-to-digital converter (ADC)

The 16-bit analog-to-digital converter (ADC) is a linear successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

28.1 Introduction

The 16-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

> For the chip specific modes of operation, see the power management information of the device.

28.1.1 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm with up to 16-bit resolution
- Up to four pairs of differential and 24 single-ended external analog inputs
- Output modes:
  - differential 16-bit, 13-bit, 11-bit, and 9-bit modes
  - single-ended 16-bit, 12-bit, 10-bit, and 8-bit modes
- Output format in 2's complement 16-bit sign extended for differential modes
- Output in right-justified unsigned format for single-ended
- Single or continuous conversion, that is, automatic return to idle after single conversion

- Configurable sample time and conversion speed/power
- Conversion complete/hardware average complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in low-power modes for lower noise
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-Calibration mode

28.1.2 Block diagram

The following figure is the ADC module block diagram.

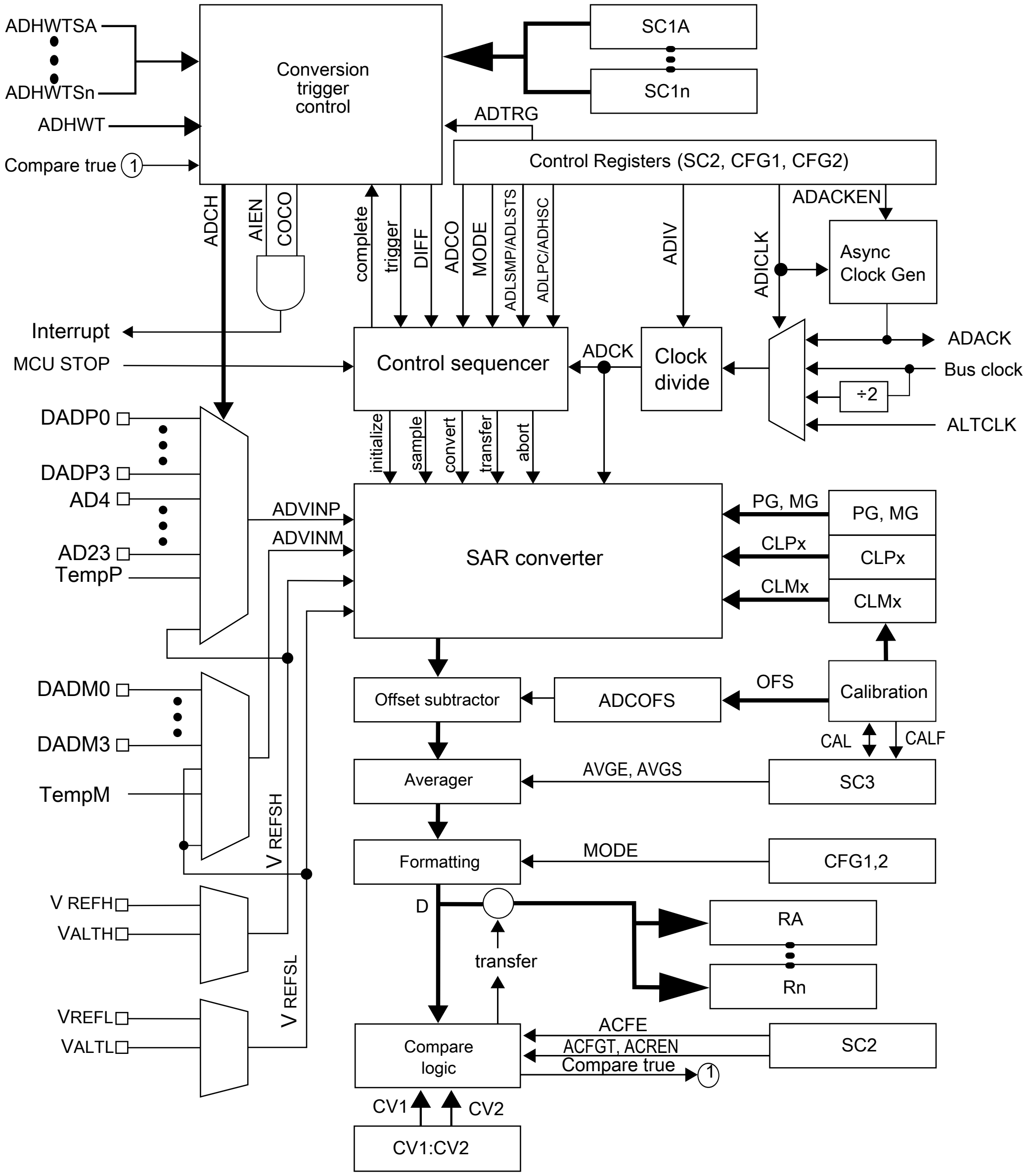


Figure 28-1. ADC block diagram

28.2 ADC signal descriptions

The ADC module supports up to 4 pairs of differential inputs and up to 24 single-ended inputs.

Each differential pair requires two inputs, DADPx and DADMx. The ADC also requires four supply/reference/ground connections.

NOTE

For the number of channels supported on this device as well as information regarding other chip-specific inputs into the ADC block, see the chip-specific ADC configuration information.

Table 28-1. ADC signal descriptions

Signal	Description	I/O
DADP3–DADP0	Differential Analog Channel Inputs	I
DADM3–DADM0	Differential Analog Channel Inputs	I
AD*n*	Single-Ended Analog Channel Inputs	I
V_{REFSH}	Voltage Reference Select High	I
V_{REFSL}	Voltage Reference Select Low	I
V_{DDA}	Analog Power Supply	I
V_{SSA}	Analog Ground	I

28.2.1 Analog Power (V_{DDA})

The ADC analog portion uses V_{DDA} as its power connection. In some packages, V_{DDA} is connected internally to V_{DD}. If externally available, connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

28.2.2 Analog Ground (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground connection. In some packages, V_{SSA} is connected internally to V_{SS}. If externally available, connect the V_{SSA} pin to the same voltage potential as V_{SS}.

28.2.3 Voltage Reference Select

V_{REFSH} and V_{REFSL} are the high and low reference voltages for the ADC module.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate V_{ALTH} and

V_{ALTL} voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the Voltage References specific to this MCU.

In some packages, V_{REFH} is connected in the package to V_{DDA} and V_{REFL} to V_{SSA}. If externally available, the positive reference(s) may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. V_{REFH} must never exceed V_{DDA}. Connect the ground references to the same voltage potential as V_{SSA}.

28.2.4 Analog Channel Inputs (ADx)

The ADC module supports up to 24 single-ended analog inputs. A single-ended input is selected for conversion through the SC1[ADCH] channel select bits when SC1n[DIFF] is low.

28.2.5 Differential Analog Channel Inputs (DADx)

The ADC module supports up to four differential analog channel inputs. Each differential analog input is a pair of external pins, DADPx and DADMx, referenced to each other to provide the most accurate analog to digital readings. A differential input is selected for conversion through SC1[ADCH] when SC1n[DIFF] is high. All DADPx inputs may be used as single-ended inputs if SC1n[DIFF] is low. In certain MCU configurations, some DADMx inputs may also be used as single-ended inputs if SC1n[DIFF] is low. For ADC connections specific to this device, see the chip-specific ADC information.

28.3 Memory map and register definitions

This section describes the ADC registers.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	ADC Status and Control Registers 1 (ADC0_SC1A)	32	R/W	0000_001Fh	28.3.1/504
4003_B004	ADC Status and Control Registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	28.3.1/504
4003_B008	ADC Configuration Register 1 (ADC0_CFG1)	32	R/W	0000_0000h	28.3.2/508
4003_B00C	ADC Configuration Register 2 (ADC0_CFG2)	32	R/W	0000_0000h	28.3.3/509
4003_B010	ADC Data Result Register (ADC0_RA)	32	R	0000_0000h	28.3.4/510

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B014	ADC Data Result Register (ADC0_RB)	32	R	0000_0000h	28.3.4/510
4003_B018	Compare Value Registers (ADC0_CV1)	32	R/W	0000_0000h	28.3.5/512
4003_B01C	Compare Value Registers (ADC0_CV2)	32	R/W	0000_0000h	28.3.5/512
4003_B020	Status and Control Register 2 (ADC0_SC2)	32	R/W	0000_0000h	28.3.6/513
4003_B024	Status and Control Register 3 (ADC0_SC3)	32	R/W	0000_0000h	28.3.7/515
4003_B028	ADC Offset Correction Register (ADC0_OFS)	32	R/W	0000_0004h	28.3.8/516
4003_B02C	ADC Plus-Side Gain Register (ADC0_PG)	32	R/W	0000_8200h	28.3.9/517
4003_B030	ADC Minus-Side Gain Register (ADC0_MG)	32	R/W	0000_8200h	28.3.10/ 517
4003_B034	ADC Plus-Side General Calibration Value Register (ADC0_CLPD)	32	R/W	0000_000Ah	28.3.11/ 518
4003_B038	ADC Plus-Side General Calibration Value Register (ADC0_CLPS)	32	R/W	0000_0020h	28.3.12/ 519
4003_B03C	ADC Plus-Side General Calibration Value Register (ADC0_CLP4)	32	R/W	0000_0200h	28.3.13/ 519
4003_B040	ADC Plus-Side General Calibration Value Register (ADC0_CLP3)	32	R/W	0000_0100h	28.3.14/ 520
4003_B044	ADC Plus-Side General Calibration Value Register (ADC0_CLP2)	32	R/W	0000_0080h	28.3.15/ 520
4003_B048	ADC Plus-Side General Calibration Value Register (ADC0_CLP1)	32	R/W	0000_0040h	28.3.16/ 521
4003_B04C	ADC Plus-Side General Calibration Value Register (ADC0_CLP0)	32	R/W	0000_0020h	28.3.17/ 521
4003_B054	ADC Minus-Side General Calibration Value Register (ADC0_CLMD)	32	R/W	0000_000Ah	28.3.18/ 522
4003_B058	ADC Minus-Side General Calibration Value Register (ADC0_CLMS)	32	R/W	0000_0020h	28.3.19/ 522
4003_B05C	ADC Minus-Side General Calibration Value Register (ADC0_CLM4)	32	R/W	0000_0200h	28.3.20/ 523
4003_B060	ADC Minus-Side General Calibration Value Register (ADC0_CLM3)	32	R/W	0000_0100h	28.3.21/ 523
4003_B064	ADC Minus-Side General Calibration Value Register (ADC0_CLM2)	32	R/W	0000_0080h	28.3.22/ 524
4003_B068	ADC Minus-Side General Calibration Value Register (ADC0_CLM1)	32	R/W	0000_0040h	28.3.23/ 524
4003_B06C	ADC Minus-Side General Calibration Value Register (ADC0_CLM0)	32	R/W	0000_0020h	28.3.24/ 525

28.3.1 ADC Status and Control Registers 1 (ADCx_SC1*n*)

SC1A is used for both software and hardware trigger modes of operation.

To allow sequential conversions of the ADC to be triggered by internal peripherals, the ADC can have more than one status and control register: one for each conversion. The SC1B–SC1n registers indicate potentially multiple SC1 registers for use only in hardware trigger mode. See the chip configuration information about the number of SC1n registers specific to this device. The SC1n registers have identical fields, and are used in a "ping-pong" approach to control ADC operation.

At any one point in time, only one of the SC1n registers is actively controlling ADC conversions. Updating SC1A while SC1n is actively controlling a conversion is allowed, and vice-versa for any of the SC1n registers specific to this MCU.

Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, writes to SC1A subsequently initiate a new conversion, if SC1[ADCH] contains a value other than all 1s (module disabled).

Writing any of the SC1n registers while that specific SC1n register is actively controlling a conversion aborts the current conversion. None of the SC1B-SC1n registers are used for software trigger operation and therefore writes to the SC1B–SC1n registers do not initiate a new conversion.

Address: 4003_B000h base + 0h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								COCO	AIEN	DIFF	ADCH				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

ADCx_SC1n field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 COCO	Conversion Complete Flag This is a read-only field that is set each time a conversion is completed when the compare function is disabled, or SC2[ACFE]=0 and the hardware average function is disabled, or SC3[AVGE]=0. When the compare function is enabled, or SC2[ACFE]=1, COCO is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled, or SC3[AVGE]=1, COCO is set upon completion of the selected number of conversions (determined by AVGS). COCO in SC1A is also set at the completion of a calibration sequence. COCO is cleared when the respective SC1n register is written or when the respective Rn register is read. 0 Conversion is not completed. 1 Conversion is completed.
6 AIEN	Interrupt Enable Enables conversion complete interrupts. When COCO becomes set while the respective AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt is disabled. 1 Conversion complete interrupt is enabled.
5 DIFF	Differential Mode Enable Configures the ADC to operate in differential mode. When enabled, this mode automatically selects from the differential channels, and changes the conversion algorithm and the number of cycles to complete a conversion. 0 Single-ended conversions and input channels are selected. 1 Differential conversions and input channels are selected.
ADCH	Input channel select Selects one of the input channels. The input channel decode depends on the value of DIFF. DAD0-DAD3 are associated with the input pin pairs DADPx and DADMx. **NOTE:** Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the actual ADC channel assignments for your device, see the Chip Configuration details. The successive approximation converter subsystem is turned off when the channel select bits are all set, that is, ADCH = 11111. This feature allows explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. It is not necessary to set ADCH to all 1s to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes. 00000 When DIFF=0, DADP0 is selected as input; when DIFF=1, DAD0 is selected as input. 00001 When DIFF=0, DADP1 is selected as input; when DIFF=1, DAD1 is selected as input. 00010 When DIFF=0, DADP2 is selected as input; when DIFF=1, DAD2 is selected as input. 00011 When DIFF=0, DADP3 is selected as input; when DIFF=1, DAD3 is selected as input. 00100 When DIFF=0, AD4 is selected as input; when DIFF=1, it is reserved. 00101 When DIFF=0, AD5 is selected as input; when DIFF=1, it is reserved. 00110 When DIFF=0, AD6 is selected as input; when DIFF=1, it is reserved. 00111 When DIFF=0, AD7 is selected as input; when DIFF=1, it is reserved.

Table continues on the next page...

ADCx_SC1*n* field descriptions (continued)

Field	Description
	01000 When DIFF=0, AD8 is selected as input; when DIFF=1, it is reserved.
	01001 When DIFF=0, AD9 is selected as input; when DIFF=1, it is reserved.
	01010 When DIFF=0, AD10 is selected as input; when DIFF=1, it is reserved.
	01011 When DIFF=0, AD11 is selected as input; when DIFF=1, it is reserved.
	01100 When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved.
	01101 When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved.
	01110 When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved.
	01111 When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved.
	10000 When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved.
	10001 When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved.
	10010 When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved.
	10011 When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved.
	10100 When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved.
	10101 When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved.
	10110 When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved.
	10111 When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved.
	11000 Reserved.
	11001 Reserved.
	11010 When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input.
	11011 When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input.
	11100 Reserved.
	11101 When DIFF=0,V_{REFSH} is selected as input; when DIFF=1, -V_{REFSH} (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL].
	11110 When DIFF=0,V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL].
	11111 Module is disabled.

28.3.2 ADC Configuration Register 1 (ADCx_CFG1)

The configuration Register 1 (CFG1) selects the mode of operation, clock source, clock divide, and configuration for low power or long sample time.

Address: 4003_B000h base + 8h offset = 4003_B008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADLPC	ADIV		ADLSMP	MODE		ADICLK	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADLPC	Low-Power Configuration Controls the power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required. 0 Normal power configuration. 1 Low-power configuration. The power is reduced at the expense of maximum clock speed.
6–5 ADIV	Clock Divide Select Selects the divide ratio used by the ADC to generate the internal clock ADCK. 00 The divide ratio is 1 and the clock rate is input clock. 01 The divide ratio is 2 and the clock rate is (input clock)/2. 10 The divide ratio is 4 and the clock rate is (input clock)/4. 11 The divide ratio is 8 and the clock rate is (input clock)/8.
4 ADLSMP	Sample Time Configuration Selects between different sample times based on the conversion mode selected. This field adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption if continuous conversions are enabled and high conversion rates are not required. When ADLSMP=1, the long sample time select bits, (ADLSTS[1:0]), can select the extent of the long sample time.

Table continues on the next page...

ADCx_CFG1 field descriptions (continued)

Field	Description
	0 Short sample time. 1 Long sample time.
3–2 MODE	Conversion mode selection Selects the ADC resolution mode. 00 When DIFF=0:It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0:It is single-ended 12-bit conversion ; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0:It is single-ended 10-bit conversion. ; when DIFF=1, it is differential 11-bit conversion with 2's complement output 11 When DIFF=0:It is single-ended 16-bit conversion..; when DIFF=1, it is differential 16-bit conversion with 2's complement output
ADICLK	Input Clock Select Selects the input clock source to generate the internal clock, ADCK. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start, when CFG2[ADACKEN]=0, the asynchronous clock is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. 00 Bus clock 01 Bus clock divided by 2(BUSCLK/2) 10 Alternate clock (ALTCLK) 11 Asynchronous clock (ADACK)

28.3.3 ADC Configuration Register 2 (ADCx_CFG2)

Configuration Register 2 (CFG2) selects the special high-speed configuration for very high speed conversions and selects the long sample time duration during long sample mode.

Address: 4003_B000h base + Ch offset = 4003_B00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			MUXSEL	ADACKEN	ADHSC	ADLSTS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MUXSEL	ADC Mux Select Changes the ADC mux setting to select between alternate sets of ADC channels. 0 ADxxa channels are selected. 1 ADxxb channels are selected.
3 ADACKEN	Asynchronous Clock Output Enable Enables the asynchronous clock source and the clock source output regardless of the conversion and status of CFG1[ADICLK]. Based on MCU configuration, the asynchronous clock may be used by other modules. See chip configuration information. Setting this field allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced because the ADACK clock is already operational. 0 Asynchronous clock output disabled; Asynchronous clock is enabled only if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output is enabled regardless of the state of the ADC.
2 ADHSC	High-Speed Configuration Configures the ADC for very high-speed operation. The conversion sequence is altered with 2 ADCK cycles added to the conversion time to allow higher speed conversion clocks. 0 Normal conversion sequence selected. 1 High-speed conversion sequence selected with 2 additional ADCK cycles to total conversion time.
ADLSTS	Long Sample Time Select Selects between the extended sample times when long sample time is selected, that is, when CFG1[ADLSMP]=1. This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 00 Default longest sample time; 20 extra ADCK cycles; 24 ADCK cycles total. 01 12 extra ADCK cycles; 16 ADCK cycles total sample time. 10 6 extra ADCK cycles; 10 ADCK cycles total sample time. 11 2 extra ADCK cycles; 6 ADCK cycles total sample time.

28.3.4 ADC Data Result Register (ADCx_R*n*)

The data result registers (Rn) contain the result of an ADC conversion of the channel selected by the corresponding status and channel control register (SC1A:SC1n). For every status and channel control register, there is a corresponding data result register.

Unused bits in R n are cleared in unsigned right-aligned modes and carry the sign bit (MSB) in sign-extended 2's complement modes. For example, when configured for 10-bit single-ended mode, D[15:10] are cleared. When configured for 11-bit differential mode, D[15:10] carry the sign bit, that is, bit 10 extended through bit 15.

The following table describes the behavior of the data result registers in the different modes of operation.

Table 28-2. Data result register description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign-extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right-justified

NOTE

S: Sign bit or sign bit extension;

D: Data, which is 2's complement data if indicated

Address: 4003_B000h base + 10h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																D															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_R*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
D	Data result

28.3.5 Compare Value Registers (ADC*x*_CV*n*)

The Compare Value Registers (CV1 and CV2) contain a compare value used to compare the conversion result when the compare function is enabled, that is, SC2[ACFE]=1. This register is formatted in the same way as the Rn registers in different modes of operation for both bit position definition and value format using unsigned or sign-extended 2's complement. Therefore, the compare function uses only the CVn fields that are related to the ADC mode of operation.

The compare value 2 register (CV2) is used only when the compare range function is enabled, that is, SC2[ACREN]=1.

Address: 4003_B000h base + 18h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CV															
W	▨																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC*x*_CV*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CV	Compare Value.

28.3.6 Status and Control Register 2 (ADCx_SC2)

The status and control register 2 (SC2) contains the conversion active, hardware/software trigger select, compare function, and voltage reference select of the ADC module.

Address: 4003_B000h base + 20h offset = 4003_B020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADACT	Conversion Active Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.
6 ADTRG	Conversion Trigger Select Selects the type of trigger used for initiating a conversion. Two types of trigger are selectable:

Table continues on the next page...

ADCx_SC2 field descriptions (continued)

Field	Description
	• Software trigger: When software trigger is selected, a conversion is initiated following a write to SC1A. • Hardware trigger: When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input after a pulse of the ADHWTSn input. 0 Software trigger selected. 1 Hardware trigger selected.
5 ACFE	Compare Function Enable Enables the compare function. 0 Compare function disabled. 1 Compare function enabled.
4 ACFGT	Compare Function Greater Than Enable Configures the compare function to check the conversion result relative to the CV1 and CV2 based upon the value of ACREN. ACFE must be set for ACFGT to have any effect. 0 Configures less than threshold, outside range not inclusive and inside range not inclusive; functionality based on the values placed in CV1 and CV2. 1 Configures greater than or equal to threshold, outside and inside ranges inclusive; functionality based on the values placed in CV1 and CV2.
3 ACREN	Compare Function Range Enable Configures the compare function to check if the conversion result of the input being monitored is either between or outside the range formed by CV1 and CV2 determined by the value of ACFGT. ACFE must be set for ACFGT to have any effect. 0 Range function disabled. Only CV1 is compared. 1 Range function enabled. Both CV1 and CV2 are compared.
2 DMAEN	DMA Enable 0 DMA is disabled. 1 DMA is enabled and will assert the ADC DMA request during an ADC conversion complete event noted when any of the SC1n[COCO] flags is asserted.
REFSEL	Voltage Reference Selection Selects the voltage reference source used for conversions. 00 Default voltage reference pin pair, that is, external pins V_{REFH} and V_{REFL} 01 Alternate reference pair, that is, V_{ALTH} and V_{ALTL} . This pair may be additional external pins or internal sources depending on the MCU configuration. See the chip configuration information for details specific to this MCU 10 Reserved 11 Reserved

28.3.7 Status and Control Register 3 (ADCx_SC3)

The Status and Control Register 3 (SC3) controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Address: 4003_B000h base + 24h offset = 4003_B024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CAL	CALF	0		ADCO	AVGE	AVGS	
W										w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC3 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CAL	Calibration Begins the calibration sequence when set. This field stays set while the calibration is in progress and is cleared when the calibration sequence is completed. CALF must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and CALF will set. Setting CAL will abort any current conversion.
6 CALF	Calibration Failed Flag Displays the result of the calibration sequence. The calibration sequence will fail if SC2[ADTRG] = 1, any ADC register is written, or any stop mode is entered before the calibration sequence completes. Writing 1 to CALF clears it. 0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.

Table continues on the next page...

ADCx_SC3 field descriptions (continued)

Field	Description
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ADCO	Continuous Conversion Enable Enables continuous conversions. 0 One conversion or one set of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion. 1 Continuous conversions or sets of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion.
2 AVGE	Hardware Average Enable Enables the hardware average function of the ADC. 0 Hardware average function disabled. 1 Hardware average function enabled.
AVGS	Hardware Average Select Determines how many ADC conversions will be averaged to create the ADC average result. 00 4 samples averaged. 01 8 samples averaged. 10 16 samples averaged. 11 32 samples averaged.

28.3.8 ADC Offset Correction Register (ADCx_OFS)

The ADC Offset Correction Register (OFS) contains the user-selected or calibration-generated offset error correction value. This register is a 2's complement, left-justified, 16-bit value . The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 28h offset = 4003_B028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ADCx_OFS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OFS	Offset Error Correction Value

28.3.9 ADC Plus-Side Gain Register (ADCx_PG)

The Plus-Side Gain Register (PG) contains the gain error correction for the plus-side input in differential mode or the overall conversion in single-ended mode. PG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between PG[15] and PG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 2Ch offset = 4003_B02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PG															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_PG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PG	Plus-Side Gain

28.3.10 ADC Minus-Side Gain Register (ADCx_MG)

The Minus-Side Gain Register (MG) contains the gain error correction for the minus-side input in differential mode. This register is ignored in single-ended mode. MG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between MG[15] and MG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 30h offset = 4003_B030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MG															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_MG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MG	Minus-Side Gain

28.3.11 ADC Plus-Side General Calibration Value Register (ADCx_CLPD)

The Plus-Side General Calibration Value Registers (CLPx) contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLP0[5:0], CLP1[6:0], CLP2[7:0], CLP3[8:0], CLP4[9:0], CLPS[5:0], and CLPD[5:0]. CLPx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 34h offset = 4003_B034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLPD					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

ADCx_CLPD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPD	Calibration Value Calibration Value

28.3.12 ADC Plus-Side General Calibration Value Register (ADCx_CLPS)

For more information, see CLPD register description.

Address: 4003_B000h base + 38h offset = 4003_B038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLPS					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLPS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPS	Calibration Value Calibration Value

28.3.13 ADC Plus-Side General Calibration Value Register (ADCx_CLP4)

For more information, see CLPD register description.

Address: 4003_B000h base + 3Ch offset = 4003_B03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLP4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLP4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP4	Calibration Value Calibration Value

28.3.14 ADC Plus-Side General Calibration Value Register (ADCx_CLP3)

For more information, see CLPD register description.

Address: 4003_B000h base + 40h offset = 4003_B040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLP3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLP3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP3	Calibration Value Calibration Value

28.3.15 ADC Plus-Side General Calibration Value Register (ADCx_CLP2)

For more information, see CLPD register description.

Address: 4003_B000h base + 44h offset = 4003_B044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLP2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLP2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP2	Calibration Value Calibration Value

28.3.16 ADC Plus-Side General Calibration Value Register (ADCx_CLP1)

For more information, see CLPD register description.

Address: 4003_B000h base + 48h offset = 4003_B048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLP1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLP1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP1	Calibration Value Calibration Value

28.3.17 ADC Plus-Side General Calibration Value Register (ADCx_CLP0)

For more information, see CLPD register description.

Address: 4003_B000h base + 4Ch offset = 4003_B04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLP0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLP0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP0	Calibration Value Calibration Value

28.3.18 ADC Minus-Side General Calibration Value Register (ADCx_CLMD)

The Minus-Side General Calibration Value (CLMx) registers contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLM0[5:0], CLM1[6:0], CLM2[7:0], CLM3[8:0], CLM4[9:0], CLMS[5:0], and CLMD[5:0]. CLMx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 54h offset = 4003_B054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLMD					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

ADCx_CLMD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMD	Calibration Value Calibration Value

28.3.19 ADC Minus-Side General Calibration Value Register (ADCx_CLMS)

For more information, see CLMD register description.

Address: 4003_B000h base + 58h offset = 4003_B058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLMS					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLMS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMS	Calibration Value Calibration Value

28.3.20 ADC Minus-Side General Calibration Value Register (ADCx_CLM4)

For more information, see CLMD register description.

Address: 4003_B000h base + 5Ch offset = 4003_B05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLM4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLM4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM4	Calibration Value Calibration Value

28.3.21 ADC Minus-Side General Calibration Value Register (ADCx_CLM3)

For more information, see CLMD register description.

Address: 4003_B000h base + 60h offset = 4003_B060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLM3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLM3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ADCx_CLM3 field descriptions (continued)

Field	Description
CLM3	Calibration Value Calibration Value

28.3.22 ADC Minus-Side General Calibration Value Register (ADCx_CLM2)

For more information, see CLMD register description.

Address: 4003_B000h base + 64h offset = 4003_B064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLM2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLM2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM2	Calibration Value Calibration Value

28.3.23 ADC Minus-Side General Calibration Value Register (ADCx_CLM1)

For more information, see CLMD register description.

Address: 4003_B000h base + 68h offset = 4003_B068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLM1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLM1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM1	Calibration Value Calibration Value

28.3.24 ADC Minus-Side General Calibration Value Register (ADCx_CLM0)

For more information, see CLMD register description.

Address: 4003_B000h base + 6Ch offset = 4003_B06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLM0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLM0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM0	Calibration Value Calibration Value

28.4 Functional description

The ADC module is disabled during reset, in Low-Power Stop mode, or when SC1n[ADCH] are all high; see the power management information for details. The module is idle when a conversion has completed and another conversion has not been initiated. When it is idle and the asynchronous clock output enable is disabled, or CFG2[ADACKEN]= 0, the module is in its lowest power state. The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications, the ADC module must be calibrated using the on-chip calibration function.

See Calibration function for details on how to perform calibration.

When the conversion is completed, the result is placed in the Rn data registers. The respective SC1n[COCO] is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, or, when SC1n[AIEN]=1.

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the CV1 and CV2 registers. The compare function is enabled by setting SC2[ACFE] and operates in any of the conversion modes and configurations.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting SC3[AVGE] and operates in any of the conversion modes and configurations.

NOTE

> For the chip specific modes of operation, see the power management information of this MCU.

28.4.1 Clock select and divide control

One of four clock sources can be selected as the clock source for the ADC module.

This clock source is then divided by a configurable value to generate the input clock ADCK, to the module. The clock is selected from one of the following sources by means of CFG1[ADICLK].

- Bus clock. This is the default selection following reset.
- Bus clock divided by two. For higher bus clock rates, this allows a maximum divide-by-16 of the bus clock using CFG1[ADIV].
- ALTCLK: As defined for this MCU. See the chip configuration information. Conversions are possible using ALTCLK as the input clock source while the MCU is in Normal Stop mode.
- Asynchronous clock (ADACK): This clock is generated from a clock source within the ADC module. When the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start CFG2[ADACKEN]=0, ADACK is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. To avoid the conversion time variability and latency associated with the ADACK clock startup, set CFG2[ADACKEN]=1 and wait the worst-case startup time of 5 µs prior to initiating any conversions using the ADACK clock source. Conversions are possible using ADACK as the input clock source while the MCU is in Normal Stop mode. See Power Control for more information.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by CFG1[ADIV] and can be divide-by 1, 2, 4, or 8.

28.4.2 Voltage reference selection

The ADC can be configured to accept one of the two voltage reference pairs as the reference voltage (V_{REFSH} and V_{REFSL}) used for conversions.

Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

28.4.3 Hardware trigger and channel selects

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when SC2[ADTRG] is set and a hardware trigger select event, ADHWTSn, has occurred.

This source is not available on all MCUs. See the chip-specific ADC information for information on the ADHWT source and the ADHWTSn configurations specific to this MCU.

When an ADHWT source is available and hardware trigger is enabled, that is SC2[ADTRG]=1, a conversion is initiated on the rising-edge of ADHWT after a hardware trigger select event, that is, ADHWTSn, has occurred. If a conversion is in progress when a rising-edge of a trigger occurs, the rising-edge is ignored. In continuous convert configuration, only the initial rising-edge to launch continuous conversions is observed, and until conversion is aborted, the ADC continues to do conversions on the same SCn register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The hardware trigger select event, ADHWTSn, must be set no less than 3 ADCK cycles prior to the receipt of the ADHWT signal. If these conditions are not met, the converter may ignore the trigger or use the incorrect configuration. If a hardware trigger select event is asserted during a conversion, it must stay asserted until the end of current conversion and remain set until the receipt of the ADHWT signal to trigger a new conversion. The channel and status fields selected for the conversion depend on the active trigger select signal:

- ADHWTSA active selects SC1A.
- ADHWTSn active selects SC1n.

Note

> Asserting more than one hardware trigger select signal (ADHWTSn) at the same time results in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to the next intended conversion.

When the conversion is completed, the result is placed in the Rn registers associated with the ADHWTSn received. For example:

- ADHWTSA active selects RA register
- ADHWTSn active selects Rn register

The conversion complete flag associated with the ADHWTSn received, that is, SC1n[COCO], is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, that is, SC1[AIEN]=1.

28.4.4 Conversion control

Conversions can be performed as determined by CFG1[MODE] and SC1n[DIFF] as shown in the description of CFG1[MODE].

Conversions can be initiated by a software or hardware trigger.

In addition, the ADC module can be configured for:

- Low-power operation
- Long sample time
- Continuous conversion
- Hardware average
- Automatic compare of the conversion result to a software determined compare value

28.4.4.1 Initiating conversions

A conversion is initiated:

- Following a write to SC1A, with SC1n[ADCH] not all 1's, if software triggered operation is selected, that is, when SC2[ADTRG]=0.
- Following a hardware trigger, or ADHWT event, if hardware triggered operation is selected, that is, SC2[ADTRG]=1, and a hardware trigger select event, ADHWTSn, has occurred. The channel and status fields selected depend on the active trigger select signal:
    - ADHWTSA active selects SC1A.

- ADHWTSn active selects SC1n.
- if neither is active, the off condition is selected

Note

> Selecting more than one ADHWTSn prior to a conversion completion will result in unknown results. To avoid this, select only one ADHWTSn prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled, that is, when SC3[ADCO] = 1.

If continuous conversions are enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, that is, when SC2[ADTRG] = 0, continuous conversions begin after SC1A is written and continue until aborted. In hardware triggered operation, that is, when SC2[ADTRG] = 1 and one ADHWTSn event has occurred, continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions are completed. In software triggered operation, conversions begin after SC1A is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions are also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

28.4.4.2 Completing conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, Rn. If the compare functions are disabled, this is indicated by setting of SC1n[COCO]. If hardware averaging is enabled, the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed. If the compare function is enabled, the respective SC1n[COCO] sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled, then the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed and the compare condition is true. An interrupt is generated if the respective SC1n[AIEN] is high at the time that the respective SC1n[COCO] is set.

28.4.4.3 Aborting conversions

Any conversion in progress is aborted when:

- Writing to SC1A while it is actively controlling a conversion, aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, a write to SC1A initiates a new conversion if SC1A[ADCH] is equal to a value other than all 1s. Writing to any of the SC1B–SC1n registers while that specific SC1B–SC1n register is actively controlling a conversion aborts the current conversion. The SC1(B-n) registers are not used for software trigger operation and therefore writes to the SC1(B-n) registers do not initiate a new conversion.
- A write to any ADC register besides the SC1A-SC1n registers occurs. This indicates that a change in mode of operation has occurred and the current conversion is therefore invalid.
- The MCU is reset or enters Low-Power Stop modes.
- The MCU enters Normal Stop mode with ADACK or Alternate Clock Sources not enabled.

When a conversion is aborted, the contents of the data registers, Rn, are not altered. The data registers continue to be the values transferred after the completion of the last successful conversion. If the conversion was aborted by a reset or Low-Power Stop modes, RA and Rn return to their reset states.

28.4.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, but the asynchronous clock output is disabled, that is CFG2[ADACKEN]=0, the ADACK clock generator also remains in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled, that is, CFG2[ADACKEN]=1, it remains active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting CFG1[ADLPC]. This results in a lower maximum value for f_{ADCK}.

28.4.4.5 Sample time and total conversion time

For short sample, that is, when CFG1[ADLSMP]=0, there is a 2-cycle adder for first conversion over the base sample time of four ADCK cycles. For high-speed conversions, that is, when CFG2[ADHSC]=1, there is an additional 2-cycle adder on any conversion. The table below summarizes sample times for the possible ADC configurations.

ADC configuration			Sample time (ADCK cycles)	
CFG1[ADLSMP]	CFG2[ADLSTS]	CFG2[ADHSC]	First or Single	Subsequent
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	
1	11	1	8	

The total conversion time depends upon:

- The sample time as determined by CFG1[ADLSMP] and CFG2[ADLSTS]
- The MCU bus frequency
- The conversion mode, as determined by CFG1[MODE] and SC1n[DIFF]
- The high-speed configuration, that is, CFG2[ADHSC]
- The frequency of the conversion clock, that is, f_{ADCK}.

CFG2[ADHSC] is used to configure a higher clock input frequency. This will allow faster overall conversion times. To meet internal ADC timing requirements, CFG2[ADHSC] adds additional ADCK cycles. Conversions with CFG2[ADHSC]=1 take two more ADCK cycles. CFG2[ADHSC] must be used when the ADCLK exceeds the limit for CFG2[ADHSC]=0.

After the module becomes active, sampling of the input begins.

1. CFG1[ADLSMP] and CFG2[ADLSTS] select between sample times based on the conversion mode that is selected.
2. When sampling is completed, the converter is isolated from the input channel and a successive approximation algorithm is applied to determine the digital value of the analog signal.
3. The result of the conversion is transferred to Rn upon completion of the conversion algorithm.

If the bus frequency is less than f_{ADCK}, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled, that is, when CFG1[ADLSMP]=0.

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by CFG1[ADICLK], and the divide ratio is specified by CFG1[ADIV].

The maximum total conversion time for all configurations is summarized in the equation below. See the following tables for the variables referenced in the equation.

$$\mathrm{ConversionTime} = \mathrm{SFCAdder} + \mathrm{AverageNum} \times \left(\mathrm{BCT} + \mathrm{LSTAdder} + \mathrm{HSCAdder}\right)$$

Equation 1. Conversion time equation

Table 28-3. Single or first continuous time adder (SFCAdder)

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles[1]
1	0	11	5 μs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles[1]
0	0	11	5 μs + 5 ADCK cycles + 5 bus clock cycles

[1] To achieve this time, CFG2[ADACKEN] must be 1 for at least 5 μs prior to the conversion is initiated.

Table 28-4. Average number factor (AverageNum)

SC3[AVGE]	SC3[AVGS]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 28-5. Base conversion time (BCT)

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
9b differential	27 ADCK cycles
10b single-ended	20 ADCK cycles
11b differential	30 ADCK cycles
12b single-ended	20 ADCK cycles
13b differential	30 ADCK cycles

Table continues on the next page...

Table 28-5. Base conversion time (BCT) (continued)

Mode	Base conversion time (BCT)
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Table 28-6. Long sample time adder (LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 28-7. High-speed conversion time adder (HSCAdder)

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

28.4.4.6 Conversion time examples

The following examples use the Equation 1 on page 532, and the information provided in Table 28-3 through Table 28-7.

28.4.4.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is:

- 10-bit mode, with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 8 MHz
- Long sample time disabled
- High-speed conversion disabled

The conversion time for a single conversion is calculated by using the Equation 1 on page 532, and the information provided in Table 28-3 through Table 28-7. The table below lists the variables of Equation 1 on page 532.

Table 28-8. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	20 ADCK cycles
LSTAdder	0
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for a bus clock and an ADCK frequency equal to 8 MHz, the resulting conversion time is 3.75 µs.

28.4.4.6.2 Long conversion time configuration

A configuration for long ADC conversion is:

- 16-bit differential mode with the bus clock selected as the input clock source
- The input clock divide-by-8 ratio selected
- Bus frequency of 8 MHz
- Long sample time enabled
- Configured for longest adder
- High-speed conversion disabled
- Average enabled for 32 conversions

The conversion time for this conversion is calculated by using the Equation 1 on page 532, and the information provided in Table 28-3 through Table 28-7. The following table lists the variables of the Equation 1 on page 532.

Table 28-9. Typical conversion time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	34 ADCK cycles
LSTAdder	20 ADCK cycles
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock equal to 8 MHz and ADCK equal to 1 MHz, the resulting conversion time is 57.625 µs, that is, AverageNum. This results in a total conversion time of 1.844 ms.

28.4.4.6.3 Short conversion time configuration

A configuration for short ADC conversion is:

- 8-bit Single-Ended mode with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 20 MHz
- Long sample time disabled
- High-speed conversion enabled

The conversion time for this conversion is calculated by using the Equation 1 on page 532, and the information provided in Table 28-3 through Table 28-7. The table below lists the variables of Equation 1 on page 532.

Table 28-10. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

The resulting conversion time is generated using the parameters listed in in the preceding table. Therefore, for bus clock and ADCK frequency equal to 20 MHz, the resulting conversion time is 1.45 µs.

28.4.4.7 Hardware average function

The hardware average function can be enabled by setting SC3[AVGE]=1 to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which can select 4, 8, 16, or 32 conversions to be averaged. While the hardware average function is in progress, SC2[ADACT] will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions have been completed. When hardware averaging is selected, the completion of a single conversion will not set SC1n[COCO].

If the compare function is either disabled or evaluates true, after the selected number of conversions are completed, the average conversion result is transferred into the data result registers, Rn, and SC1n[COCO] is set. An ADC interrupt is generated upon the setting of SC1n[COCO] if the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

> The hardware average function can perform conversions on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the hardware average is completed if SC1n[AIEN] is set.

28.4.5 Automatic compare function

The compare function can be configured to check whether the result is less than or greater-than-or-equal-to a single compare value, or, if the result falls within or outside a range determined by two compare values.

The compare mode is determined by SC2[ACFGT], SC2[ACREN], and the values in the compare value registers, CV1 and CV2. After the input is sampled and converted, the compare values in CV1 and CV2 are used as described in the following table. There are six Compare modes as shown in the following table.

Table 28-11. Compare modes

SC2[ACFGT]	SC2[ACREN]	ADCCV1 relative to ADCCV2	Function	Compare mode description
0	0	—	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	—	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 **Or** the result is greater than CV2.
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 **And** the result is greater than CV2.
1	1	Less than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 **And** the result is less than or equal to CV2.
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 **Or** the result is less than or equal to CV2.

With SC2[ACREN] =1, and if the value of CV1 is less than or equal to the value of CV2, then setting SC2[ACFGT] will select a trigger-if-inside-compare-range inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-outside-compare-range, not-inclusive-of-endpoints function.

If CV1 is greater than CV2, setting SC2[ACFGT] will select a trigger-if-outside-compare-range, inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-inside-compare-range, not-inclusive-of-endpoints function.

If the condition selected evaluates true, SC1n[COCO] is set.

Upon completion of a conversion while the compare function is enabled, if the compare condition is not true, SC1n[COCO] is not set and the conversion result data will not be transferred to the result register, Rn. If the hardware averaging function is enabled, the compare function compares the averaged result to the compare values. The same compare function definitions apply. An ADC interrupt is generated when SC1n[COCO] is set and the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

> The compare function can monitor the voltage on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the compare condition is met.

28.4.6 Calibration function

The ADC contains a self-calibration function that is required to achieve the specified accuracy.

Calibration must be run, or valid calibration values written, after any reset and before a conversion is initiated. The calibration function sets the offset calibration value, the minus-side calibration values, and the plus-side calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), and the plus-side and minus-side calibration values are automatically stored in the ADC plus-side and minus-side calibration registers, CLPx and CLMx. The user must configure the ADC correctly prior to calibration, and must generate the plus-side and minus-side gain calibration results and store them in the ADC plus-side gain register (PG) after the calibration function completes.

Prior to calibration, the user must configure the ADC's clock source and frequency, low power configuration, voltage reference selection, sample time, and high speed configuration according to the application's clock source availability and needs. If the application uses the ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected, or multiple calibrations can be done for the different configurations. For best calibration results:

- Set hardware averaging to maximum, that is, SC3[AVGE]=1 and SC3[AVGS]=11 for an average of 32
- Set ADC clock frequency f_{ADCK} less than or equal to 4 MHz
- $V_{REFH}=V_{DDA}$
- Calibrate at nominal voltage and temperature

The input channel, conversion mode continuous function, compare function, resolution mode, and differential/single-ended mode are all ignored during the calibration function.

To initiate calibration, the user sets SC3[CAL] and the calibration will automatically begin if the SC2[ADTRG] is 0. If SC2[ADTRG] is 1, SC3[CAL] will not get set and SC3[CALF] will be set. While calibration is active, no ADC register can be written and no stop mode may be entered, or the calibration routine will be aborted causing SC3[CAL] to clear and SC3[CALF] to set. At the end of a calibration sequence, SC1n[COCO] will be set. SC1n[AIEN] can be used to allow an interrupt to occur at the end of a calibration sequence. At the end of the calibration routine, if SC3[CALF] is not set, the automatic calibration routine is completed successfully.

To complete calibration, the user must generate the gain calibration values using the following procedure:

1. Initialize or clear a 16-bit variable in RAM.
2. Add the plus-side calibration results CLP0, CLP1, CLP2, CLP3, CLP4, and CLPS to the variable.
3. Divide the variable by two.
4. Set the MSB of the variable.
5. The previous two steps can be achieved by setting the carry bit, rotating to the right through the carry bit on the high byte and again on the low byte.
6. Store the value in the plus-side gain calibration register PG.
7. Repeat the procedure for the minus-side gain calibration value.

When calibration is complete, the user may reconfigure and use the ADC as desired. A second calibration may also be performed, if desired, by clearing and again setting SC3[CAL].

Overall, the calibration routine may take as many as 14k ADCK cycles and 100 bus cycles, depending on the results and the clock source chosen. For an 8 MHz clock source, this length amounts to about 1.7 ms. To reduce this latency, the calibration values, which are offset, plus-side and minus-side gain, and plus-side and minus-side calibration values, may be stored in flash memory after an initial calibration and recovered prior to the first ADC conversion. This method can reduce the calibration latency to 20 register store operations on all subsequent power, reset, or Low-Power Stop mode recoveries.

Further information on the calibration procedure can be found in the Calibration section of AN3949: ADC16 Calibration Procedure and Programmable Delay Block Synchronization.

28.4.7 User-defined offset function

OFS contains the user-selected or calibration-generated offset error correction value.

This register is a 2's complement, left-justified. The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

The formatting of the OFS is different from the data result register, Rn, to preserve the resolution of the calibration value regardless of the conversion mode selected. Lower order bits are ignored in lower resolution modes. For example, in 8-bit single-ended mode, OFS[14:7] are subtracted from D[7:0]; OFS[15] indicates the sign (negative numbers are effectively added to the result) and OFS[6:0] are ignored. The same bits are used in 9-bit differential mode because OFS[15] indicates the sign bit, which maps to D[8]. For 16-bit differential mode, OFS[15:0] are directly subtracted from the conversion result data D[15:0]. In 16-bit single-ended mode, there is no field in the OFS corresponding to the least significant result D[0], so odd values, such as -1 or +1, cannot be subtracted from the result.

OFS is automatically set according to calibration requirements once the self-calibration sequence is done, that is, SC3[CAL] is cleared. The user may write to OFS to override the calibration result if desired. If the OFS is written by the user to a value that is different from the calibration value, the ADC error specifications may not be met. Storing the value generated by the calibration function in memory before overwriting with a user-specified value is recommended.

Note

> There is an effective limit to the values of offset that can be set by the user. If the magnitude of the offset is too high, the results of the conversions will cap off at the limits.

The offset calibration function may be employed by the user to remove application offsets or DC bias values. OFS may be written with a number in 2's complement format and this offset will be subtracted from the result, or hardware averaged value. To add an offset, store the negative offset in 2's complement format and the effect will be an addition. An offset correction that results in an out-of-range value will be forced to the minimum or maximum value. The minimum value for single-ended conversions is 0x0000; for a differential conversion it is 0x8000.

To preserve accuracy, the calibrated offset value initially stored in OFS must be added to the user-defined offset. For applications that may change the offset repeatedly during operation, store the initial offset calibration value in flash so it can be recovered and added to any user offset adjustment value and the sum stored in OFS.

28.4.8 Temperature sensor

The ADC module includes a temperature sensor whose output is connected to one of the ADC analog channel inputs.

The following equation provides an approximate transfer function of the temperature sensor.

$$\mathrm{Temp} = 25 - \left(\left(V_{TEMP} - V_{TEMP25}\right) \div \mathrm{m}\right)$$

Equation 2. Approximate transfer function of the temperature sensor

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25 °C.
- m is referred as temperature sensor slope in the device data sheet. It is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and temperature sensor slope values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in the preceding equation. If V_{TEMP} is less than V_{TEMP25}, the hot slope value is applied in the preceding equation. ADC Electricals table may only specify one temperature sensor slope value. In that case, the user could use the same slope for the calculation across the operational temperature range.

For more information on using the temperature sensor, see the application note titled *Temperature Sensor for the HCS08 Microcontroller Family* (document AN3031).

28.4.9 MCU wait mode operation

Wait mode is a lower-power consumption Standby mode from which recovery is fast because the clock sources remain active.

If a conversion is in progress when the MCU enters Wait mode, it continues until completion. Conversions can be initiated while the MCU is in Wait mode by means of the hardware trigger or if continuous conversions are enabled.

The bus clock, bus clock divided by two; and ADACK are available as conversion clock sources while in Wait mode. The use of ALTCLK as the conversion clock source in Wait is dependent on the definition of ALTCLK for this MCU. See the Chip Configuration information on ALTCLK specific to this MCU.

If the compare and hardware averaging functions are disabled, a conversion complete event sets SC1n[COCO] and generates an ADC interrupt to wake the MCU from Wait mode if the respective ADC interrupt is enabled, that is, when SC1n[AIEN]=1. If the hardware averaging function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, when the selected number of conversions are completed. If the compare function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, only if the compare conditions are met. If a single conversion is selected and the compare trigger is not met, the ADC will return to its idle state and cannot wake the MCU from Wait mode unless a new conversion is initiated by the hardware trigger.

28.4.10 MCU Normal Stop mode operation

Stop mode is a low-power consumption Standby mode during which most or all clock sources on the MCU are disabled.

28.4.10.1 Normal Stop mode with ADACK disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its Idle state. The contents of the ADC registers, including Rn, are unaffected by Normal Stop mode. After exiting from Normal Stop mode, a software or hardware trigger is required to resume conversions.

28.4.10.2 Normal Stop mode with ADACK enabled

If ADACK is selected as the conversion clock, the ADC continues operation during Normal Stop mode. See the chip-specific ADC information for configuration information for this device.

If a conversion is in progress when the MCU enters Normal Stop mode, it continues until completion. Conversions can be initiated while the MCU is in Normal Stop mode by means of the hardware trigger or if continuous conversions are enabled.

If the compare and hardware averaging functions are disabled, a conversion complete event sets SC1n[COCO] and generates an ADC interrupt to wake the MCU from Normal Stop mode if the respective ADC interrupt is enabled, that is, when SC1n[AIEN]=1. The result register, Rn, will contain the data from the first completed conversion that occurred during Normal Stop mode. If the hardware averaging function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, when the selected number of conversions are completed. If the compare function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, only if the compare conditions are met. If a single conversion is selected and the compare is not true, the ADC will return to its idle state and cannot wake the MCU from Normal Stop mode unless a new conversion is initiated by another hardware trigger.

28.4.11 MCU Low-Power Stop mode operation

The ADC module is automatically disabled when the MCU enters Low-Power Stop mode.

All module registers contain their reset values following exit from Low-Power Stop mode. Therefore, the module must be re-enabled and re-configured following exit from Low-Power Stop mode.

NOTE

For the chip specific modes of operation, see the power management information for the device.

28.5 Initialization information

This section gives an example that provides some basic direction on how to initialize and configure the ADC module.

The user can configure the module for 16-bit, 12-bit, 10-bit, or 8-bit single-ended resolution or 16-bit, 13-bit, 11-bit, or 9-bit differential resolution, single or continuous conversion, and a polled or interrupt approach, among many other options. For information used in this example, refer to Table 28-6, Table 28-7, and Table 28-8.

Note

> Hexadecimal values are designated by a preceding 0x, binary values designated by a preceding %, and decimal values have no preceding character.

28.5.1 ADC module initialization example

28.5.1.1 Initialization sequence

Before the ADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is:

1. Calibrate the ADC by following the calibration instructions in Calibration function.
2. Update CFG to select the input clock source and the divide ratio used to generate ADCK. This register is also used for selecting sample time and low-power configuration.
3. Update SC2 to select the conversion trigger, hardware or software, and compare function options, if enabled.
4. Update SC3 to select whether conversions will be continuous or completed only once (ADCO) and whether to perform hardware averaging.
5. Update SC1:SC1n registers to select whether conversions will be single-ended or differential and to enable or disable conversion complete interrupts. Also, select the input channel which can be used to perform conversions.

28.5.1.2 Pseudo-code example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low-power with a long sample time on input channel 1, where ADCK is derived from the bus clock divided by 1.

CFG1 = 0x98 (%10011000)

```
        Bit 7   ADLPC   1      Configures for low power, lowers maximum clock speed.
        Bit 6:5 ADIV    00     Sets the ADCK to the input clock ÷ 1.
        Bit 4   ADLSMP  1      Configures for long sample time.
        Bit 3:2   MODE   10    Selects the single-ended 10-bit conversion, differential 11-
bit conversion.
        Bit 1:0   ADICLK   00    Selects the bus clock.
```

SC2 = 0x00 (%00000000)

```
        Bit 7   ADACT   0       Flag indicates if a conversion is in progress.
        Bit 6   ADTRG   0       Software trigger selected.
        Bit 5   ACFE    0       Compare function disabled.
        Bit 4   ACFGT   0       Not used in this example.
        Bit 3   ACREN   0      Compare range disabled.
 Bit 2 DMAEN 0 DMA request disabled.
        Bit 1:0 REFSEL  00     Selects default voltage reference pin pair (External pins
VREFH and VREFL).
```

SC1A = 0x41 (%01000001)

```
        Bit 7   COCO   0       Read-only flag which is set when a conversion completes.
        Bit 6   AIEN   1       Conversion complete interrupt enabled.
 Bit 5 DIFF 0 Single-ended conversion selected.
        Bit 4:0 ADCH   00001   Input channel 1 selected as ADC input channel.
```

RA = 0xxx

```
        Holds results of conversion.
```

CV = 0xxx

```
        Holds compare value when compare function enabled.
```

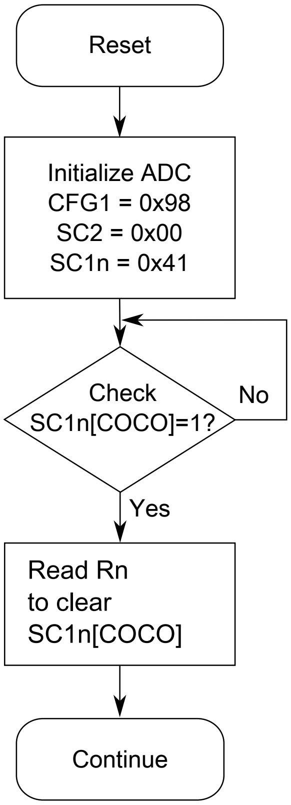


Figure 28-2. Initialization flowchart example

28.6 Application information

The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an ADC.

For guidance on selecting optimum external component values and converter parameters see AN4373: Cookbook for SAR ADC Measurements.

28.6.1 External pins and routing

28.6.1.1 Analog supply pins

Depending on the device, the analog power and ground supplies, V_{DDA} and V_{SSA}, of the ADC module are available as:

- V_{DDA} and V_{SSA} available as separate pins—When available on a separate pin, both V_{DDA} and V_{SSA} must be connected to the same voltage potential as their corresponding MCU digital supply, V_{DD} and V_{SS}, and must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

- V_{SSA} is shared on the same pin as the MCU digital V_{SS}.
- V_{SSA} and V_{DDA} are shared with the MCU digital supply pins—In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSA} pin. This must be the only ground connection between these supplies, if possible. V_{SSA} makes a good single point ground location.

28.6.1.2 Analog voltage reference pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs used by the converter:

- V_{REFSH} is the high reference voltage for the converter.
- V_{REFSL} is the low reference voltage for the converter.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference and a ground reference. The two pairs are external, V_{REFH} and V_{REFL} and alternate, V_{ALTH} and V_{ALTL}. These voltage references are selected using SC2[REFSEL]. The alternate voltage reference pair, V_{ALTH} and V_{ALTL}, may select additional external pins or internal sources based on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

In some packages, the external or alternate pairs are connected in the package to V_{DDA} and V_{SSA}, respectively. One of these positive references may be shared on the same pin as V_{DDA} on some devices. One of these ground references may be shared on the same pin as V_{SSA} on some devices.

If externally available, the positive reference may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. The positive reference must never exceed V_{DDA}. If externally available, the ground reference must be connected to the same voltage potential as V_{SSA}. The voltage reference pairs must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 μF capacitor with good high-frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum, that is, parasitic only.

28.6.1.3 Analog input pins

The external analog inputs are typically shared with digital I/O pins on MCU devices.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used, they must be placed as near as possible to the package pins and be referenced to V_{SSA}.

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL}. If the input is equal to or exceeds V_{REFH}, the converter circuit converts the signal to 0xFFF, which is full scale 12-bit representation, 0x3FF, which is full scale 10-bit representation, or 0xFF, which is full scale 8-bit representation. If the input is equal to or less than V_{REFL}, the converter circuit converts it to 0x000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There is a brief current associated with V_{REFL} when the sampling capacitor is charging.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins must not be transitioning during conversions.

28.6.2 Sources of error

28.6.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy.

$$\mathrm{RAS} + \mathrm{RADIN} = \mathrm{SC} / (\mathrm{FMAX} * \mathrm{NUMTAU} * \mathrm{CADIN})$$

Figure 28-3. Sampling equation

Where:

RAS = External analog source resistance

SC = Number of ADCK cycles used during sample window

CADIN = Internal ADC input capacitance

NUMTAU = -ln(LSBERR / 2^N)

LSBERR = value of acceptable sampling error in LSBs

N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode or 16 in 16-bit mode

Higher source resistances or higher-accuracy sampling is possible by setting CFG1[ADLSMP] and changing CFG2[ADLSTS] to increase the sample window, or decreasing ADCK frequency to increase sample time.

28.6.2.2 Pin leakage error

Leakage on the I/O pins can cause conversion error if the external analog source resistance, R_{AS}, is high. If this error cannot be tolerated by the application, keep R_{AS} lower than $V_{REFH} / (4 \times I_{LEAK} \times 2^N)$ for less than 1/4 LSB leakage error, where N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode, or 16 in 16-bit mode.

28.6.2.3 Noise-induced errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 µF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 µF low-ESR capacitor from V_{DDA} to V_{SSA}.
- If inductive isolation is used from the primary supply, an additional 1 µF capacitor is placed from V_{DDA} to V_{SSA}.
- V_{SSA}, and V_{REFL}, if connected, is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in Wait or Normal Stop mode before initiating (hardware-triggered conversions) or immediately after initiating (hardware- or software-triggered conversions) the ADC conversion.

  - For software triggered conversions, immediately follow the write to SC1 with a Wait instruction or Stop instruction.
  - For Normal Stop mode operation, select ADACK as the clock source. Operation in Normal Stop reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in Wait or Normal Stop mode, or I/O activity cannot be halted, the following actions may reduce the effect of noise on the accuracy:

- Place a 0.01 μF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSA}. This improves noise issues, but affects the sample rate based on the external analog source resistance.
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1 LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock, that is, ADACK, and averaging. Noise that is synchronous to ADCK cannot be averaged out.

28.6.2.4 Code width and quantization error

The ADC quantizes the ideal straight-line transfer function into 65536 steps in the 16-bit mode.. Each step ideally has the same height, that is, 1 code, and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N-bit converter, where N can be 16, 12, 10, or 8, defined as 1 LSB, is:

$$1\mathrm{LSB}=\left(V_{REFH}\right)/2^{N}$$

Equation 3. Ideal code width for an N-bit converter

There is an inherent quantization error due to the digitization of the result. For 8-bit, 10-bit, or 12-bit conversions, the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2 LSB in 8-bit, 10-bit, or 12-bit modes. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 LSB and the code width of the last (0xFF or 0x3FF) is 1.5 LSB.

For 16-bit conversions, the code transitions only after the full code width is present, so the quantization error is -1 LSB to 0 LSB and the code width of each step is 1 LSB.

28.6.2.5 Linearity errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors, but the system designers must be aware of these errors because they affect overall accuracy:

- Zero-scale error (E_{ZS}), sometimes called offset: This error is defined as the difference between the actual code width of the first conversion and the ideal code width. This is 1/2 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal (1 LSB) is used.
- Full-scale error (E_{FS}): This error is defined as the difference between the actual code width of the last conversion and the ideal code width. This is 1.5 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the last conversion is 0x3FE, the difference between the actual 0x3FE code width and its ideal (1 LSB) is used.
- Differential non-linearity (DNL): This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral non-linearity (INL): This error is defined as the highest-value or absolute value that the running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE): This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

28.6.2.6 Code jitter, non-monotonicity, and missing codes

Analog-to-digital converters are susceptible to three special forms of error:

- Code jitter: Code jitter occurs when a given input voltage converts to one of the two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code, and vice-versa. However, even small amounts of system noise can cause the converter to be indeterminate, between two codes, for a range of input voltages around the transition voltage.

This error may be reduced by repeatedly sampling the input and averaging the result. Additionally, the techniques discussed in Noise-induced errors reduces this error.

- Non-monotonicity: Non-monotonicity occurs when, except for code jitter, the converter converts to a lower code for a higher input voltage.
- Missing codes: Missing codes are those values never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and have no missing codes.

Chapter 29
Comparator (CMP)

The Comparator module (CMP) provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full supply voltage range (rail-to-rail operation) and supports programmable hysteresis control. The output of the comparator can be samples, windowed, or digitally filtered.

29.1 Introduction

The comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 6-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The 6-bit DAC is 64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed. The 64-tap resistor ladder network divides the supply reference V_{in} into 64 voltage levels. A 6-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/64$. V_{in} can be selected from two voltage sources, V_{in1} and V_{in2}. The 6-bit DAC from a comparator is available as an on-chip internal signal only and is not available externally to a pin.

29.1.1 CMP features

The CMP has the following features:

- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control

- Selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output
- Selectable inversion on comparator output
- Capability to produce a wide range of outputs such as:
  - Sampled
  - Digitally filtered:
    - Filter can be bypassed
    - Can be clocked via scaled bus clock
- External hysteresis can be used at the same time that the output filter is used for internal functions
- Two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with longer propagation delay
- DMA transfer support
  - A comparison event can be selected to trigger a DMA transfer
- Functional in all modes of operation except VLLS0
- The filter functions are not available in the following modes:
  - Stop
  - VLPS
  - LLS
  - VLLSx

29.1.2 6-bit DAC key features

The 6-bit DAC has the following features:

- 6-bit resolution
- Selectable supply reference source
- Power Down mode to conserve power when not in use
- Option to route the output to internal comparator input

29.1.3 ANMUX key features

The ANMUX has the following features:

- Two 8-to-1 channel mux
- Operational over the entire supply range

29.1.4 CMP, DAC and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

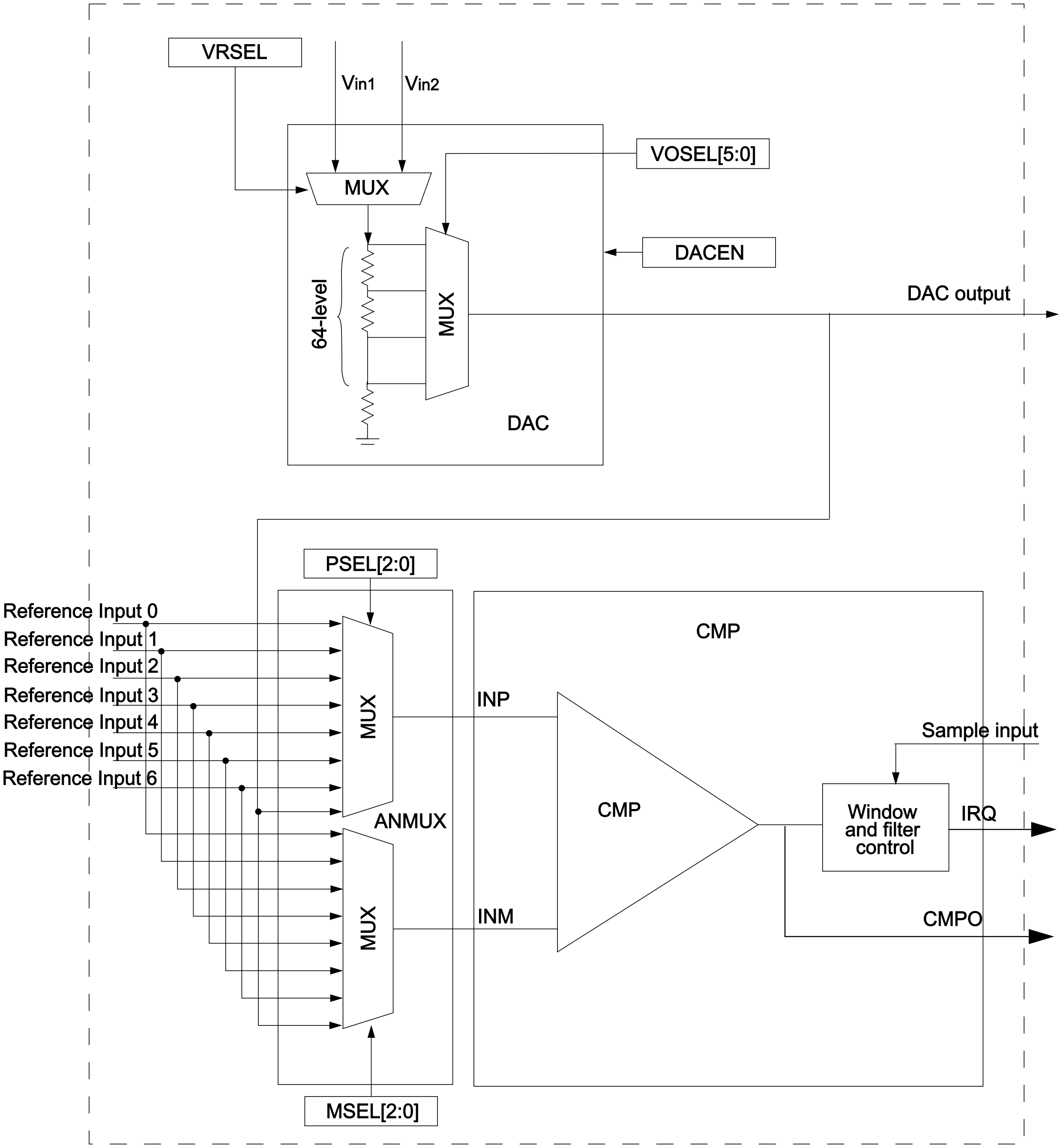


Figure 29-1. CMP, DAC and ANMUX block diagram

29.1.5 CMP block diagram

The following figure shows the block diagram for the CMP module.

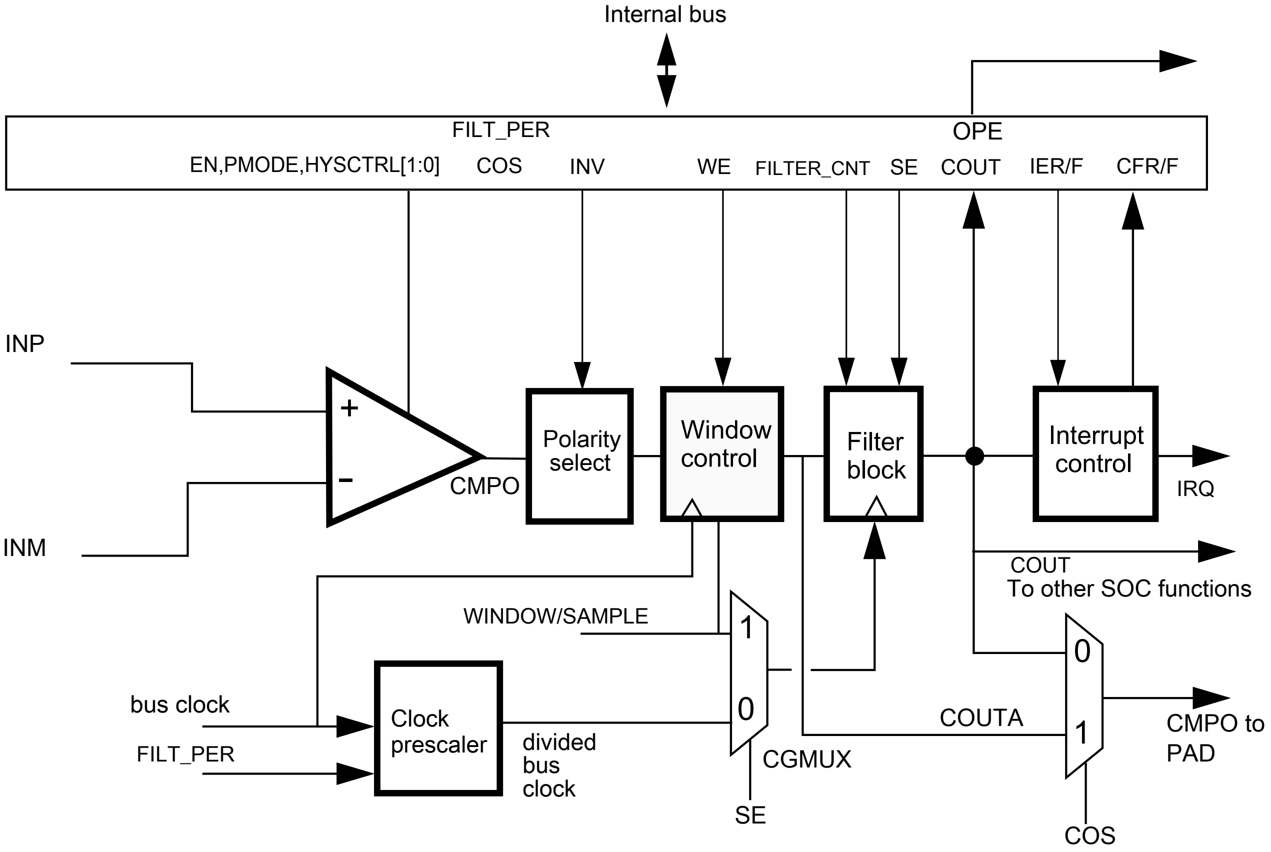


Figure 29-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when CR1[WE] = 0
- The Filter block is bypassed when not in use.
- The Filter block acts as a simple sampler if the filter is bypassed and CR0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and CR0[FILTER_CNT] is set greater than 0x01.
  - CR1[SE] = 0, the divided bus clock is used as sampling clock
- If enabled, the Filter block will incur up to one bus clock additional latency penalty on COUT due to the fact that COUT, which is crossing clock domain boundaries, must be resynchronized to the bus clock.

29.2 Memory map/register definitions

CMP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_3000	CMP Control Register 0 (CMP0_CR0)	8	R/W	00h	29.2.1/558
4007_3001	CMP Control Register 1 (CMP0_CR1)	8	R/W	00h	29.2.2/559
4007_3002	CMP Filter Period Register (CMP0_FPR)	8	R/W	00h	29.2.3/560
4007_3003	CMP Status and Control Register (CMP0_SCR)	8	R/W	00h	29.2.4/561
4007_3004	DAC Control Register (CMP0_DACCR)	8	R/W	00h	29.2.5/562
4007_3005	MUX Control Register (CMP0_MUXCR)	8	R/W	00h	29.2.6/562

29.2.1 CMP Control Register 0 (CMPx_CR0)

Address: 4007_3000h base + 0h offset = 4007_3000h

Bit	7	6	5	4	3	2	1	0
Read	0	FILTER_CNT			0	0	HYSTCTR	
Write	▨				▨	▨		
Reset	0	0	0	0	0	0	0	0

CMPx_CR0 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 FILTER_CNT	Filter Sample Count Represents the number of consecutive samples that must agree prior to the comparator ouput filter accepting a new output state. For information regarding filter programming and latency, see the Functional description. 000 Filter is disabled. SE = 0, COUT = COUTA. 001 One sample must agree. The comparator output is simply sampled. 010 2 consecutive samples must agree. 011 3 consecutive samples must agree. 100 4 consecutive samples must agree. 101 5 consecutive samples must agree. 110 6 consecutive samples must agree. 111 7 consecutive samples must agree.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HYSTCTR	Comparator hard block hysteresis control Defines the programmable hysteresis level. The hysteresis values associated with each level are device-specific. See the Data Sheet of the device for the exact values. 00 Level 0

Table continues on the next page...

CMPx_CR0 field descriptions (continued)

Field	Description
	01 Level 1 10 Level 2 11 Level 3

29.2.2 CMP Control Register 1 (CMPx_CR1)

Address: 4007_3000h base + 1h offset = 4007_3001h

Bit	7	6	5	4	3	2	1	0
Read Write	SE	WE	TRIGM	PMODE	INV	COS	OPE	EN
Reset	0	0	0	0	0	0	0	0

CMPx_CR1 field descriptions

Field	Description
7 SE	Sample Enable SE must be clear to 0 and usage of sample operation is limited to a divided version of the bus clock. 0 Sampling mode is not selected. 1 Sampling mode is selected.
6 WE	Windowing Enable The CMP does not support window compare function and a 0 must always be written to WE. 0 Windowing mode is not selected. 1 Windowing mode is selected.
5 TRIGM	Trigger Mode Enable CMP and DAC are configured to CMP Trigger mode when CMP_CR1[TRIGM] is set to 1. In addition, the CMP should be enabled. If the DAC is to be used as a reference to the CMP, it should also be enabled. CMP Trigger mode depends on an external timer resource to periodically enable the CMP and 6-bit DAC in order to generate a triggered compare. Upon setting TRIGM, the CMP and DAC are placed in a standby state until an external timer resource trigger is received. See the chip configuration for details about the external timer resource. 0 Trigger mode is disabled. 1 Trigger mode is enabled.
4 PMODE	Power Mode Select See the electrical specifications table in the device Data Sheet for details. 0 Low-Speed (LS) Comparison mode selected. In this mode, CMP has slower output propagation delay and lower current consumption. 1 High-Speed (HS) Comparison mode selected. In this mode, CMP has faster output propagation delay and higher current consumption.

Table continues on the next page...

CMPx_CR1 field descriptions (continued)

Field	Description
3 INV	Comparator INVERT Allows selection of the polarity of the analog comparator function. It is also driven to the COUT output, on both the device pin and as SCR[COUT], when OPE=0. 0 Does not invert the comparator output. 1 Inverts the comparator output.
2 COS	Comparator Output Select 0 Set the filtered comparator output (CMPO) to equal COUT. 1 Set the unfiltered comparator output (CMPO) to equal COUTA.
1 OPE	Comparator Output Pin Enable 0 CMPO is not available on the associated CMPO output pin. If the comparator does not own the pin, this field has no effect. 1 CMPO is available on the associated CMPO output pin. The comparator output (CMPO) is driven out on the associated CMPO output pin if the comparator owns the pin. If the comparator does not own the field, this bit has no effect.
0 EN	Comparator Module Enable Enables the Analog Comparator module. When the module is not enabled, it remains in the off state, and consumes no power. When the user selects the same input from analog mux to the positive and negative port, the comparator is disabled automatically. 0 Analog Comparator is disabled. 1 Analog Comparator is enabled.

29.2.3 CMP Filter Period Register (CMPx_FPR)

Address: 4007_3000h base + 2h offset = 4007_3002h

Bit	7	6	5	4	3	2	1	0
Read Write	FILT_PER							
Reset	0	0	0	0	0	0	0	0

CMPx_FPR field descriptions

Field	Description
FILT_PER	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when CR1[SE]=0. Setting FILT_PER to 0x0 disables the filter. Filter programming and latency details appear in the Functional description.

29.2.4 CMP Status and Control Register (CMPx_SCR)

Address: 4007_3000h base + 3h offset = 4007_3003h

Bit	7	6	5	4	3	2	1	0
Read	0	DMAEN	0	IER	IEF	CFR	CFF	COUT
Write	▨		▨			w1c	w1c	▨
Reset	0	0	0	0	0	0	0	0

CMPx_SCR field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DMAEN	DMA Enable Control Enables the DMA transfer triggered from the CMP module. When this field is set, a DMA request is asserted when CFR or CFF is set. 0 DMA is disabled. 1 DMA is enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from the CMP. When this field is set, an interrupt will be asserted when CFR is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from the CMP. When this field is set, an interrupt will be asserted when CFF is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 CFR	Analog Comparator Flag Rising Detects a rising-edge on COUT, when set, during normal operation. CFR is cleared by writing 1 to it. During Stop modes, CFR is level sensitive . 0 Rising-edge on COUT has not been detected. 1 Rising-edge on COUT has occurred.
1 CFF	Analog Comparator Flag Falling Detects a falling-edge on COUT, when set, during normal operation. CFF is cleared by writing 1 to it. During Stop modes, CFF is level sensitive . 0 Falling-edge on COUT has not been detected. 1 Falling-edge on COUT has occurred.

Table continues on the next page...

CMPx_SCR field descriptions (continued)

Field	Description
0 COUT	Analog Comparator Output Returns the current value of the Analog Comparator output, when read. The field is reset to 0 and will read as CR1[INV] when the Analog Comparator module is disabled, that is, when CR1[EN] = 0. Writes to this field are ignored.

29.2.5 DAC Control Register (CMPx_DACCR)

Address: 4007_3000h base + 4h offset = 4007_3004h

Bit	7	6	5	4	3	2	1	0
Read Write	DACEN	VRSEL	VOSEL					
Reset	0	0	0	0	0	0	0	0

CMPx_DACCR field descriptions

Field	Description
7 DACEN	DAC Enable Enables the DAC. When the DAC is disabled, it is powered down to conserve power. 0 DAC is disabled. 1 DAC is enabled.
6 VRSEL	Supply Voltage Reference Source Select 0 V_{in1} is selected as resistor ladder network supply reference. 1 V_{in2} is selected as resistor ladder network supply reference.
VOSEL	DAC Output Voltage Select Selects an output voltage from one of 64 distinct levels. `DACO = (V in /64) * (VOSEL[5:0] + 1)` , so the DACO range is from V_{in}/64 to V_{in}.

29.2.6 MUX Control Register (CMPx_MUXCR)

Address: 4007_3000h base + 5h offset = 4007_3005h

Bit	7	6	5	4	3	2	1	0
Read	PSTM	0	PSEL			MSEL		
Write		▨						
Reset	0	0	0	0	0	0	0	0

CMPx_MUXCR field descriptions

Field	Description
7 PSTM	Pass Through Mode Enable This bit is used to enable to MUX pass through mode. Pass through mode is always available but for some devices this feature must be always disabled due to the lack of package pins. 0 Pass Through Mode is disabled. 1 Pass Through Mode is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–3 PSEL	Plus Input Mux Control Determines which input is selected for the plus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
MSEL	Minus Input Mux Control Determines which input is selected for the minus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7

29.3 Functional description

The CMP module can be used to compare two analog input voltages applied to INP and INM.

CMPO is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. This signal can be selectively inverted by setting CR1[INV] = 1.

SCR[IER] and SCR[IEF] are used to select the condition which will cause the CMP module to assert an interrupt to the processor. SCR[CFF] is set on a falling-edge and SCR[CFR] is set on rising-edge of the comparator output. The optionally filtered CMPO can be read directly through SCR[COUT].

29.3.1 CMP functional modes

There are the following main sub-blocks to the CMP module:

- The comparator itself
- The filter function

The filter, CR0[FILTER_CNT], can be clocked from an internal clock source only. The filter is programmable with respect to the number of samples that must agree before a change in the output is registered. In the simplest case, only one sample must agree. In this case, the filter acts as a simple sampler.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

Table 29-1. Comparator sample/filter controls

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation
1	0	X	X	X	X	**Disabled** See the Disabled mode (# 1).
2A	1	0	0	0x00	X	**Continuous Mode** See the Continuous mode (#s 2A & 2B).
2B	1	0	0	X	0x00	
3B	1	0	0	0x01	> 0x00	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3B).
4B	1	0	0	> 0x01	> 0x00	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4B).
All other combinations of CR1[EN], CR1[WE], CR1[SE], CR0[FILTER_CNT], and FPR[FILT_PER] are illegal.						

For cases where a comparator is used to drive a fault input, for example, for a motor-control module such as FTM, it must be configured to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

Note

> Filtering and sampling settings must be changed only after setting CR1[SE]=0 and CR0[FILTER_CNT]=0x00. This resets the filter to a known state.

29.3.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

29.3.1.2 Continuous mode (#s 2A & 2B)

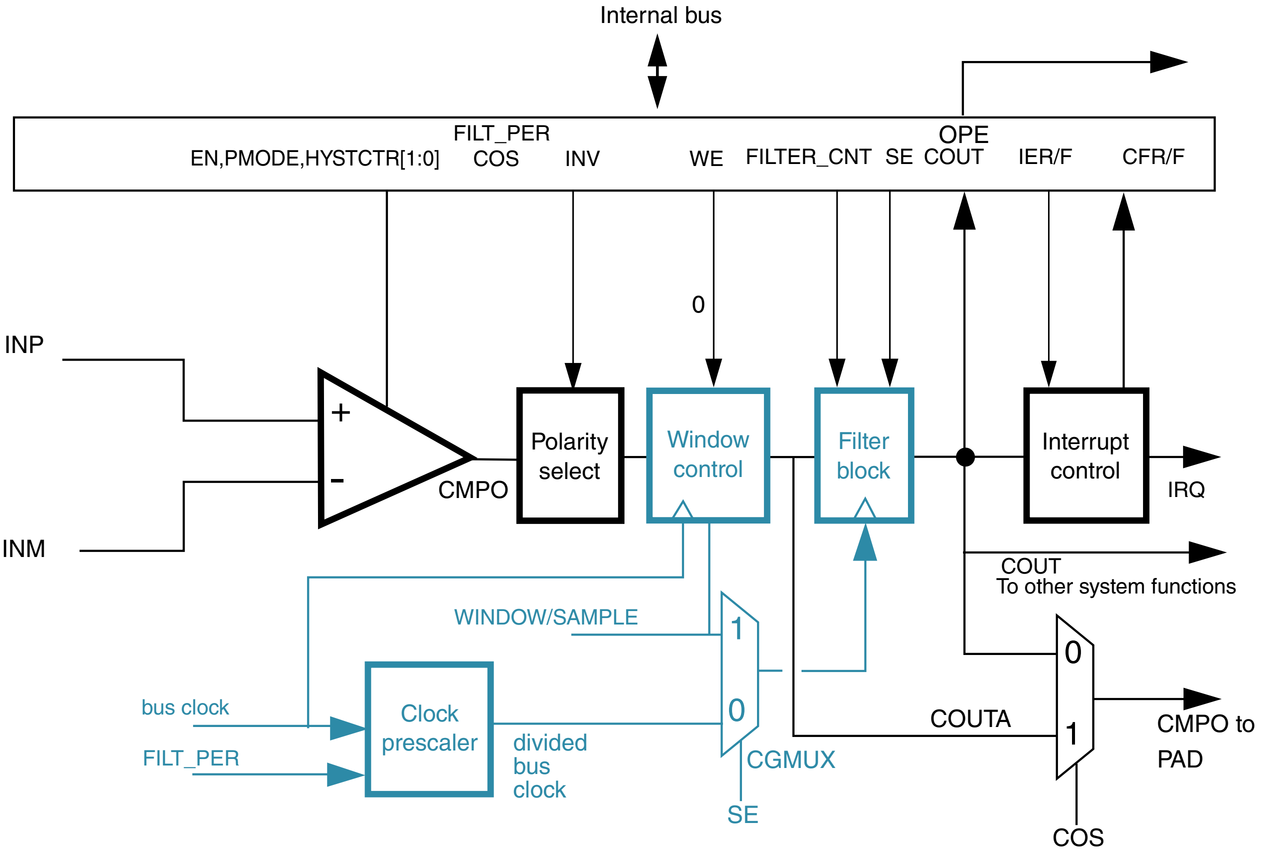


Figure 29-3. Comparator operation in Continuous mode

The analog comparator block is powered and active. CMPO may be optionally inverted, but is not subject to external sampling or filtering. Both window control and filter blocks are completely bypassed. SCR[COUT] is updated continuously. The path from comparator input pins to output pin is operating in combinational unclocked mode. COUT and COUTA are identical.

For control configurations which result in disabling the filter block, see the Filter Block Bypass Logic diagram.

29.3.1.3 Sampled, Non-Filtered mode (#s 3B)

In Sampled, Non-Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising-edge is detected on the filter block clock input.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

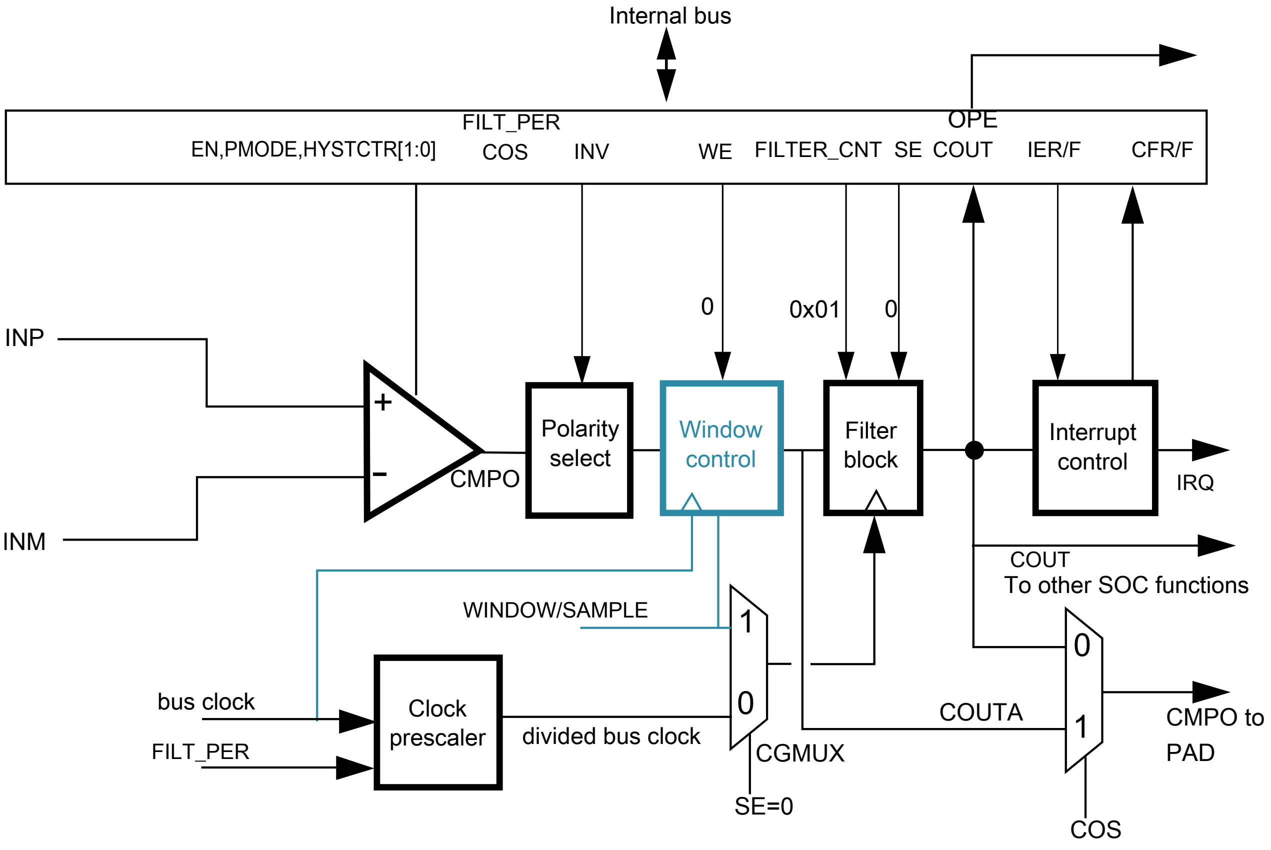


Figure 29-4. Sampled, Non-Filtered (# 3B): sampling interval internally derived

29.3.1.4 Sampled, Filtered mode (#s 4B)

In Sampled, Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

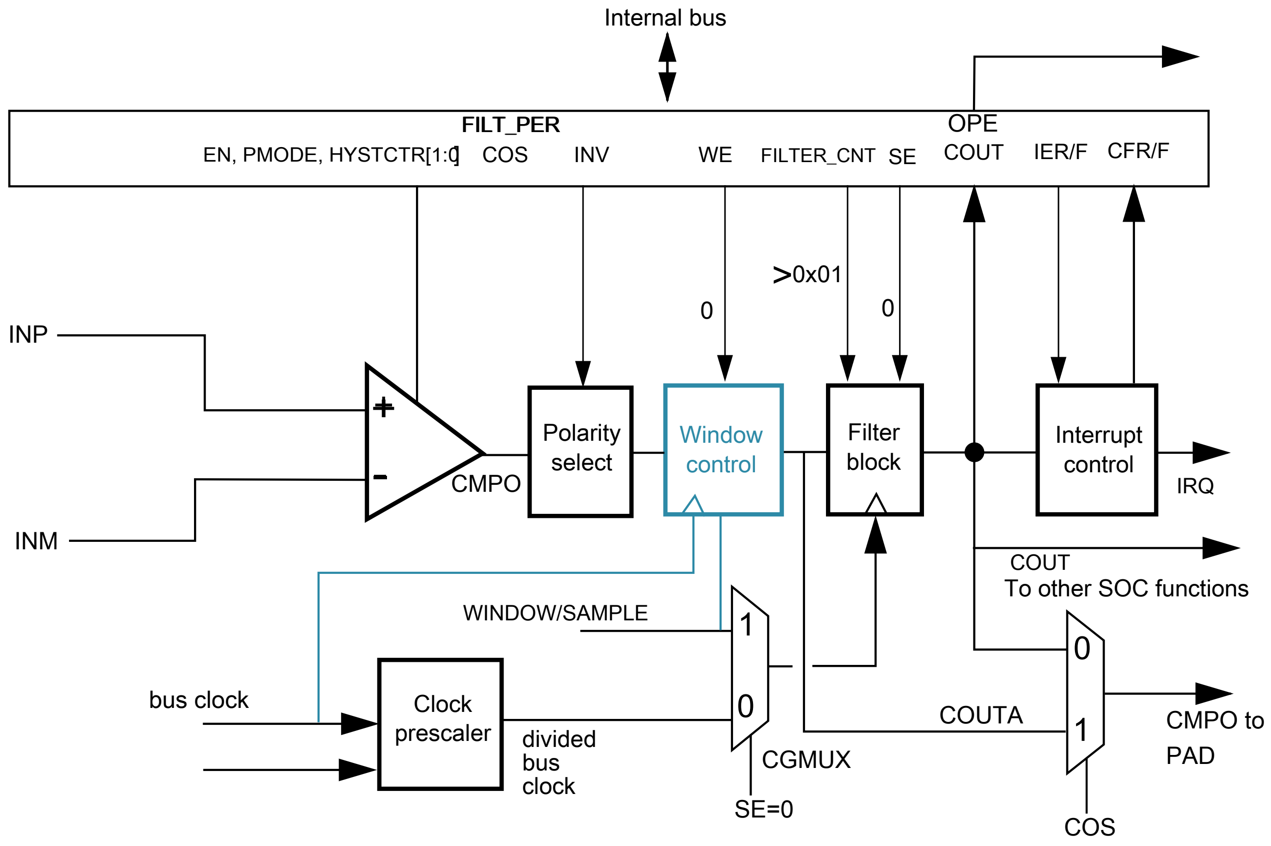


Figure 29-5. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, CR0[FILTER_CNT]>1, which activates filter operation.

29.3.2 Power modes

29.3.2.1 Wait mode operation

During Wait and VLPW modes, the CMP, if enabled, continues to operate normally and a CMP interrupt can wake the MCU.

29.3.2.2 Stop mode operation

Depending on clock restrictions related to the MCU core or core peripherals, the MCU is brought out of stop when a compare event occurs and the corresponding interrupt is enabled. Similarly, if CR1[OPE] is enabled, the comparator output operates as in the normal operating mode and comparator output is placed onto the external pin. In Stop modes, the comparator can be operational in both:

- High-Speed (HS) Comparison mode when CR1[PMODE] = 1
- Low-Speed (LS) Comparison mode when CR1[PMODE] = 0

It is recommended to use the LS mode to minimize power consumption.

If stop is exited with a reset, all comparator registers are put into their reset state.

29.3.2.3 Low-Leakage mode operation

When the chip is in Low-Leakage modes:

- The CMP module is partially functional and is limited to Low-Speed mode, regardless of CR1[PMODE] setting
- Windowed, Sampled, and Filtered modes are not supported
- The CMP output pin is latched and does not reflect the compare output state.

The positive- and negative-input voltage can be supplied from external pins or the DAC output. The MCU can be brought out of the Low-Leakage mode if a compare event occurs and the CMP interrupt is enabled. After wakeup from low-leakage modes, the CMP module is in the reset state except for SCR[CFF] and SCR[CFR].

29.3.2.4 Background Debug Mode Operation

When the microcontroller is in active background debug mode, the CMP continues to operate normally.

29.3.3 Startup and operation

A typical startup sequence is listed here.

- The time required to stabilize COUT will be the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog

comparator and filter. See the Data Sheets for power-on delays of the comparators. The filter delay is specified in the Low-pass filter.
- During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and SCR[CFR]/SCR[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.
- When programmed for filtering modes, COUT will initially be equal to 0, until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

29.3.4 Low-pass filter

The low-pass filter operates on the unfiltered and unsynchronized and optionally inverted comparator output COUTA and generates the filtered and synchronized output COUT.

Both COUTA and COUT can be configured as module outputs and are used for different purposes within the system.

Synchronization and edge detection are always used to determine status register bit values. They also apply to COUT for all sampling modes. Filtering can be performed using an internal timebase defined by FPR[FILT_PER] to determine sample time.

The need for digital filtering and the amount of filtering is dependent on user requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, high-frequency oscillations can be generated at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

29.3.4.1 Enabling filter modes

Filter modes can be enabled by:

- Setting CR0[FILTER_CNT] > 0x01 and
- Setting FPR[FILT_PER] to a nonzero value

Using the divided bus clock to drive the filter, it will take samples of COUTA every FPR[FILT_PER] bus clock cycles.

The filter output will be at logic 0 when first initalized, and will subsequently change when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed. In other words, SCR[COUT] will be 0 for some initial period, even when COUTA is at logic 1.

Setting FPR[FILT_PER] to 0 disables the filter and eliminates switching current associated with the filtering process.

Note

> Always switch to this setting prior to making any changes in filter parameters. This resets the filter to a known state. Switching CR0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

29.3.4.2 Latency issues

The value of FPR[FILT_PER] or SAMPLE period must be set such that the sampling period is just longer than the period of the expected noise. This way a noise spike will corrupt only one sample. The value of CR0[FILTER_CNT] must be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CR0[FILTER_CNT].

The values of FPR[FILT_PER] or SAMPLE period and CR0[FILTER_CNT] must also be traded off against the desire for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CR0[FILTER_CNT].

The following table summarizes maximum latency values for the various modes of operation *in the absence of noise*. Filtering latency is restarted each time an actual output transition is masked by noise.

Table 29-2. Comparator sample/filter maximum latencies

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
1	0	X	X	X	X	Disabled	N/A
2A	1	0	0	0x00	X	Continuous Mode	T_{PD}
2B	1	0	0	X	0x00		
3B	1	0	0	0x01	> 0x00	Sampled, Non-Filtered mode	T_{PD} + (FPR[FILT_PER] * T_{per}) + T_{per}
4B	1	0	0	> 0x01	> 0x00	Sampled, Filtered mode	T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + T_{per}

[1] T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{per} is the period of the bus clock.

29.4 CMP interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both.

The following table gives the conditions in which the interrupt request is asserted and deasserted.

When	Then
SCR[IER] and SCR[CFR] are set	The interrupt request is asserted
SCR[IEF] and SCR[CFF] are set	The interrupt request is asserted
SCR[IER] and SCR[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
SCR[IEF] and SCR[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

29.5 DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt instead. When the DMA has completed the transfer, it sends a transfer completing indicator that deasserts the DMA transfer request and clears the flag to allow a subsequent change on comparator output to occur and force another DMA request.

The comparator can remain functional in STOP modes.

When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, system will go back to STOP modes. Refer to DMA chapters in the device reference manual for the asynchronous DMA function for details.

29.6 CMP Asynchronous DMA support

The comparator can remain functional in STOP modes.

When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, system will go back to STOP modes. Refer to DMA chapters in the device reference manual for the asynchronous DMA function for details.

29.7 Digital-to-analog converter

The figure found here shows the block diagram of the DAC module.

It contains a 64-tap resistor ladder network and a 64-to-1 multiplexer, which selects an output voltage from one of 64 distinct levels that outputs from DACO. It is controlled through the DAC Control Register (DACCR). Its supply reference source can be selected from two sources V_{in1} and V_{in2}. The module can be powered down or disabled when not in use. When in Disabled mode, DACO is connected to the analog ground.

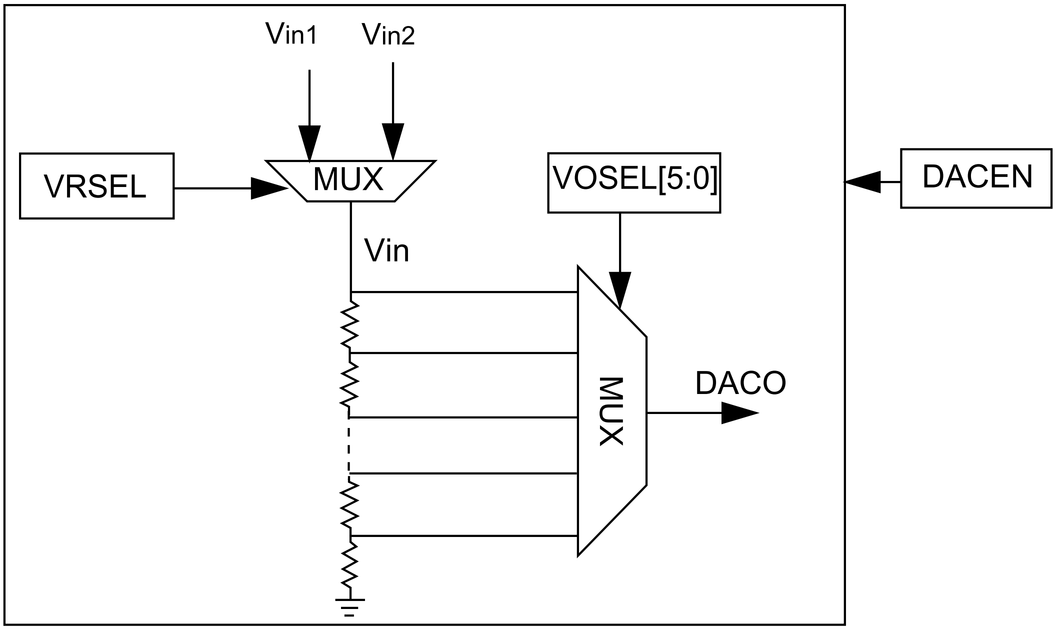


Figure 29-6. 6-bit DAC block diagram

29.8 DAC functional description

This section provides DAC functional description information.

29.8.1 Voltage reference source select

- V_{in1} connects to the primary voltage source as supply reference of 64 tap resistor ladder
- V_{in2} connects to an alternate voltage source

29.9 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

29.10 DAC clocks

This module has a single clock input, the bus clock.

29.11 DAC interrupts

This module has no interrupts.

29.12 CMP Trigger Mode

CMP and DAC are configured to CMP Trigger mode when CMP_CR1[TRIGM] is set to 1.

In addition, the CMP must be enabled. If the DAC is to be used as a reference to the CMP, it must also be enabled.

CMP Trigger mode depends on an external timer resource to periodically enable the CMP and 6-bit DAC in order to generate a triggered compare.

Upon setting TRIGM, the CMP and DAC are placed in a standby state until an external timer resource trigger is received.

Chapter 30
12-bit digital-to-analog converter (DAC)

The 12-bit digital-to-analog converter (DAC) is a low-power general purpose DAC whose output can be placed on an external pin, or set as one of the inputs to the analog comparator, OPAMPs, ADC, or other peripherals.

30.1 Introduction

The 12-bit digital-to-analog converter (DAC) is a low-power, general-purpose DAC. The output of the DAC can be placed on an external pin or set as one of the inputs to the analog comparator, op-amps, or ADC.

30.2 Features

The features of the DAC module include:

- On-chip programmable reference generator output. The voltage output range is from 1⁄4096 V_{in} to V_{in}, and the step is 1⁄4096 V_{in}, where V_{in} is the input voltage.
- V_{in} can be selected from two reference sources
- Static operation in Normal Stop mode
- 2-word data buffer supported with multiple operation modes
- DMA support

30.3 Block diagram

The block diagram of the DAC module is as follows:

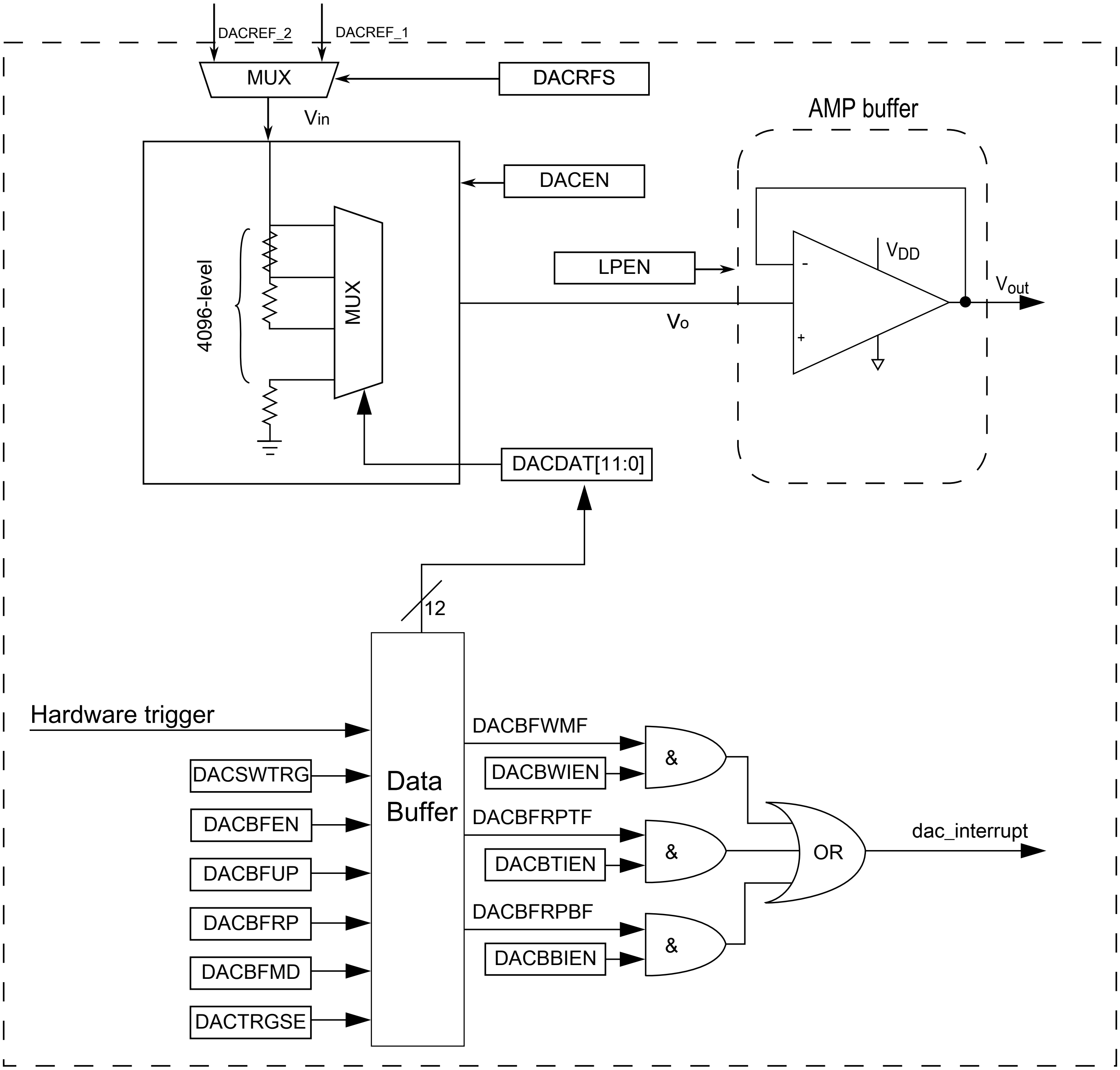


Figure 30-1. DAC block diagram

30.4 Memory map/register definition

The DAC has registers to control analog comparator and programmable voltage divider to perform the digital-to-analog functions.

DAC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_F000	DAC Data Low Register (DAC0_DAT0L)	8	R/W	00h	30.4.1/577
4003_F001	DAC Data High Register (DAC0_DAT0H)	8	R/W	00h	30.4.2/577
4003_F002	DAC Data Low Register (DAC0_DAT1L)	8	R/W	00h	30.4.1/577
4003_F003	DAC Data High Register (DAC0_DAT1H)	8	R/W	00h	30.4.2/577
4003_F020	DAC Status Register (DAC0_SR)	8	R/W	See section	30.4.3/578
4003_F021	DAC Control Register (DAC0_C0)	8	R/W	00h	30.4.4/579
4003_F022	DAC Control Register 1 (DAC0_C1)	8	R/W	00h	30.4.5/580
4003_F023	DAC Control Register 2 (DAC0_C2)	8	R/W	01h	30.4.6/581

30.4.1 DAC Data Low Register (DACx_DAT*n*L)

Address: 4003_F000h base + 0h offset + (2d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	DATA0							
Reset	0	0	0	0	0	0	0	0

DACx_DAT*n*L field descriptions

Field	Description
DATA0	DATA0 When the DAC buffer is not enabled, DATA[11:0] controls the output voltage based on the following formula: $V_{out} = V_{in} * (1 + DACDAT0[11:0])/4096$ When the DAC buffer is enabled, DATA is mapped to the 16-word buffer.

30.4.2 DAC Data High Register (DACx_DAT*n*H)

Address: 4003_F000h base + 1h offset + (2d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read	0				DATA1			
Write								
Reset	0	0	0	0	0	0	0	0

DACx_DAT*n*H field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DATA1	DATA1

Table continues on the next page...

DACx_DAT*n*H field descriptions (continued)

Field	Description
	When the DAC Buffer is not enabled, DATA[11:0] controls the output voltage based on the following formula. $V_{out} = V_{in}$ * (1 + DACDAT0[11:0])/4096 When the DAC buffer is enabled, DATA[11:0] is mapped to the 16-word buffer.

30.4.3 DAC Status Register (DACx_SR)

If DMA is enabled, the flags can be cleared automatically by DMA when the DMA request is done. Writing 0 to a field clears it whereas writing 1 has no effect. After reset, DACBFRPTF is set and can be cleared by software, if needed. The flags are set only when the data buffer status is changed.

NOTE

Do not use 32/16-bit accesses to this register.

Address: 4003_F000h base + 20h offset = 4003_F020h

Bit	7	6	5	4	3	2	1	0
Read	0					0	DACBFRPTF	DACBFRPBF
Write								
Reset	0	0	0	0	0	0	1	0

DACx_SR field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DACBFRPTF	DAC Buffer Read Pointer Top Position Flag 0 The DAC buffer read pointer is not zero. 1 The DAC buffer read pointer is zero.
0 DACBFRPBF	DAC Buffer Read Pointer Bottom Position Flag 0 The DAC buffer read pointer is not equal to C2[DACBFUP]. 1 The DAC buffer read pointer is equal to C2[DACBFUP].

30.4.4 DAC Control Register (DACx_C0)

NOTE

Do not use 32- or 16-bit accesses to this register.

Address: 4003_F000h base + 21h offset = 4003_F021h

Bit	7	6	5	4	3	2	1	0
Read	DACEN	DACRFS	DACTRGSEL	0	LPEN	0	DACBTIEN	DACBBIEN
Write				DACSWTRG				
Reset	0	0	0	0	0	0	0	0

DACx_C0 field descriptions

Field	Description
7 DACEN	DAC Enable Starts the Programmable Reference Generator operation. 0 The DAC system is disabled. 1 The DAC system is enabled.
6 DACRFS	DAC Reference Select 0 The DAC selects DACREF_1 as the reference voltage. 1 The DAC selects DACREF_2 as the reference voltage.
5 DACTRGSEL	DAC Trigger Select 0 The DAC hardware trigger is selected. 1 The DAC software trigger is selected.
4 DACSWTRG	DAC Software Trigger Active high. This is a write-only field, which always reads 0. If DAC software trigger is selected and buffer is enabled, writing 1 to this field will advance the buffer read pointer once. 0 The DAC soft trigger is not valid. 1 The DAC soft trigger is valid.
3 LPEN	DAC Low Power Control **NOTE:** See the 12-bit DAC electrical characteristics of the device data sheet for details on the impact of the modes below. 0 High-Power mode 1 Low-Power mode
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DACBTIEN	DAC Buffer Read Pointer Top Flag Interrupt Enable

Table continues on the next page...

DACx_C0 field descriptions (continued)

Field	Description
	0 The DAC buffer read pointer top flag interrupt is disabled. 1 The DAC buffer read pointer top flag interrupt is enabled.
0 DACBBIEN	DAC Buffer Read Pointer Bottom Flag Interrupt Enable 0 The DAC buffer read pointer bottom flag interrupt is disabled. 1 The DAC buffer read pointer bottom flag interrupt is enabled.

30.4.5 DAC Control Register 1 (DACx_C1)

NOTE

Do not use 32- or 16-bit accesses to this register.

Address: 4003_F000h base + 22h offset = 4003_F022h

Bit	7	6	5	4	3	2	1	0
Read	DMAEN	0				DACBFMD	0	DACBFEN
Write								
Reset	0	0	0	0	0	0	0	0

DACx_C1 field descriptions

Field	Description
7 DMAEN	DMA Enable Select 0 DMA is disabled. 1 DMA is enabled. When DMA is enabled, the DMA request will be generated by original interrupts. The interrupts will not be presented on this module at the same time.
6–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 DACBFMD	DAC Buffer Work Mode Select 0 Normal mode 1 One-Time Scan mode
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DACBFEN	DAC Buffer Enable 0 Buffer read pointer is disabled. The converted data is always the first word of the buffer. 1 Buffer read pointer is enabled. The converted data is the word that the read pointer points to. It means converted data can be from any word of the buffer.

30.4.6 DAC Control Register 2 (DACx_C2)

Address: 4003_F000h base + 23h offset = 4003_F023h

Bit	7	6	5	4	3	2	1	0
Read		0		DACBFRP		0		DACBFUP
Write								
Reset	0	0	0	0	0	0	0	1

DACx_C2 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DACBFRP	DAC Buffer Read Pointer Keeps the current value of the buffer read pointer.
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DACBFUP	DAC Buffer Upper Limit Selects the upper limit of the DAC buffer. The buffer read pointer cannot exceed it.

30.5 Functional description

The 12-bit DAC module can select one of the two reference inputs—DACREF_1 and DACREF_2 as the DAC reference voltage, V_{in} by C0 [DACRFS]. See the chip-specific DAC information to determine the source options for DACREF_1 and DACREF_2.

When the DAC is enabled, it converts the data in DACDAT0[11:0] or the data from the DAC data buffer to a stepped analog output voltage. The output voltage range is from V_{in} to $V_{in}/4096$, and the step is $V_{in}/4096$.

30.5.1 DAC data buffer operation

When the DAC is enabled and the buffer is not enabled, the DAC module always converts the data in DAT0 to the analog output voltage.

When both the DAC and the buffer are enabled, the DAC converts the data in the data buffer to analog output voltage. The data buffer read pointer advances to the next word whenever a hardware or software trigger event occurs.

The data buffer can be configured to operate in Normal mode, One-Time Scan mode. When the buffer operation is switched from one mode to another, the read pointer does not change. The read pointer can be set to any value between 0 and C2[DACBFUP] by writing C2[DACBFRP].

30.5.1.1 DAC data buffer interrupts

There are several interrupts and associated flags that can be configured for the DAC buffer. SR[DACBFRPBF] is set when the DAC buffer read pointer reaches the DAC buffer upper limit, that is, C2[DACBFRP] = C2[DACBFUP]. SR[DACBFRPTF] is set when the DAC read pointer is equal to the start position, 0.

30.5.1.2 Modes of DAC data buffer operation

The following table describes the different modes of data buffer operation for the DAC module.

Table 30-1. Modes of DAC data buffer operation

Modes	Description
Buffer Normal mode	This is the default mode. The buffer works as a circular buffer. The read pointer increases by one, every time the trigger occurs. When the read pointer reaches the upper limit, it goes to 0 directly in the next trigger event.
Buffer One-time Scan mode	The read pointer increases by 1 every time the trigger occurs. When it reaches the upper limit, it stops there. If read pointer is reset to the address other than the upper limit, it will increase to the upper address and stop there again. **NOTE:** If the software set the read pointer to the upper limit, the read pointer will not advance in this mode.

30.5.2 DMA operation

When DMA is enabled, DMA requests are generated instead of interrupt requests. The DMA Done signal clears the DMA request.

The status register flags are still set and are cleared automatically when the DMA completes.

30.5.3 Resets

During reset, the DAC is configured in the default mode and is disabled.

30.5.4 Low-Power mode operation

The following table shows the wait mode and the stop mode operation of the DAC module.

Table 30-2. Modes of operation

Modes of operation	Description
Wait mode	The DAC will operate normally, if enabled.
Stop mode	If enabled, the DAC module continues to operate in Normal Stop mode and the output voltage will hold the value before stop. In low-power stop modes, the DAC is fully shut down.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

Chapter 31
Front End Module Interface

31.1 External front end module (FEM)

31.1.1 Introduction

This chapter defines the KW40Z SoC RF Front End Interface requirements.

31.1.2 Block Diagram

The following block diagram shows the connections between KW40Z SoC and the RF front end matching circuitry. The RF port configuration of KW40Z is differential. The input impedance of the on-chip LNA is approximately 100 ohm differential. A 2:1 RF balun is used to transform the RF_P/RF_N ports to 50 ohm. A harmonic trap filter is included in the matching circuitry.

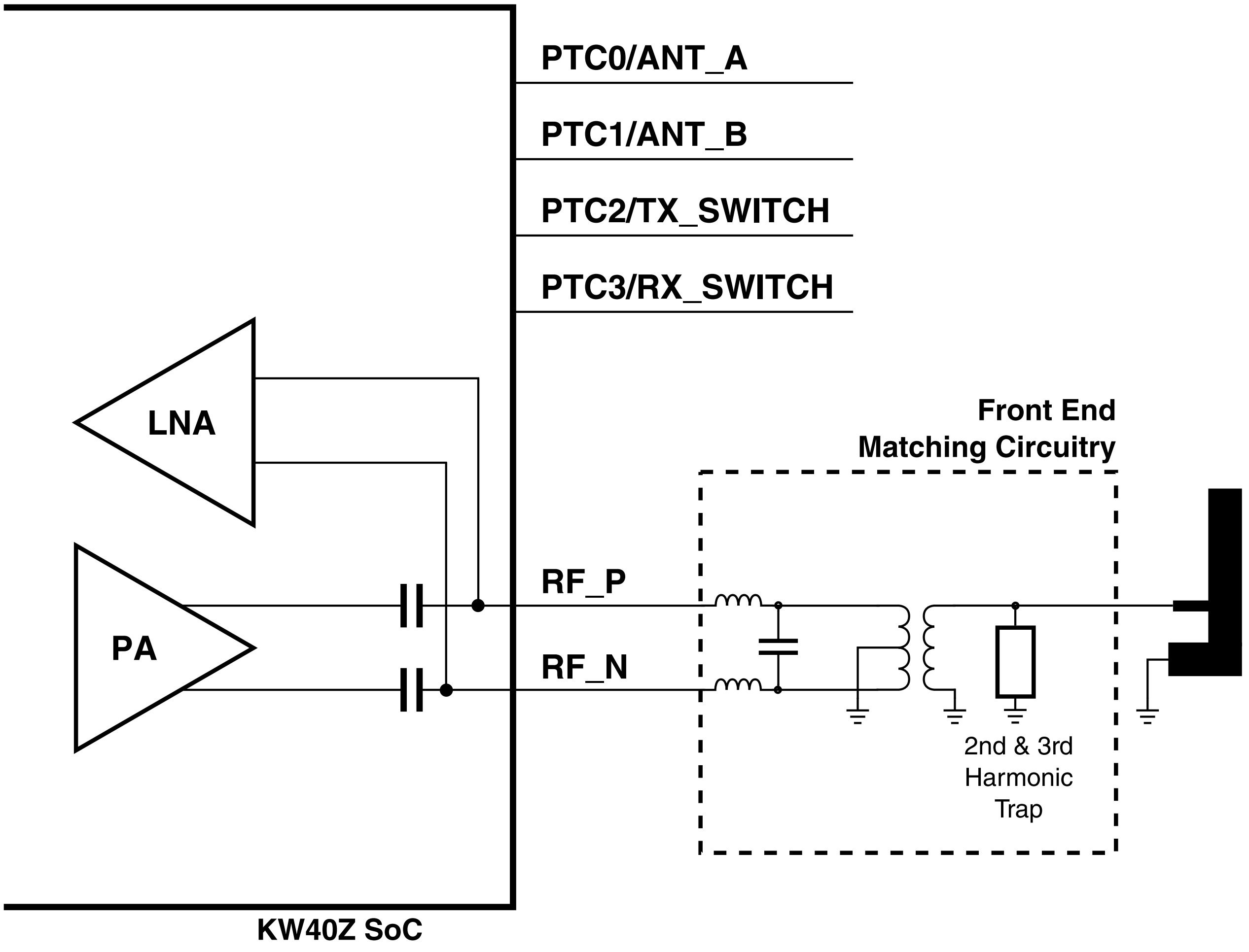


Figure 31-1. KW40Z SoC RF front end matching circuitry

31.1.3 KW40Z SoC RF Front End Characteristics

Table 31-1. KW40Z SoC RF Front End Characteristics

Characteristic	Description	Min	Typ	Max	Unit
Output Frequency	Transmitter output frequency range	2360		2483.5	MHz
Input Frequency	Receiver input frequency range	2360		2480	MHz
Temperature Range	Operating temperature range	-40	25	85	°C
Maximum Tx RF Output Power	Transmitter maximum output power		4.5		dBm
Minimum Tx RF Output Power	Transmitter minimum output power		-16		dBm
TX RF Output Power Step	Power Step 0		-16		dBm

Table continues on the next page...

Table 31-1. KW40Z SoC RF Front End Characteristics (continued)

Characteristic	Description	Min	Typ	Max	Unit
	Power Step 1		-11		dBm
	Power Step 2		-6		dBm
	Power Step 3		-3.3		dBm
	Power Step 4		-1		dBm
	Power Step 5		0		dBm
	Power Step 6		1.3		dBm
	Power Step 7		2.0		dBm
	Power Step 8		2.5		dBm
	Power Step 9		3.0		dBm
	Power Step 10		3.5		dBm
	Power Step 11		3.8		dBm
	Power Step 12		4.1		dBm
	Power Step 13		4.3		dBm
	Power Step 14		4.4		dBm
	Power Step 15		4.5		dBm
Tx 2nd Harmonic Level				-36	dBm
Tx 3rd Harmonic Level				-24	dBm
Max RF Input Power	Receiver maximum input power			-10	dBm
Logic Output Voltage Low	GPIO VOL			0.5	V
Logic Output Voltage High	GPIO VOH	VDD_INT - 0.5	-	-	V
Logic Output Drive Current (Normal)	1.71V <= VDD <= 2.7V			2.5	mA
	2.7V <= VDD <= 3.6V			5	mA
Logic Output Drive Current (High)	1.71V <= VDD <= 2.7V			10	mA
	2.7V <= VDD <= 3.6V			20	mA
Supply Voltage (VDD_1/ VDD_2)	Digital Supply Voltage	1.71	-	3.6	V
# of GPIO for FEM Control	Number of control signals for FEM	-	4	-	-

31.1.4 RF Control Signals

This chapter describes the KW40Z SoC RF Control Signals.

31.1.4.1 Introduction

The KW40Z SoC provides four dedicated signals for the control of external RF components. These signals are designated as PTC0-3(ANT_A, ANT_B, TX_SWITCH, RX_SWITCH) and can be enabled to control external amplifiers, antenna switches, and other modules. Typical uses include:

- Antenna Diversity
- External PA
- External LNA
- T/R switching

These four signals are controlled by the Transceiver Sequence Manager (TSM) in KW40Z. They can also be controlled manually by software as a GPIO.

31.1.4.2 Control Signal Drive Strength Programmability

These four signals can be programmed to select sink and source current from 2.5mA to 20mA to control FEM (Front End Module) features such as PA, LNA and RF switches. The polarity of these signals is programmable. See section KW40Z SoC RF Front End Characteristics for the voltage and current specification of these signals.

31.1.4.3 Single Antenna Configuration

In single antenna configuration, both TX_SWITCH and RX_SWITCH will be used to control the FEM. ANT_A and ANT_B will not be used because there is no need for antenna selection. To program KW40Z to single antenna configuration, the antenna controls enabled ANTX_EN[1:0] should be programmed with <01>.

Table 31-2. Single Antenna Configuration (ANTX_EN[1:0] = <01>)

Mode	TX_SWITCH	RX_SWITCH
Power Down	0	0
TX	1	0
RX - LNA	0	1

31.1.4.4 Dual Antenna Configuration: Single and Dual Modes

To be able to control various dual antenna FEM configurations, these four control signals can be set to two modes: Single or Dual modes. In single mode, only one antenna control signal (ANT_A) will be used. In dual mode, both antenna control signals (ANT_A and ANT_B) will be used to control the FEM. Also, in single mode, RX_SWITCH will stay high in both TX and RX modes. Therefore only TX_SWITCH is used to select the TX and RX modes in the FEM. The ANTX_ctrlmode register bit is used to select between these two modes. To program KW40Z to dual antenna configuration, the antenna controls enabled ANTX_EN[1:0] should be programmed with <11>.

31.1.4.4.1 Control Signal Logic

The following two tables show the state of the control signals in different operation modes. In single mode, only ANT_A is used for antenna selection. In dual mode, both ANT_A and ANT_B are used.

Table 31-3. Dual Antenna (ANTX_EN = <11>), Single Mode (ANTX_ctrlmode = 0)

Mode	TX_SWITCH	RX_SWITCH	ANT_A
Power Down	0	0	0
Antenna A - RX	0	1	1
Antenna B - RX	0	1	0
Antenna A - TX	1	1	1
Antenna B - TX	1	1	0

Table 31-4. Dual Antenna (ANTX_EN = <11>), Dual Mode (ANTX_ctrlmode = 1)

Mode	TX_SWITCH	RX_SWITCH	ANT_A	ANT_B
Power Down	0	0	0	0
Antenna A - RX	0	1	1	0
Antenna B - RX	0	1	0	1
Antenna A - TX	1	0	1	0
Antenna B - TX	1	0	0	1

31.1.4.4.2 Timing Diagrams

The following two diagrams show the timing diagrams of single and dual mode in dual antenna configuration.

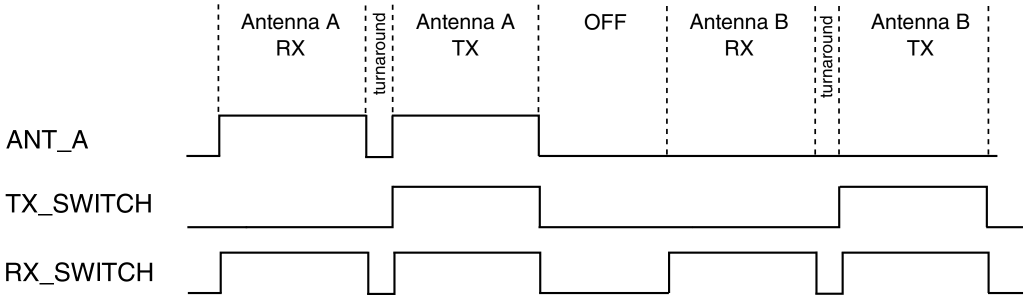


Figure 31-2. Dual Antenna: Single Mode (ANTX_EN[1:0] = <11>, ANTX_ctrlmode = 0)

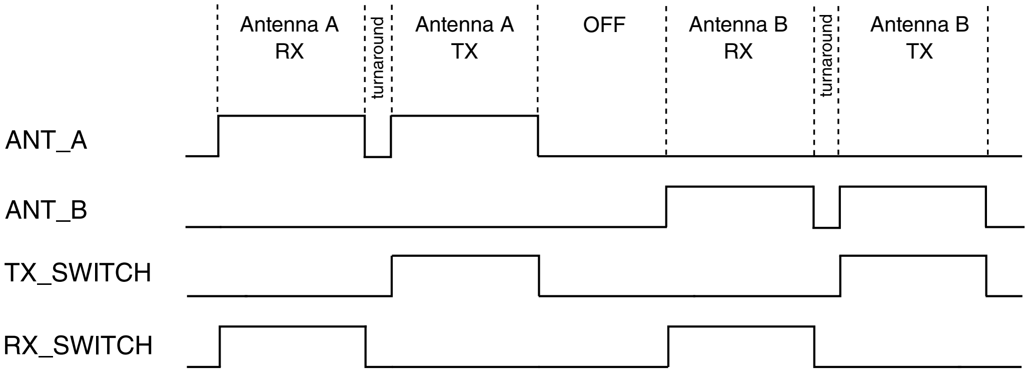


Figure 31-3. Dual Antenna: Dual Mode (ANTX_EN[1:0] = <11>, ANTX_ctrlmode = 1)

31.1.5 Typical Configuration of KW40 with FEM

The following figure shows a typical configuration of KW40Z with a 2.4GHz FEM in single antenna configuration. PTC2 and PTC3 are used to interfacing with the FEM.

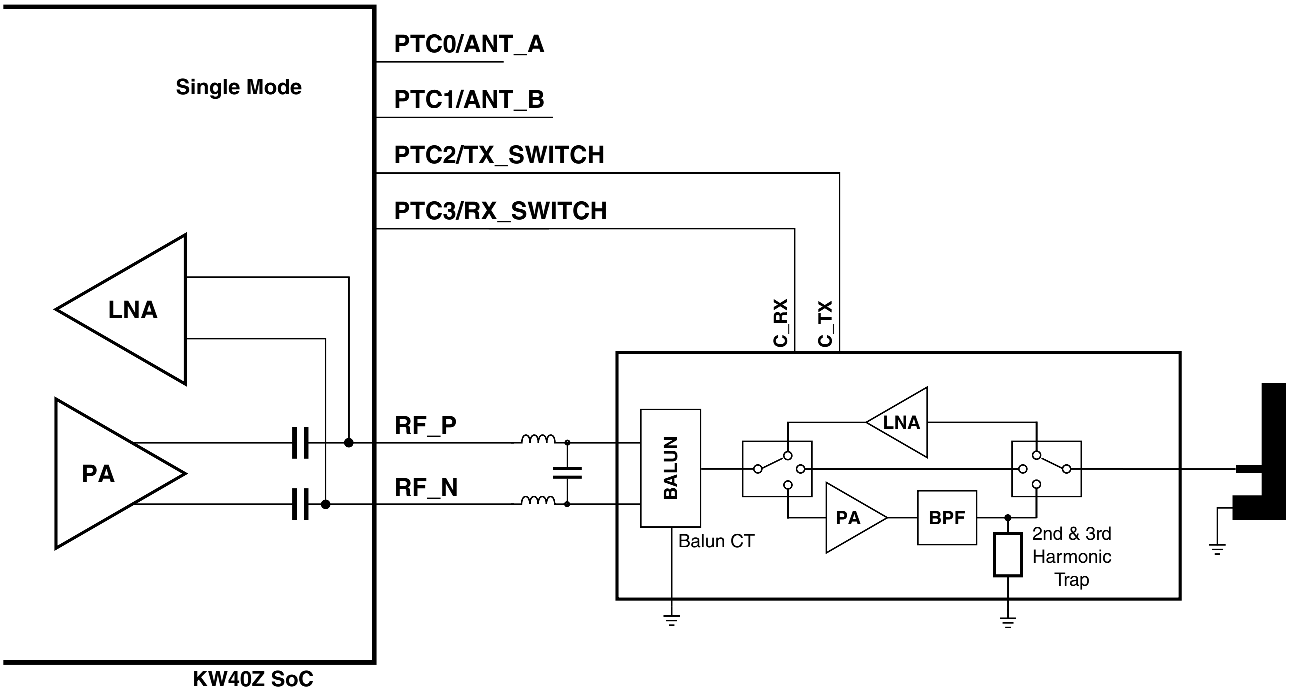


Figure 31-4. Single Antenna Configuration

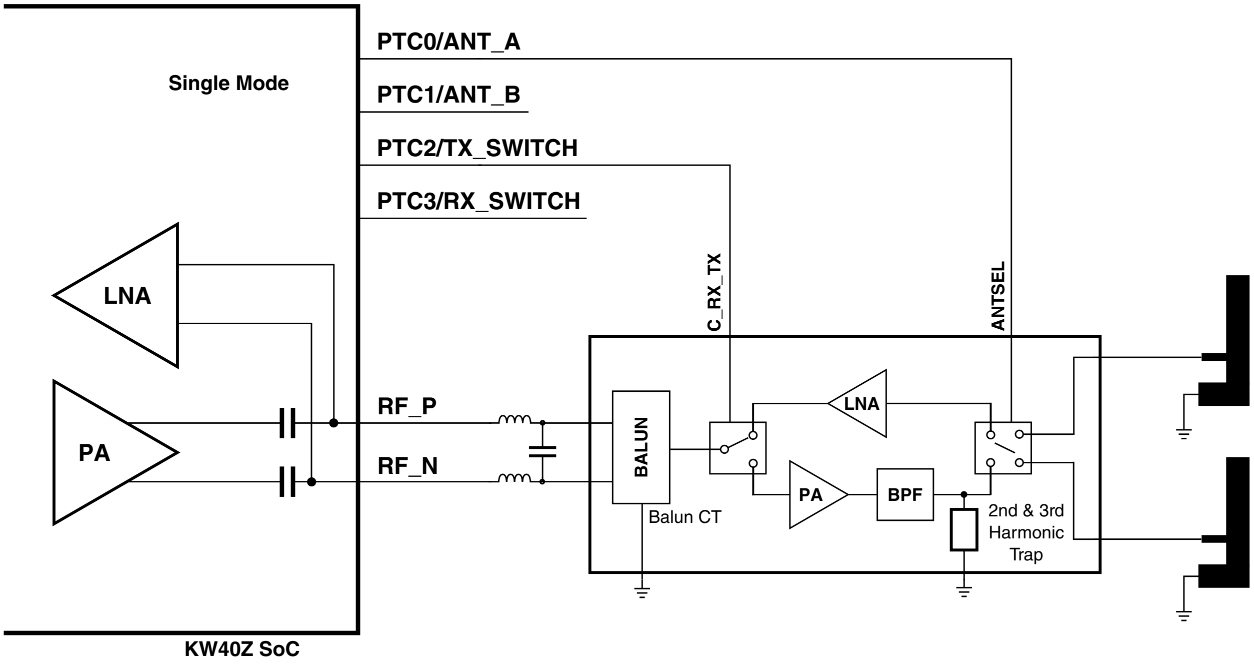


Figure 31-5. Dual Antenna Configuration: Single Mode

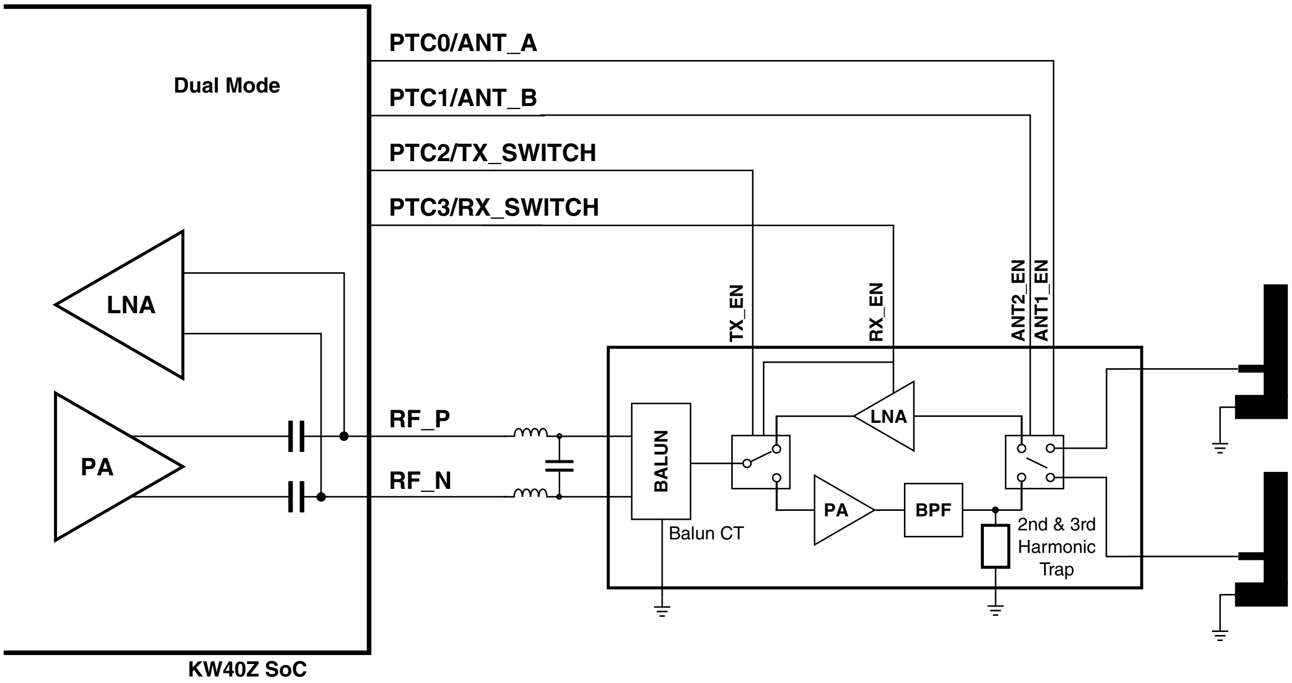


Figure 31-6. Dual Antenna Configuration: Dual Mode

Chapter 32
Timer/PWM Module (TPM)

32.1 Introduction

The TPM (Timer/PWM Module) is a 2- to 8-channel timer which supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications.

The counter, compare and capture registers are clocked by an asynchronous clock that can remain enabled in low power modes. An example of using the TPM with the asynchronous DMA is described in AN4631:Using the Asynchronous DMA features of the Kinetis L Series .

32.1.1 TPM Philosophy

The TPM is built upon a very simple timer (HCS08 Timer PWM Module – TPM) used for many years on Freescale's 8-bit microcontrollers. The TPM extends the functionality to support operation in low power modes by clocking the counter, compare and capture registers from an asynchronous clock that can remain functional in low power modes.

32.1.2 Features

The TPM features include:

- TPM clock mode is selectable
    - Can increment on every edge of the asynchronous counter clock
    - Can increment on rising edge of an external clock input synchronized to the asynchronous counter clock
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128

- TPM includes a 16-bit counter
  - It can be a free-running counter or modulo counter
  - The counting can be up or up-down
- Includes 4 channels that can be configured for input capture, output compare, edge-aligned PWM mode, or center-aligned PWM mode
  - In input capture mode the capture can occur on rising edges, falling edges or both edges
  - In output compare mode the output signal can be set, cleared, pulsed, or toggled on match
  - All channels can be configured for edge-aligned PWM mode or center-aligned PWM mode
- Support the generation of an interrupt and/or DMA request per channel
- Support the generation of an interrupt and/or DMA request when the counter overflows
- Support selectable trigger input to optionally reset or cause the counter to start incrementing.
  - The counter can also optionally stop incrementing on counter overflow
- Support the generation of hardware triggers when the counter overflows and per channel

32.1.3 Modes of operation

During debug mode, the TPM can can be configured to temporarily pause all counting until the core returns to normal user operating mode or to operate normally. When the counter is paused, trigger inputs and input capture events are ignored.

During doze mode, the TPM can be configured to operate normally or to pause all counting for the duration of doze mode. When the counter is paused, trigger inputs and input capture events are ignored.

During stop mode, the TPM counter clock can remain functional and the TPM can generate an asynchronous interrupt to exit the MCU from stop mode.

32.1.4 Block diagram

The TPM uses one input/output (I/O) pin per channel, CHn (TPM channel (n)) where n is the channel number.

The following figure shows the TPM structure. The central component of the TPM is the 16-bit counter with programmable final value and its counting can be up or up-down.

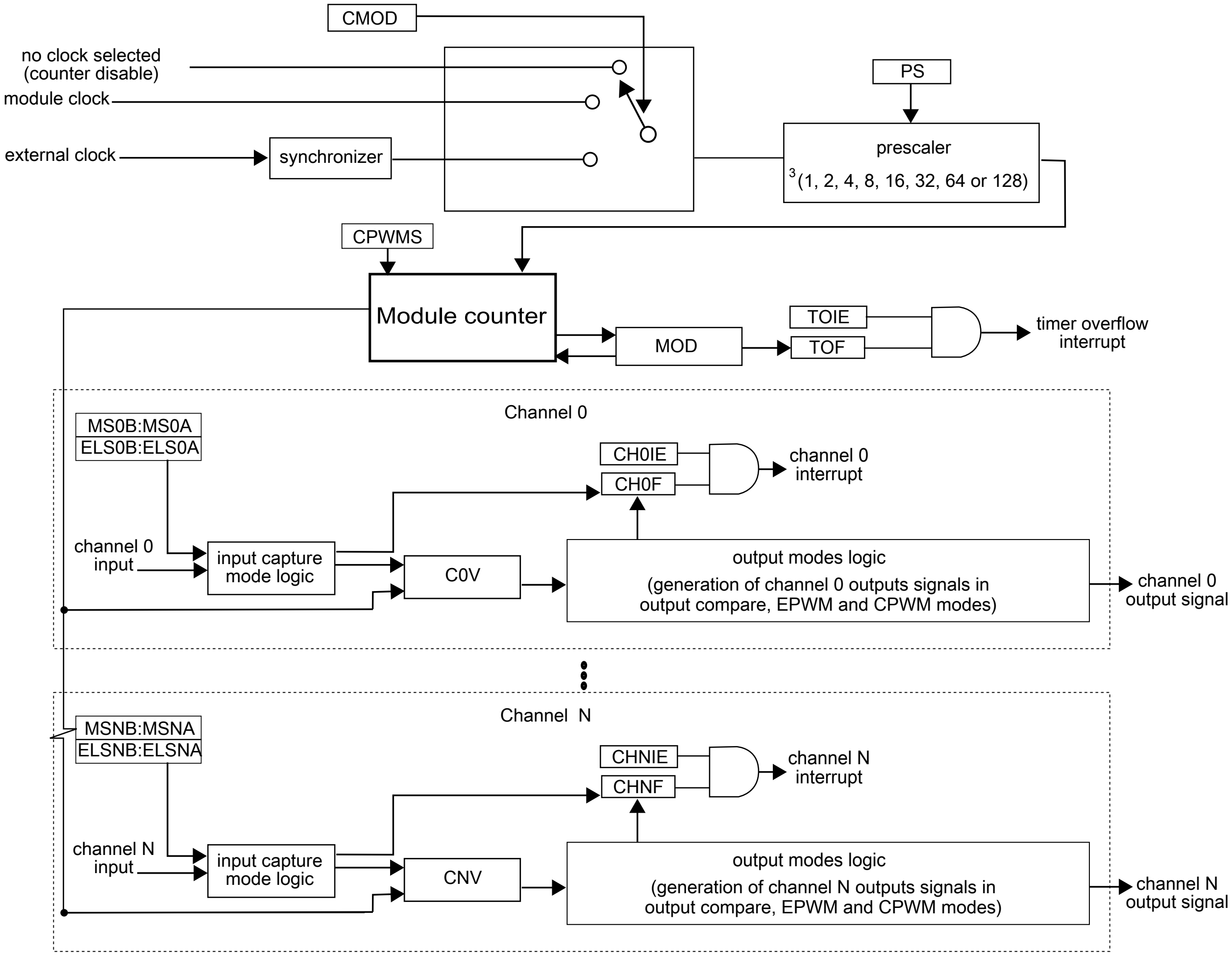


Figure 32-1. TPM block diagram

32.2 TPM Signal Descriptions

Table 32-1 shows the user-accessible signals for the TPM.

Table 32-1. TPM signal descriptions

Signal	Description	I/O
TPM_EXTCLK	External clock. TPM external clock can be selected to increment the TPM counter on every rising edge synchronized to the counter clock.	I
TPM_CHn	TPM channel (n = 3 to 0). A TPM channel pin is configured as output when configured in an output compare or PWM mode and the TPM counter is enabled, otherwise the TPM channel pin is an input.	I/O

32.2.1 TPM_EXTCLK — TPM External Clock

The rising edge of the external input signal is used to increment the TPM counter if selected by CMOD[1:0] bits in the SC register. This input signal must be less than half of the TPM counter clock frequency. The TPM counter prescaler selection and settings are also used when an external input is selected.

32.2.2 TPM_CHn — TPM Channel (n) I/O Pin

Each TPM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.

32.3 Memory Map and Register Definition

This section provides a detailed description of all TPM registers.

Attempting to access a reserved register location in the TPM memory map will generate a bus error.

TPM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8000	Status and Control (TPM0_SC)	32	R/W	0000_0000h	32.3.1/598
4003_8004	Counter (TPM0_CNT)	32	R/W	0000_0000h	32.3.2/600
4003_8008	Modulo (TPM0_MOD)	32	R/W	0000_FFFFh	32.3.3/600
4003_800C	Channel (n) Status and Control (TPM0_C0SC)	32	R/W	0000_0000h	32.3.4/601
4003_8010	Channel (n) Value (TPM0_C0V)	32	R/W	0000_0000h	32.3.5/603
4003_8014	Channel (n) Status and Control (TPM0_C1SC)	32	R/W	0000_0000h	32.3.4/601

Table continues on the next page...

TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8018	Channel (n) Value (TPM0_C1V)	32	R/W	0000_0000h	32.3.5/603
4003_801C	Channel (n) Status and Control (TPM0_C2SC)	32	R/W	0000_0000h	32.3.4/601
4003_8020	Channel (n) Value (TPM0_C2V)	32	R/W	0000_0000h	32.3.5/603
4003_8024	Channel (n) Status and Control (TPM0_C3SC)	32	R/W	0000_0000h	32.3.4/601
4003_8028	Channel (n) Value (TPM0_C3V)	32	R/W	0000_0000h	32.3.5/603
4003_8050	Capture and Compare Status (TPM0_STATUS)	32	R/W	0000_0000h	32.3.6/604
4003_8064	Combine Channel Register (TPM0_COMBINE)	32	R/W	0000_0000h	32.3.7/606
4003_8070	Channel Polarity (TPM0_POL)	32	R/W	0000_0000h	32.3.8/607
4003_8078	Filter Control (TPM0_FILTER)	32	R/W	0000_0000h	32.3.9/608
4003_8080	Quadrature Decoder Control and Status (TPM0_QDCTRL)	32	R/W	0000_0000h	32.3.10/ 609
4003_8084	Configuration (TPM0_CONF)	32	R/W	0000_0000h	32.3.11/ 610
4003_9000	Status and Control (TPM1_SC)	32	R/W	0000_0000h	32.3.1/598
4003_9004	Counter (TPM1_CNT)	32	R/W	0000_0000h	32.3.2/600
4003_9008	Modulo (TPM1_MOD)	32	R/W	0000_FFFFh	32.3.3/600
4003_900C	Channel (n) Status and Control (TPM1_C0SC)	32	R/W	0000_0000h	32.3.4/601
4003_9010	Channel (n) Value (TPM1_C0V)	32	R/W	0000_0000h	32.3.5/603
4003_9014	Channel (n) Status and Control (TPM1_C1SC)	32	R/W	0000_0000h	32.3.4/601
4003_9018	Channel (n) Value (TPM1_C1V)	32	R/W	0000_0000h	32.3.5/603
4003_901C	Channel (n) Status and Control (TPM1_C2SC)	32	R/W	0000_0000h	32.3.4/601
4003_9020	Channel (n) Value (TPM1_C2V)	32	R/W	0000_0000h	32.3.5/603
4003_9024	Channel (n) Status and Control (TPM1_C3SC)	32	R/W	0000_0000h	32.3.4/601
4003_9028	Channel (n) Value (TPM1_C3V)	32	R/W	0000_0000h	32.3.5/603
4003_9050	Capture and Compare Status (TPM1_STATUS)	32	R/W	0000_0000h	32.3.6/604
4003_9064	Combine Channel Register (TPM1_COMBINE)	32	R/W	0000_0000h	32.3.7/606
4003_9070	Channel Polarity (TPM1_POL)	32	R/W	0000_0000h	32.3.8/607
4003_9078	Filter Control (TPM1_FILTER)	32	R/W	0000_0000h	32.3.9/608
4003_9080	Quadrature Decoder Control and Status (TPM1_QDCTRL)	32	R/W	0000_0000h	32.3.10/ 609
4003_9084	Configuration (TPM1_CONF)	32	R/W	0000_0000h	32.3.11/ 610
4003_A000	Status and Control (TPM2_SC)	32	R/W	0000_0000h	32.3.1/598
4003_A004	Counter (TPM2_CNT)	32	R/W	0000_0000h	32.3.2/600
4003_A008	Modulo (TPM2_MOD)	32	R/W	0000_FFFFh	32.3.3/600
4003_A00C	Channel (n) Status and Control (TPM2_C0SC)	32	R/W	0000_0000h	32.3.4/601
4003_A010	Channel (n) Value (TPM2_C0V)	32	R/W	0000_0000h	32.3.5/603
4003_A014	Channel (n) Status and Control (TPM2_C1SC)	32	R/W	0000_0000h	32.3.4/601
4003_A018	Channel (n) Value (TPM2_C1V)	32	R/W	0000_0000h	32.3.5/603

Table continues on the next page...

TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_A01C	Channel (n) Status and Control (TPM2_C2SC)	32	R/W	0000_0000h	32.3.4/601
4003_A020	Channel (n) Value (TPM2_C2V)	32	R/W	0000_0000h	32.3.5/603
4003_A024	Channel (n) Status and Control (TPM2_C3SC)	32	R/W	0000_0000h	32.3.4/601
4003_A028	Channel (n) Value (TPM2_C3V)	32	R/W	0000_0000h	32.3.5/603
4003_A050	Capture and Compare Status (TPM2_STATUS)	32	R/W	0000_0000h	32.3.6/604
4003_A064	Combine Channel Register (TPM2_COMBINE)	32	R/W	0000_0000h	32.3.7/606
4003_A070	Channel Polarity (TPM2_POL)	32	R/W	0000_0000h	32.3.8/607
4003_A078	Filter Control (TPM2_FILTER)	32	R/W	0000_0000h	32.3.9/608
4003_A080	Quadrature Decoder Control and Status (TPM2_QDCTRL)	32	R/W	0000_0000h	32.3.10/ 609
4003_A084	Configuration (TPM2_CONF)	32	R/W	0000_0000h	32.3.11/ 610

32.3.1 Status and Control (TPMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, module configuration and prescaler factor. These controls relate to all channels within this module.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DMA	TOF	TOIE	CPWMS	CMOD		PS		
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_SC field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DMA	DMA Enable Enables DMA transfers for the overflow flag. 0 Disables DMA transfers. 1 Enables DMA transfers.
7 TOF	Timer Overflow Flag Set by hardware when the TPM counter equals the value in the MOD register and increments. Writing a 1 to TOF clears it. Writing a 0 to TOF has no effect. If another TPM overflow occurs between the flag setting and the flag clearing, the write operation has no effect; therefore, TOF remains set indicating another overflow has occurred. In this case a TOF interrupt request is not lost due to a delay in clearing the previous TOF. 0 TPM counter has not overflowed. 1 TPM counter has overflowed.
6 TOIE	Timer Overflow Interrupt Enable Enables TPM overflow interrupts. 0 Disable TOF interrupts. Use software polling or DMA request. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the TPM to operate in up-down counting mode. This field is write protected. It can be written only when the counter is disabled. 0 TPM counter operates in up counting mode. 1 TPM counter operates in up-down counting mode.
4–3 CMOD	Clock Mode Selection Selects the TPM counter clock modes. When disabling the counter, this field remain set until acknolwedged in the TPM clock domain. 00 TPM counter is disabled 01 TPM counter increments on every TPM counter clock 10 TPM counter increments on rising edge of TPM_EXTCLK synchronized to the TPM counter clock 11 Reserved.
PS	Prescale Factor Selection Selects one of 8 division factors for the clock mode selected by CMOD. This field is write protected. It can be written only when the counter is disabled. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32

Table continues on the next page...

TPMx_SC field descriptions (continued)

Field	Description
	110 Divide by 64 111 Divide by 128

32.3.2 Counter (TPMx_CNT)

The CNT register contains the TPM counter value.

Reset clears the CNT register. Writing any value to COUNT also clears the counter.

When debug is active, the TPM counter does not increment unless configured otherwise.

Reading the CNT register adds two wait states to the register access due to synchronization delays.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter value

32.3.3 Modulo (TPMx_MOD)

The Modulo register contains the modulo value for the TPM counter. When the TPM counter reaches the modulo value and increments, the overflow flag (TOF) is set and the next value of TPM counter depends on the selected counting method (see Counter).

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to MOD Register Update . Additional writes to the MOD write buffer are ignored until the register has been updated.

It is recommended to initialize the TPM counter (write to CNT) before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TPMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MOD	Modulo value This field must be written with single 16-bit or 32-bit access.

32.3.4 Channel (n) Status and Control (TPMx_C*n*SC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function. When switching from one channel mode to a different channel mode, the channel must first be disabled and this must be acknowledged in the TPM counter clock domain.

Table 32-2. Mode, Edge, and Level Selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	00	00	None	Channel disabled
X	01	00	Software compare	Pin not used for TPM
0	00	01	Input capture	Capture on Rising Edge Only
		10		Capture on Falling Edge Only
		11		Capture on Rising or Falling Edge
	01	01	Output compare	Toggle Output on match
		10		Clear Output on match
		11		Set Output on match
	10	10	Edge-aligned PWM	High-true pulses (clear Output on match, set Output on reload)
		X1		Low-true pulses (set Output on match, clear Output on reload)
	11	10	Output compare	Pulse Output low on match
		01		Pulse Output high on match

Table continues on the next page...

Table 32-2. Mode, Edge, and Level Selection (continued)

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
1	10	10	Center-aligned PWM	High-true pulses (clear Output on match-up, set Output on match-down)
		01		Low-true pulses (set Output on match-up, clear Output on match-down)

Address: Base address + Ch offset + (8d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CnSC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by writing a 1 to the CHF bit. Writing a 0 to CHF has no effect. If another event occurs between the CHF sets and the write operation, the write operation has no effect; therefore, CHF remains set indicating another event has occurred. In this case a CHF interrupt request is not lost due to the delay in clearing the previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 Disable channel interrupts. 1 Enable channel interrupts.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
4 MSA	Channel Mode Select

Table continues on the next page...

TPMx_*Cn*SC field descriptions (continued)

Field	Description
	Used for further selections in the channel logic. Its functionality is dependent on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

32.3.5 Channel (n) Value (TPMx_*Cn*V)

These registers contain the captured TPM counter value for the input modes or the match value for the output modes.

In input capture mode, any write to a CnV register is ignored.

In compare modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to CnV Register Update . Additional writes to the CnV write buffer are ignored until the register has been updated.

Address: Base address + 10h offset + (8d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																VAL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_*Cn*V field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
VAL	Channel Value Captured TPM counter value of the input modes or the match value for the output modes. This field must be written with single 16-bit or 32-bit access.

32.3.6 Capture and Compare Status (TPMx_STATUS)

The STATUS register contains a copy of the status flag, CnSC[CHnF] for each TPM channel, as well as SC[TOF], for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by writing all ones to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. Writing a 1 to CHF clears it. Writing a 0 to CHF has no effect.

If another event occurs between the flag setting and the write operation, the write operation has no effect; therefore, CHF remains set indicating another event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF.

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							TOF	0				CH3F	CH2F	CH1F	CH0F
W								w1c					w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_STATUS field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TOF	Timer Overflow Flag See register description

Table continues on the next page...

TPMx_STATUS field descriptions (continued)

Field	Description
	0 TPM counter has not overflowed. 1 TPM counter has overflowed.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

32.3.7 Combine Channel Register (TPMx_COMBINE)

This register contains the control bits used to configure the combine channel modes for each pair of channels (n) and (n+1), where n is all the even numbered channels.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						COMSWAP1	COMBINE1	0						COMSWAP0	COMBINE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_COMBINE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 COMSWAP1	Combine Channels 2 and 3 Swap When set in combine mode, the odd channel is used for the input capture and 1st compare, the even channel is used for the 2nd compare. 0 Even channel is used for input capture and 1st compare. 1 Odd channel is used for input capture and 1st compare.
8 COMBINE1	Combine Channels 2 and 3 Enables the combine feature for channels 2 and 3. In input capture mode, the combined channels use the even channel input. In software compare modes, the even channel match asserts the output trigger and the odd channel match negates the output trigger. In PWM modes, the even channel match is used for the 1st compare and odd channel match for the 2nd compare. 0 Channels 2 and 3 are independent. 1 Channels 2 and 3 are combined.
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 COMSWAP0	Combine Channel 0 and 1 Swap

Table continues on the next page...

TPMx_COMBINE field descriptions (continued)

Field	Description
	When set in combine mode, the even channel is used for the input capture and 1st compare, the odd channel is used for the 2nd compare. 0 Even channel is used for input capture and 1st compare. 1 Odd channel is used for input capture and 1st compare.
0 COMBINE0	Combine Channels 0 and 1 Enables the combine feature for channels 0 and 1. In input capture mode, the combined channels use the even channel input. In software compare modes, the even channel match asserts the output trigger and the odd channel match negates the output trigger. In PWM modes, the even channel match is used for the 1st compare and odd channel match for the 2nd compare. 0 Channels 0 and 1 are independent. 1 Channels 0 and 1 are combined.

32.3.8 Channel Polarity (TPMx_POL)

This register defines the input and output polarity of each of the channels.

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												POL3	POL2	POL1	POL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_POL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 POL3	Channel 3 Polarity 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity 0 The channel polarity is active high. 1 The channel polarity is active low.
1 POL1	Channel 1 Polarity 0 The channel polarity is active high. 1 The channel polarity is active low.

Table continues on the next page...

TPMx_POL field descriptions (continued)

Field	Description
0 POL0	Channel 0 Polarity 0 The channel polarity is active high. 1 The channel polarity is active low.

32.3.9 Filter Control (TPMx_FILTER)

This register selects the filter value of the channel inputs, and an additional output delay value for the channel outputs. In PWM combine modes, the filter can effectively implements deadtime insertion.

Address: Base address + 78h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_FILTER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 CH3FVAL	Channel 3 Filter Value Selects the filter value for the channel input and the delay value for the channel output. The filter/delay is disabled when the value is zero, otherwise the filter/delay is configured as (CH3FVAL * 4) clock cycles.
11–8 CH2FVAL	Channel 2 Filter Value Selects the filter value for the channel input and the delay value for the channel output. The filter/delay is disabled when the value is zero, otherwise the filter/delay is configured as (CH2FVAL * 4) clock cycles.
7–4 CH1FVAL	Channel 1 Filter Value Selects the filter value for the channel input and the delay value for the channel output. The filter/delay is disabled when the value is zero, otherwise the filter/delay is configured as (CH1FVAL * 4) clock cycles.
CH0FVAL	Channel 0 Filter Value Selects the filter value for the channel input and the delay value for the channel output. The filter/delay is disabled when the value is zero, otherwise the filter/delay is configured as (CH0FVAL * 4) clock cycles.

32.3.10 Quadrature Decoder Control and Status (TPMx_QDCTRL)

This register has the control and status bits for the quadrature decoder mode.

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												QUADMODE	QUADIR	TOFDIR	QUADEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_QDCTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the quadrature decoder mode. 0 Phase encoding mode. 1 Count and direction encoding mode.
2 QUADIR	Counter Direction in Quadrature Decode Mode

Table continues on the next page...

TPMx_QDCTRL field descriptions (continued)

Field	Description
	Indicates the counting direction. 0 Counter direction is decreasing (counter decrement). 1 Counter direction is increasing (counter increment).
1 TOFDIR	Indicates if the TOF bit was set on the top or the bottom of counting. 0 TOF bit was set on the bottom of counting. There was an FTM counter decrement and FTM counter changes from its minimum value (zero) to its maximum value (MOD register). 1 TOF bit was set on the top of counting. There was an FTM counter increment and FTM counter changes from its maximum value (MOD register) to its minimum value (zero).
0 QUADEN	Enables the quadrature decoder mode. In this mode, the channel 0 and channel 1 inputs control the TPM counter direction and can only be used for software compare. The quadrature decoder mode has precedence over the other modes. 0 Quadrature decoder mode is disabled. 1 Quadrature decoder mode is enabled.

32.3.11 Configuration (TPMx_CONF)

This register selects the behavior in debug and wait modes and the use of an external global time base.

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				TRGSEL				TRGSRC	TRGPOL	0		CPOT	CROT	CSOO	CSOT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						GTBEEN	GTBSYNC	DBGMODE		DOZEEN	0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CONF field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
27–24 TRGSEL	Trigger Select Selects the input trigger to use for starting, reloading and/or pausing the counter. The source of the trigger (external or internal to the TPM) is configured by the TRGSRC field. This field should only be changed when the TPM counter is disabled. Refer to the chip configuration section for available external trigger options. The available internal trigger sources are listed below. 0001 Channel 0 pin input capture 0010 Channel 1 pin input capture 0011 Channel 0 or Channel 1 pin input capture 0100 Channel 2 pin input capture 0101 Channel 0 or Channel 2 pin input capture 0110 Channel 1 or Channel 2 pin input capture 0111 Channel 0 or Channel 1 or Channel 2 pin input capture 1000 Channel 3 pin input capture 1001 Channel 0 or Channel 3 pin input capture 1010 Channel 1 or Channel 3 pin input capture 1011 Channel 0 or Channel 1 or Channel 3 pin input capture 1100 Channel 2 or Channel 3 pin input capture 1101 Channel 0 or Channel 2 or Channel 3 pin input capture 1110 Channel 1 or Channel 2 or Channel 3 pin input capture 1111 Channel 0 or Channel 1 or Channel 2 or Channel 3 pin input capture
23 TRGSRC	Trigger Source Selects between internal (channel pin input capture) or external trigger sources. When selecting an internal trigger, the channel selected should be configured for input capture. Only a rising edge input capture can be used to initially start the counter using the CSOT configuration; either rising edge or falling edge input capture can be used to reload the counter using the CROT configuration; and the state of the channel input pin is used to pause the counter using the CPOT configuration. The channel polarity register can be used to invert the polarity of the channel input pins. This field should only be changed when the TPM counter is disabled. 0 Trigger source selected by TRGSEL is external. 1 Trigger source selected by TRGSEL is internal (channel pin input capture).
22 TRGPOL	Trigger Polarity Selects the polarity of the external trigger source. This field should only be changed when the TPM counter is disabled. 0 Trigger is active high. 1 Trigger is active low.
21–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 CPOT	Counter Pause On Trigger When enabled, the counter will pause incrementing while the trigger remains asserted (level sensitive). This field should only be changed when the TPM counter is disabled.
18 CROT	Counter Reload On Trigger

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
	When set, the TPM counter will reload with 0 (and initialize PWM outputs to their default value) when a rising edge is detected on the selected trigger input. The trigger input is ignored if the TPM counter is paused during debug mode or doze mode. This field should only be changed when the TPM counter is disabled. 0 Counter is not reloaded due to a rising edge on the selected input trigger 1 Counter is reloaded when a rising edge is detected on the selected input trigger
17 CSOO	Counter Stop On Overflow When set, the TPM counter will stop incrementing once the counter equals the MOD value and incremented (this also sets the TOF). Reloading the counter with 0 due to writing to the counter register or due to a trigger input does not cause the counter to stop incrementing. Once the counter has stopped incrementing, the counter will not start incrementing unless it is disabled and then enabled again, or a rising edge on the selected trigger input is detected when CSOT set. This field should only be changed when the TPM counter is disabled. 0 TPM counter continues incrementing or decrementing after overflow 1 TPM counter stops incrementing or decrementing after overflow.
16 CSOT	Counter Start on Trigger When set, the TPM counter will not start incrementing after it is enabled until a rising edge on the selected trigger input is detected. If the TPM counter is stopped due to an overflow, a rising edge on the selected trigger input will also cause the TPM counter to start incrementing again. The trigger input is ignored if the TPM counter is paused during debug mode or doze mode. This field should only be changed when the TPM counter is disabled. 0 TPM counter starts to increment immediately, once it is enabled. 1 TPM counter only starts to increment when it a rising edge on the selected input trigger is detected, after it has been enabled or after it has stopped due to overflow.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 GTBEEN	Global time base enable Configures the TPM to use an externally generated global time base counter. When an externally generated timebase is used, the internal TPM counter is not used by the channels but can be used to generate a periodic interruptor DMA request using the Modulo register and timer overflow flag. 0 All channels use the internally generated TPM counter as their timebase 1 All channels use an externally generated global timebase as their timebase
8 GTBSYNC	Global Time Base Synchronization When enabled, the TPM counter is synchronized to the global time base. It uses the global timebase enable, trigger and overflow to ensure the TPM counter starts incrementing at the same time as the global timebase, stops incrementing at the same time as the global timebase and is reset at the same time as the global timebase. This field should only be changed when the TPM counter is disabled. 0 Global timebase synchronization disabled. 1 Global timebase synchronization enabled.
7–6 DBGMODE	Debug Mode Configures the TPM behavior in debug mode. All other configurations are reserved.

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
	00 TPM counter is paused and does not increment during debug mode. Trigger inputs and input capture events are also ignored. 11 TPM counter continues in debug mode.
5 DOZEEN	Doze Enable Configures the TPM behavior in wait mode. 0 Internal TPM counter continues in Doze mode. 1 Internal TPM counter is paused and does not increment during Doze mode. Trigger inputs and input capture events are also ignored.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

32.4 Functional description

The following sections describe the TPM features.

32.4.1 Clock domains

The TPM module supports two clock domains.

The bus clock domain is used by the register interface and for synchronizing interrupts and DMA requests.

The TPM counter clock domain is used to clock the counter and prescaler along with the output compare and input capture logic. The TPM counter clock is considered asynchronous to the bus clock, can be a higher or lower frequency than the bus clock and can remain operational in Stop mode. Multiple TPM instances are all clocked by the same TPM counter clock in support of the external timebase feature.

32.4.1.1 Counter Clock Mode

The CMOD[1:0] bits in the SC register either disable the TPM counter or select one of two possible clock modes for the TPM counter. After any reset, CMOD[1:0] = 0:0 so the TPM counter is disabled.

The CMOD[1:0] bits may be read or written at any time. Disabling the TPM counter by writing zero to the CMOD[1:0] bits does not affect the TPM counter value or other registers, but must be acknowledged by the TPM counter clock domain before they read as zero.

The external clock input passes through a synchronizer clocked by the TPM counter clock to assure that counter transitions are properly aligned to counter clock transitions. Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must be less than half of the counter clock frequency.

32.4.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter.

The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and TPM counter.

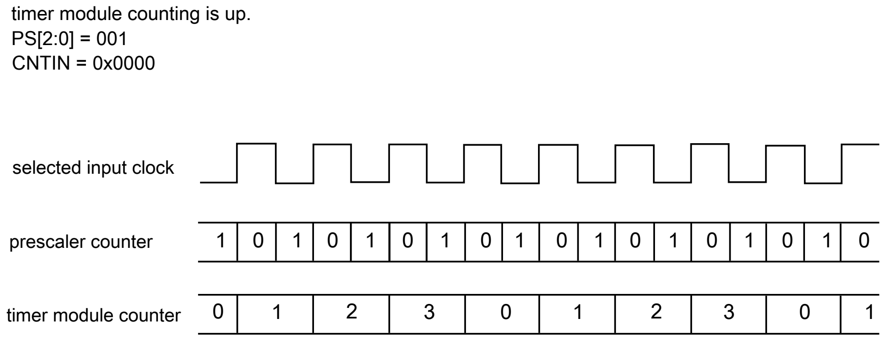


Figure 32-2. Example of the Prescaler Counter

32.4.3 Counter

The TPM has a 16-bit counter that is used by the channels either for input or output modes.

The counter updates from the selected clock divided by the prescaler.

The TPM counter has these modes of operation:

- up counting (see Up counting)
- up-down counting (see Up-down counting)

32.4.3.1 Up counting

Up counting is selected when SC[CPWMS] = 0.

The value of zero is loaded into the TPM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with zero.

The TPM period when using up counting is (MOD + 0x0001) × period of the TPM counter clock.

The TOF bit is set when the TPM counter changes from MOD to zero.

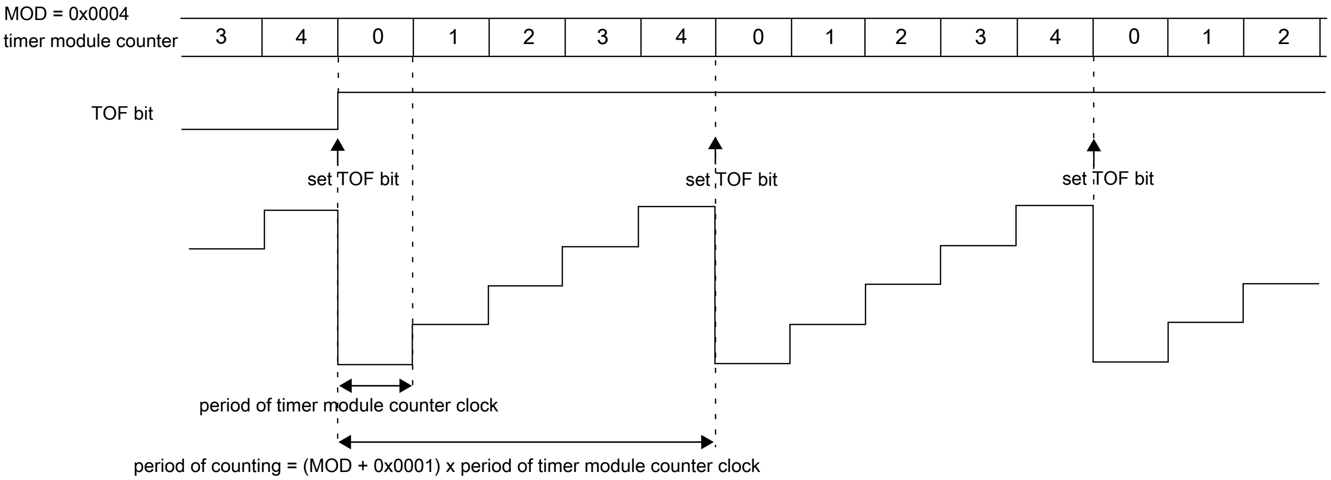

MOD = 0x0004
timer module counter
3
4
0
1
2
3
4
0
1
2
3
4
0
1
2
TOF bit
set TOF bit
set TOF bit
set TOF bit
period of timer module counter clock
period of counting = (MOD + 0x0001) x period of timer module counter clock

Figure 32-3. Example of TPM Up Counting

Note

- MOD = 0000 is a redundant condition. In this case, the TPM counter is always equal to MOD and the TOF bit is set in each rising edge of the TPM counter clock.

32.4.3.2 Up-down counting

Up-down counting is selected when SC[CPWMS] = 1. When configured for up-down counting, configuring CONF[MOD] to less than 2 is not supported.

The value of 0 is loaded into the TPM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to zero and the up-down counting restarts.

The TPM period when using up-down counting is 2 × MOD × period of the TPM counter clock.

The TOF bit is set when the TPM counter changes from MOD to (MOD – 1).

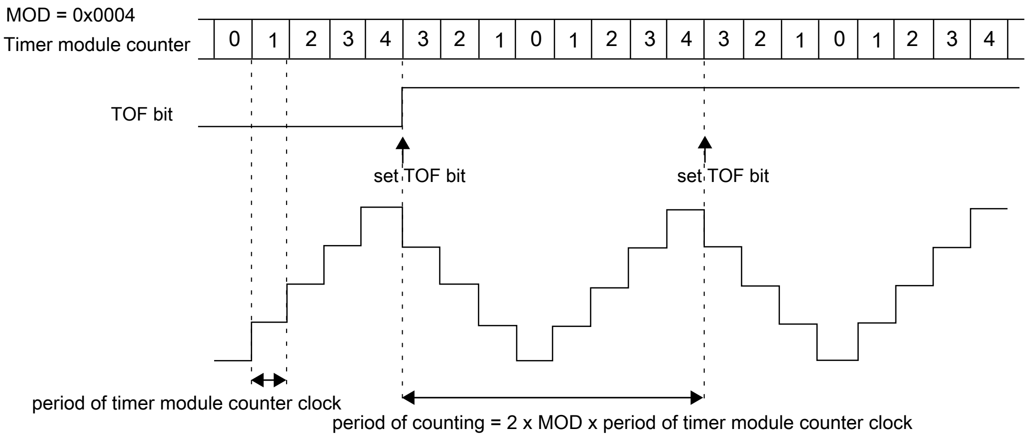

MOD = 0x0004
Timer module counter
0 1 2 3 4 3 2 1 0 1 2 3 4 3 2 1 0 1 2 3 4
TOF bit
set TOF bit
set TOF bit
period of timer module counter clock
period of counting = 2 x MOD x period of timer module counter clock

Figure 32-4. Example of up-down counting

32.4.3.3 Counter Reset

Any write to CNT resets the TPM counter and the channel outputs to their initial values (except for channels in output compare mode).

32.4.3.4 Global time base (GTB)

The global time base (GTB) is a TPM function that allows multiple TPM modules to share the same timebase. When the global time base is enabled (CONF[GTBEEN] = 1), the local TPM channels use the counter value, counter enable and overflow indication from the TPM generating the global time base. If the local TPM counter is not generating the global time base, then it can be used as an independent counter or pulse accumulator.

The local TPM counter can also be configured to synchronize to the global time base, by configuring (GTBSYNC = 1). When synchronized to the global time base, the local counter will use the counter enable and counter overflow indication from the TPM generating the global time base. This enables multiple TPM to be configured with the same phase, but with different periods (although the global time base must be configured with the longest period).

32.4.3.5 Counter trigger

The TPM counter can be configured to start, stop or reset in response to a hardware trigger input. The trigger input is synchronized to the asynchronous counter clock, so there is a 3 counter clock delay between the trigger assertion and the counter responding.

- When (CSOT = 1), the counter will not start incrementing until a rising edge is detected on the trigger input.
- When (CSOO= 1), the counter will stop incrementing whenever the TOF flag is set. The counter does not increment again unless it is disabled, or if CSOT = 1 and a rising edge is detected on the trigger input.
- When (CROT= 1), the counter will reset to zero as if an overflow occurred whenever a rising edge is detected on the trigger input.
- When (CPOT = 1), the counter will pause incrementing whenever the trigger input is asserted. The counter will continue incrementing when the trigger input negates.

The polarity of the external input trigger can be configured by the TRGPOL register bit.

When an internal trigger source is selected, the trigger input is selected from one or more channel input capture events. The input capture filters are used with the internal trigger sources and the POLn bits can be used to invert the polarity of the input channels. Note that following restrictions apply with input capture channel sources.

- When (CSOT = 1), the counter will only start incrementing on a rising edge on the channel input, provided ELSnA = 1.
- When (CROT= 1), the counter will reset to zero on either edge of the channel input, as configured by ELSnB:ELSnA.
- When (CPOT = 1), the counter will pause incrementing whenever the channel input is asserted.

32.4.4 Input Capture Mode

The input capture mode is selected when (CPWMS = 0), (MSnB:MSnA = 0:0), and (ELSnB:ELSnA ≠ 0:0).

When a selected edge occurs on the channel input, the current value of the TPM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1 (see the following figure).

When a channel is configured for input capture, the TPM_CHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is counter clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register are ignored in input capture mode.

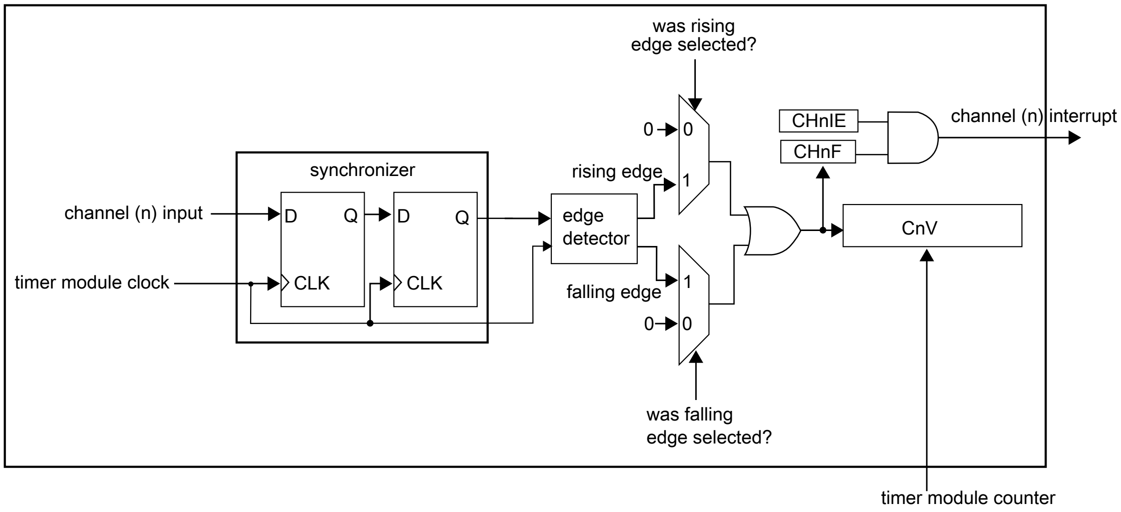


Figure 32-5. Input capture mode

The CHnF bit is set on the third rising edge of the counter clock after a valid edge occurs on the channel input.

32.4.5 Output Compare Mode

The output compare mode is selected when (CPWMS = 0), and (MSnB:MSnA = X:1).

In output compare mode, the TPM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared or toggled if MSnB is clear. If MSnB is set then the channel (n) output is pulsed high or low for as long as the counter matches the value in the CnV register.

When a channel is initially configured to output compare mode, the channel output updates with its negated value (logic 0 for set/toggle/pulse high and logic one for clear/ pulse low).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV).

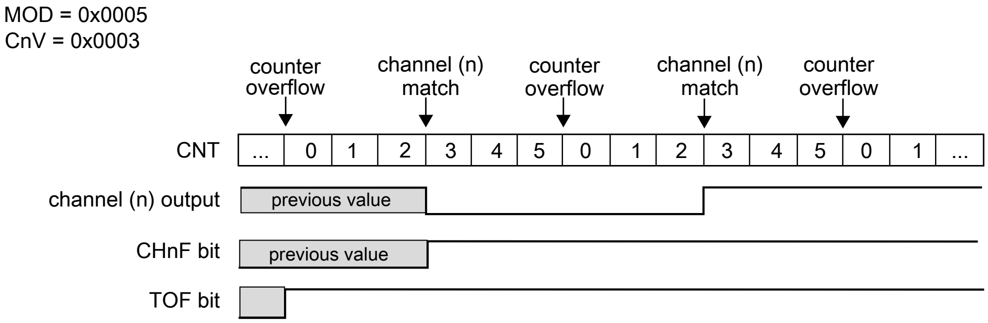


Figure 32-6. Example of the output compare mode when the match toggles the channel output

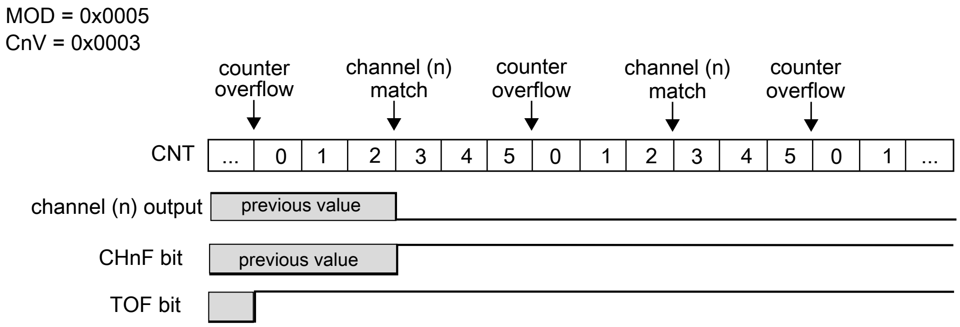


Figure 32-7. Example of the output compare mode when the match clears the channel output

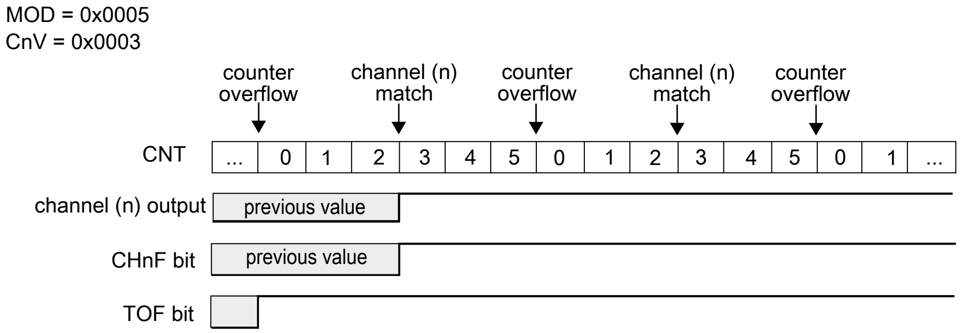


Figure 32-8. Example of the output compare mode when the match sets the channel output

It is possible to use the output compare mode with (ELSnB:ELSnA = 0:0). In this case, when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not modified and controlled by TPM.

32.4.6 Edge-Aligned PWM (EPWM) Mode

The edge-aligned mode is selected when (CPWMS = 0), and (MSnB:MSnA = 1:0).

The EPWM period is determined by (MOD + 0x0001) and the pulse width (duty cycle) is determined by CnV.

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an TPM.

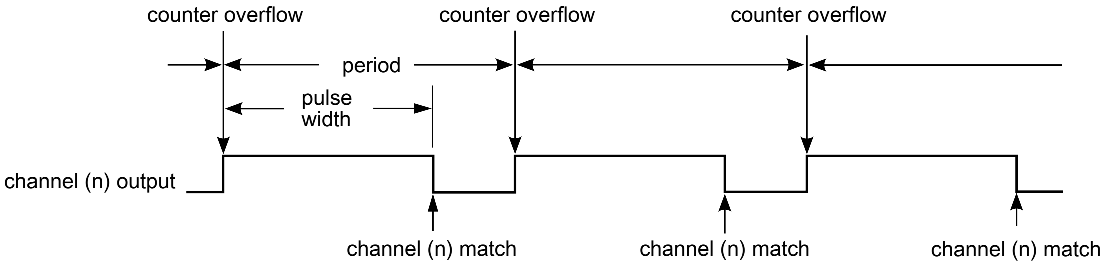


Figure 32-9. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by TPM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow (when the zero is loaded into the TPM counter), and it is forced low at the channel (n) match (TPM counter = CnV) (see the following figure).

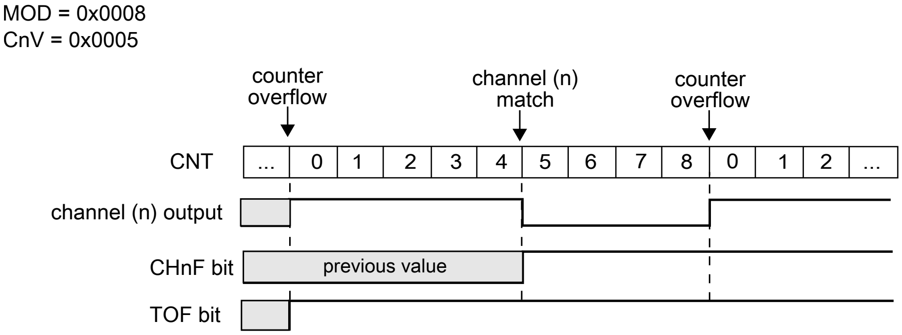


Figure 32-10. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow (when zero is loaded into the TPM counter), and it is forced high at the channel (n) match (TPM counter = CnV) (see the following figure).

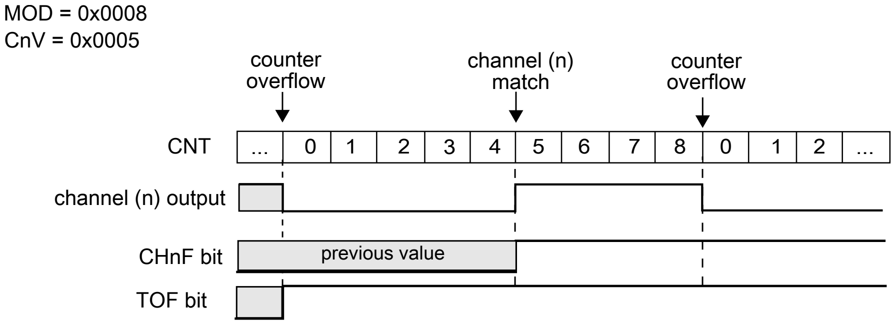


Figure 32-11. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal. If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set since there is never a channel (n) match. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

32.4.7 Center-Aligned PWM (CPWM) Mode

The center-aligned mode is selected when (CPWMS = 1) and (MSnB:MSnA = 1:0).

The CPWM pulse width (duty cycle) is determined by 2 × CnV and the period is determined by 2 × MOD (see the following figure). MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the TPM counter counts up until it reaches MOD and then counts down until it reaches zero.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV) when the TPM counting is down (at the begin of the pulse width) and when the TPM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are when the TPM counter is zero.

The other channel modes are not designed to be used with the up-down counter (CPWMS = 1). Therefore, all TPM channels should be used in CPWM mode when (CPWMS = 1).

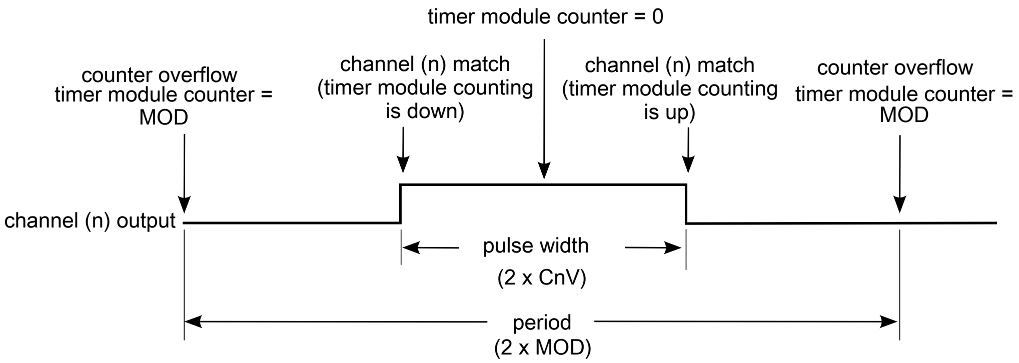


Figure 32-12. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the TPM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by TPM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (TPM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up (see the following figure).

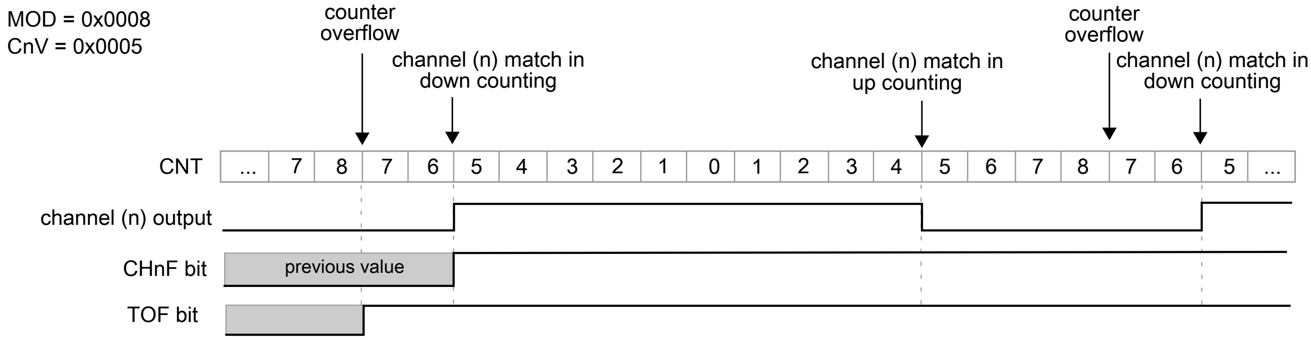


Figure 32-13. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (TPM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up (see the following figure).

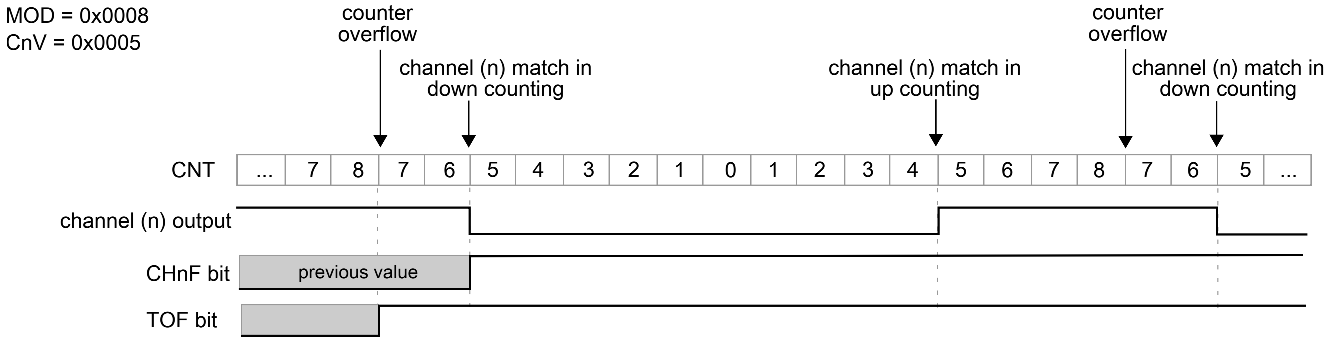


Figure 32-14. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) then the channel (n) output is a 0% duty cycle CPWM signal.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle CPWM signal, although the CHnF bit is set when the counter changes from incrementing to decrementing. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle CPWM signal.

32.4.8 Combine PWM mode

The Combine PWM mode is selected when:

- MSnB:MSnA = 10
- COMBINEn = 1
- QUADEN = 0, and
- CPWMS = 0

In Combine PWM mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V − C(n)V|).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = C(n)V). The CH(n+1)F bit is set and the channel (n +1) interrupt is generated, if CH(n+1)IE = 1, at the channel (n+1) match (TPM counter = C(n+1)V).

If channel (n) (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the beginning of the period (TPM counter is zero) and at the channel (n+1) match (TPM counter = C(n+1)V). It is forced high at the channel (n) match (TPM counter = C(n)V).

If channel (n) (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the beginning of the period (TPM counter is zero) and at the channel (n+1) match (TPM counter = C(n+1)V). It is forced low at the channel (n) match (TPM counter = C(n)V).

When (COMSWAPn = 1), then the channel (n) output is forced low or high at the beginning of the period (TPM counter is zero) and at the channel (n) match (TPM counter = C(n)V). It is forced high or low at the channel (n+1) match (TPM counter = C(n+1)V).

The channel (n+1) output is generated the same as the channel (n) output, but the output polarity is controlled by the channel (n+1) ELSnB:ELSnA configuration.

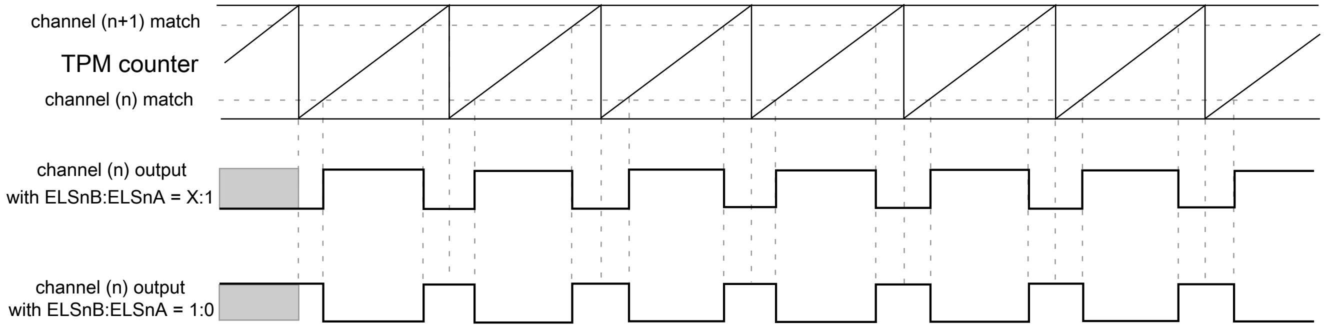


Figure 32-15. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

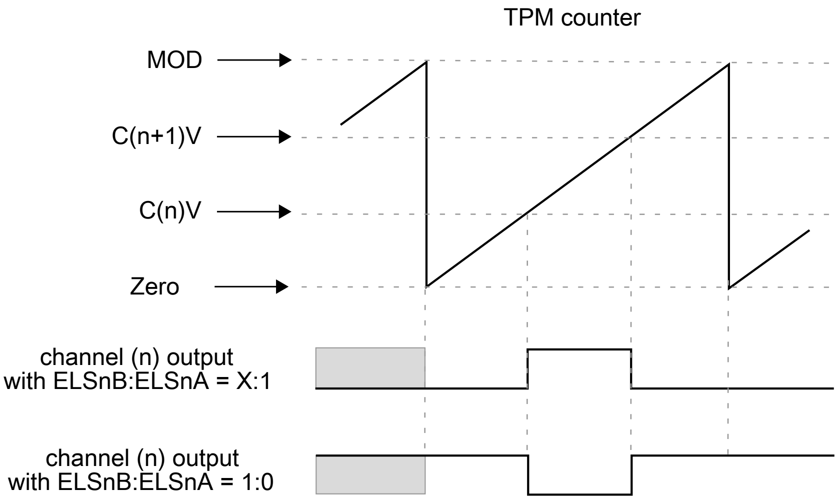


Figure 32-16. Channel (n) output if (C(n)V < MOD) and (C(n+1)V < MOD) and (C(n)V < C(n+1)V)

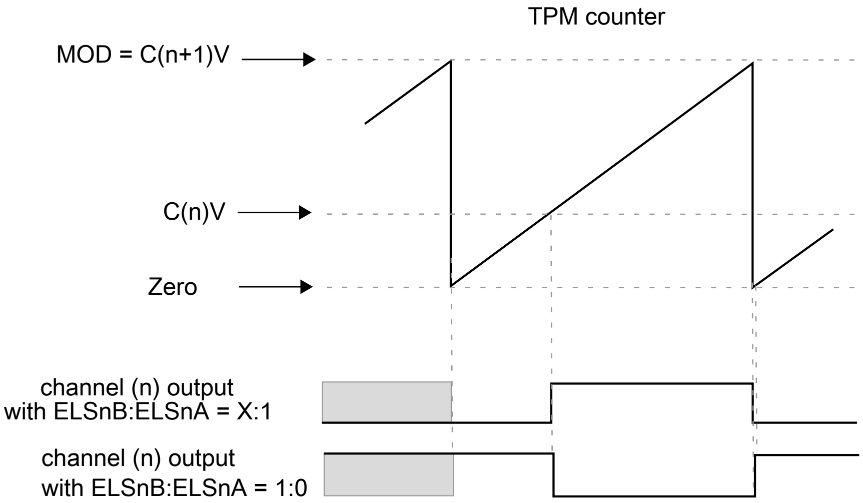


Figure 32-17. Channel (n) output if (C(n)V < MOD) and (C(n+1)V = MOD)

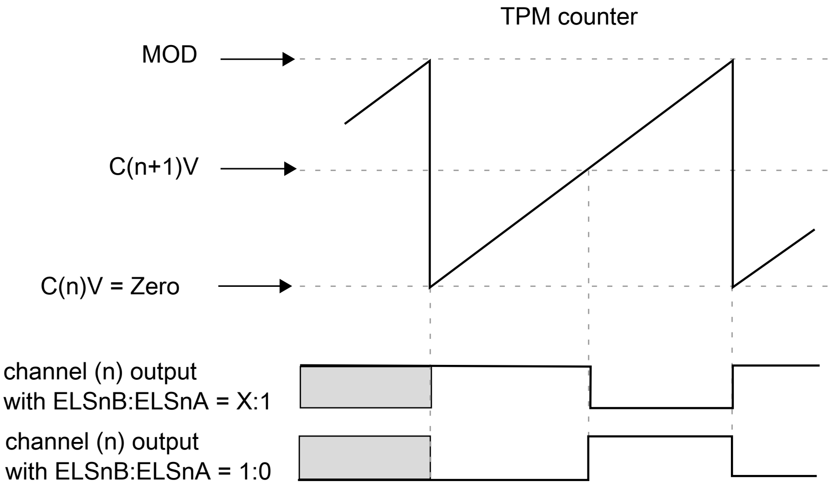


Figure 32-18. Channel (n) output if (C(n)V = zero) and (C(n+1)V < MOD)

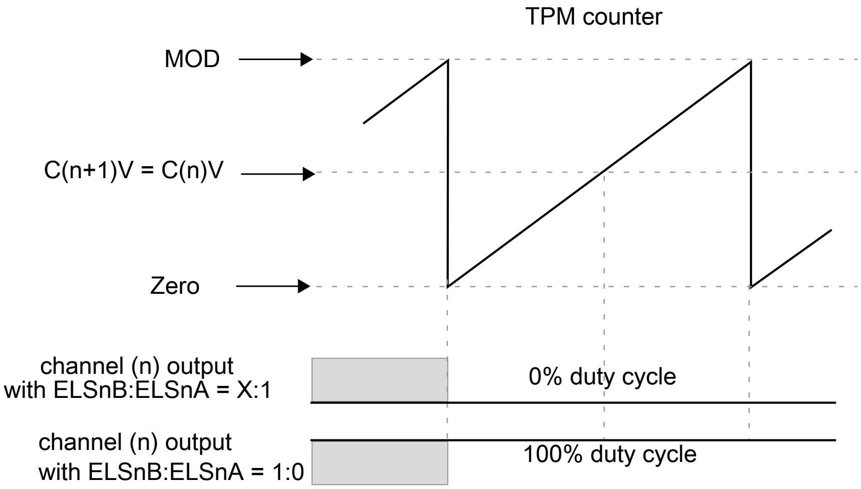


Figure 32-19. Channel (n) output if (C(n)V < MOD) and (C(n+1)V < MOD) and (C(n)V = C(n+1)V)

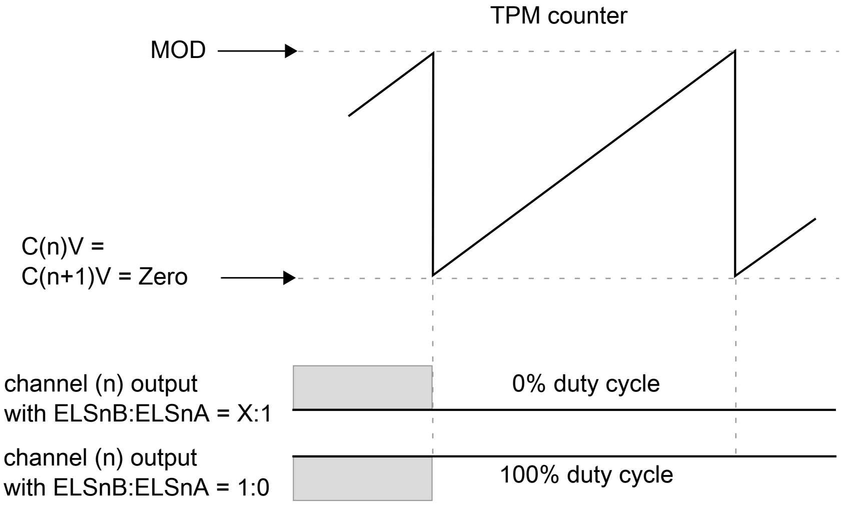


Figure 32-20. Channel (n) output if (C(n)V = C(n+1)V = zero)

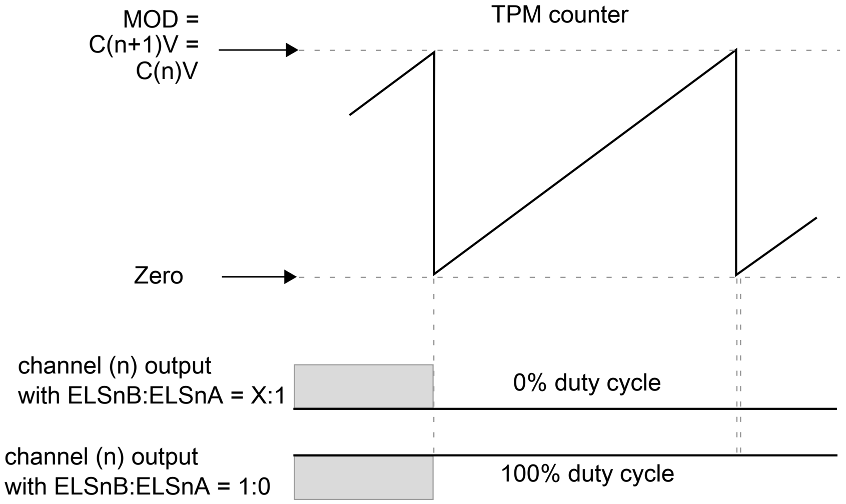


Figure 32-21. Channel (n) output if (C(n)V = C(n+1)V = MOD)

32.4.9 Combine Input Capture mode

The Combine Input Capture mode is selected if COMBINEn= 1 and MSnB:MSnA = 00 and ELSnB:ELSnA != 00. This mode allows to measure a pulse width of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode.

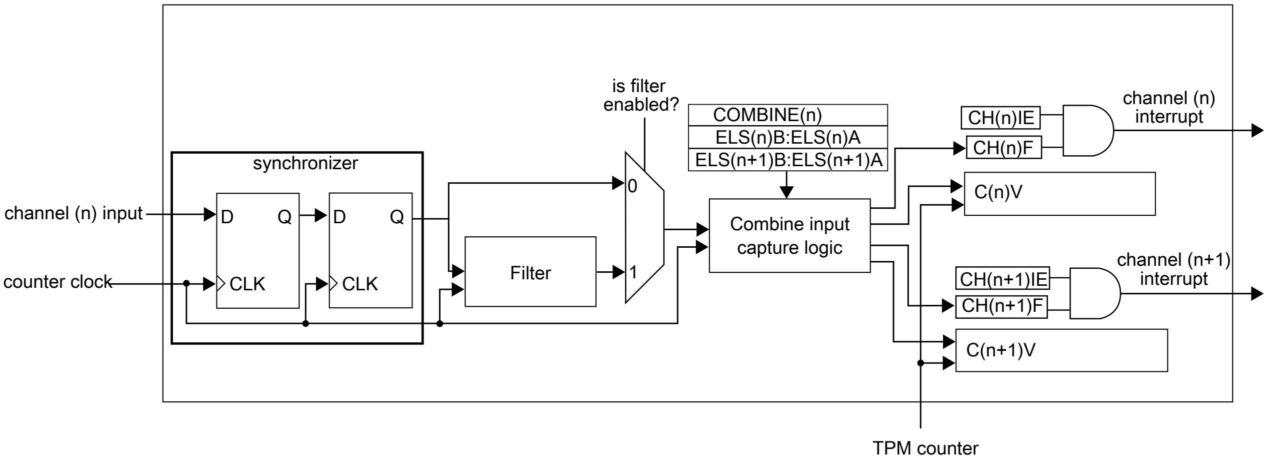


Figure 32-22. Combine Input Capture mode block diagram

The ELS(n)B:ELS(n)A bits select the edge that is captured by channel (n), and ELS(n +1)B:ELS(n+1)A bits select the edge that is captured by channel (n+1).

In the Combine Input Capture mode, only channel (n) input is used and channel (n+1) input is ignored, when COMSWAPn=1 then only channel (n+1) input is used and channel (n) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then CH(n)F bit is set and the channel (n) interrupt is generated (if CH(n)IE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input, then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

The C(n)V register stores the value of TPM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of TPM counter when the selected edge by channel (n+1) is detected at channel (n) input.

Note

- The CH(n)F, CH(n)IE, MS(n)A, ELS(n)B, and ELS(n)A bits are channel (n) bits.
- The CH(n+1)F, CH(n+1)IE, MS(n+1)A, ELS(n+1)B, and ELS(n+1)A bits are channel (n+1) bits.
- The Combine Input Capture mode must be used with ELS(n)B:ELS(n)A = 0:1 or 1:0, ELS(n+1)B:ELS(n+1)A = 0:1 or 1:0.

32.4.10 Input Capture Filter

The input capture filter function is only in input capture mode, or in software compare mode when quadrature decoder mode is enabled.

First, the input signal is synchronized by the counter clock. Following synchronization, the input signal enters the filter block. See the following figure.

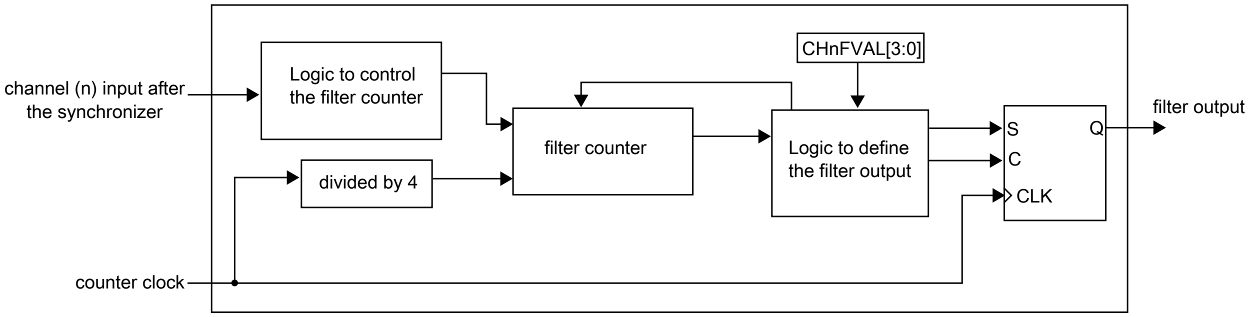

channel (n) input after the synchronizer
Logic to control the filter counter
divided by 4
counter clock
filter counter
CHnFVAL[3:0]
Logic to define the filter output
S
C
CLK
Q
filter output

Figure 32-23. Channel input filter

When there is a state change in the input signal, the counter is reset and starts counting up. As long as the new state is stable on the input, the counter continues to increment. When the counter is equal to (CHnFVAL[3:0] × 4), the state change of the input signal is validated.

If the opposite edge appears on the input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by (CHnFVAL[3:0] × 4 counter clocks) is regarded as a glitch and is not passed through the filter. A timing diagram of the input filter is shown in the following figure.

The filter function is disabled when CHnFVAL[3:0] bits are zero. In this case, the input signal is delayed by 2 rising edges of the counter clock. If (CHnFVAL[3:0] ≠ 0000), then the input signal is delayed by the minimum pulse width (CHnFVAL[3:0] × 4 system clocks) plus a further 3 rising edges of the system clock: two rising edges to the synchronizer, plus one more to the edge detector. In other words, CHnF is set (3 + 4 × CHnFVAL[3:0]) counter clock periods after a valid edge occurs on the channel input.

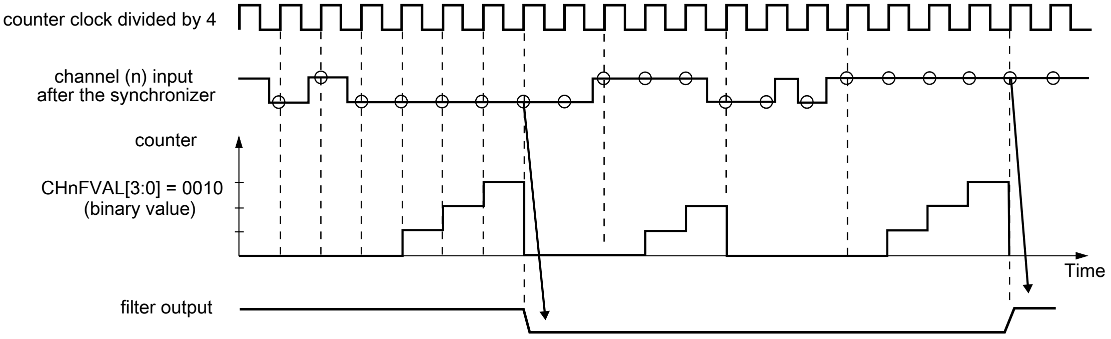


Figure 32-24. Channel input filter example

32.4.11 Deadtime insertion

The deadtime insertion is enabled in PWM combine modes when CHnFVAL is non-zero. The deadtime delay that is used for each TPM channel is defined as (CHnFVAL[3:0] x 4).

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (TPM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (TPM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (TPM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly, when the channel (n+1) match (TPM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n +1) output is cleared.

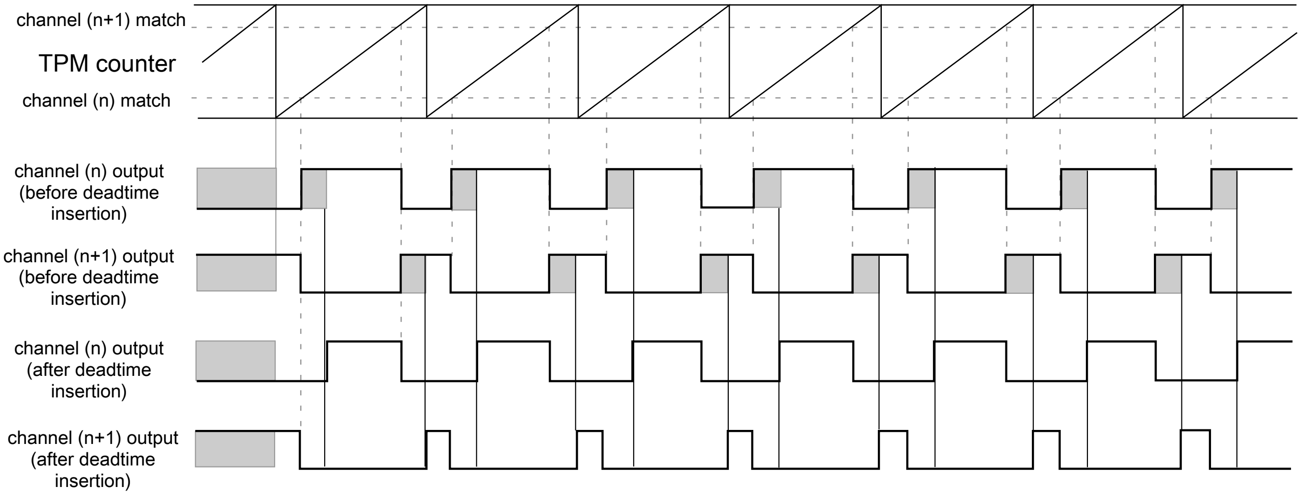

channel (n+1) match
TPM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 32-25. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 1

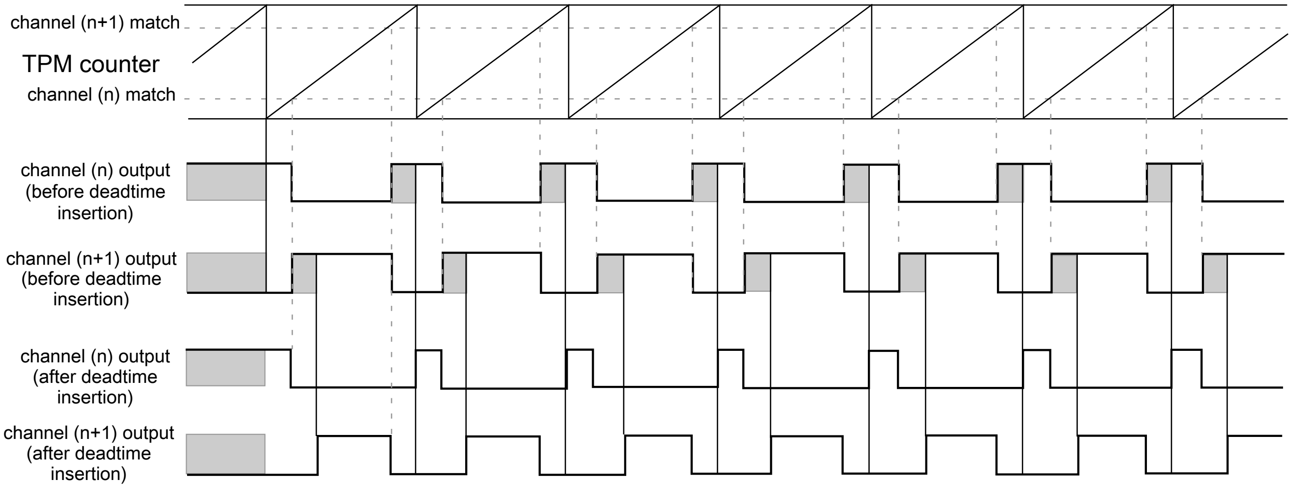

channel (n+1) match
TPM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 32-26. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 1

32.4.12 Quadrature Decoder mode

The Quadrature Decoder mode is selected if (QUADEN = 1). The Quadrature Decoder mode uses the channel 0 (phase A) and channel 1 (phase B) input signals to control the TPM counter increment and decrement. The following figure shows the quadrature decoder block diagram.

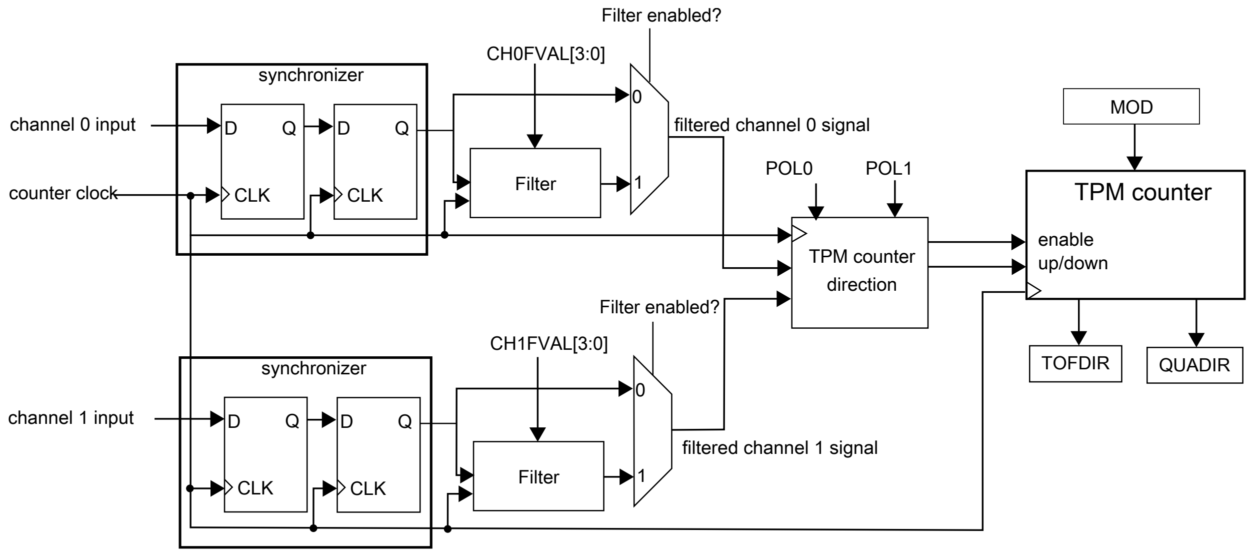


Figure 32-27. Quadrature Decoder block diagram

The input capture filter and channel polarity registers are used to configure the input filter and polarity for the channel 0 and channel 1 inputs in quadrature decode mode.

Note

Notice that the TPM counter is clocked by the channel 0 and channel 1 input signals when quadrature decoder mode is selected. Therefore In quadrature decoder mode, channel 0 and channel 1 can only be used in software compare mode and other TPM channels can only be used in input capture or output compare modes.

The QUADMODE selects the encoding mode used in the Quadrature Decoder mode. If QUADMODE = 1, then the count and direction encoding mode is enabled; see the following figure. In this mode, the channel 1 input value indicates the counting direction, and the channel 0 input defines the counting rate. The TPM counter is updated when there is a rising edge at channel 0 input signal.

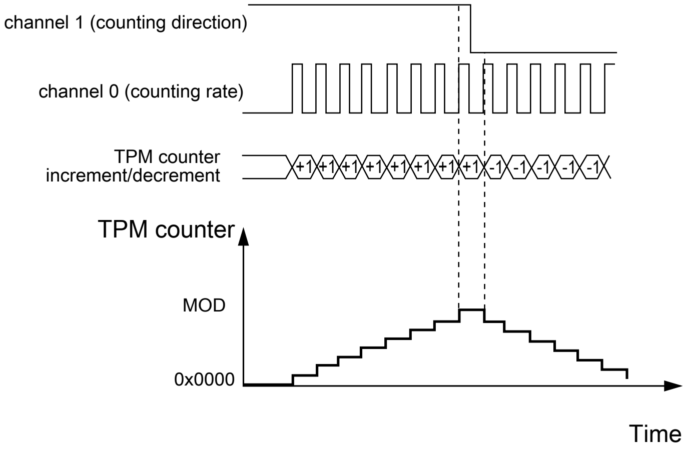

channel 1 (counting direction)
channel 0 (counting rate)
TPM counter increment/decrement
+1 +1 +1 +1 +1 +1 +1 +1 -1 -1 -1 -1 -1
TPM counter
MOD
0x0000
Time

Figure 32-28. Quadrature Decoder – Count and Direction Encoding mode

If QUADMODE = 0, then the Phase Encoding mode is enabled; see the following figure. In this mode, the relationship between channel 0 and channel 1 signals indicates the counting direction, and channel 0 and channel 1 signals define the counting rate. The TPM counter is updated when there is an edge either at the channel 0 or channel 1 signals.

If CH0POL= 0 and CH1POL = 0, then the TPM counter increment happens when:

- there is a rising edge at channel 0 signal and channel 1 signal is at logic zero;
- there is a rising edge at channel 1 signal and channel 0 signal is at logic one;
- there is a falling edge at channel 1 signal and channel 0 signal is at logic zero;
- there is a falling edge at channel 0 signal and channel 1 signal is at logic one;

and the TPM counter decrement happens when:

- there is a falling edge at channel 0 signal and channel 1 signal is at logic zero;
- there is a falling edge at channel 1 signal and channel 0 signal is at logic one;
- there is a rising edge at channel 1 signal and channel 0 signal is at logic zero;
- there is a rising edge at channel 0 signal and channel 1 signal is at logic one.

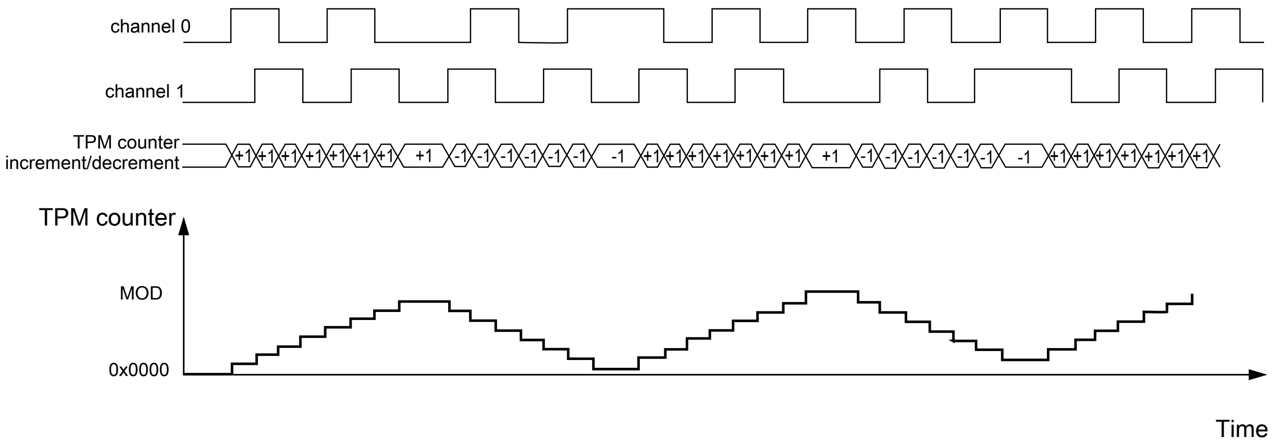


Figure 32-29. Quadrature Decoder – Phase Encoding mode

The following figure shows the TPM counter overflow in up counting. In this case, when the TPM counter changes from MOD to zero, TOF and TOFDIR bits are set. TOF bit indicates the TPM counter overflow occurred. TOFDIR indicates the counting was up when the TPM counter overflow occurred.

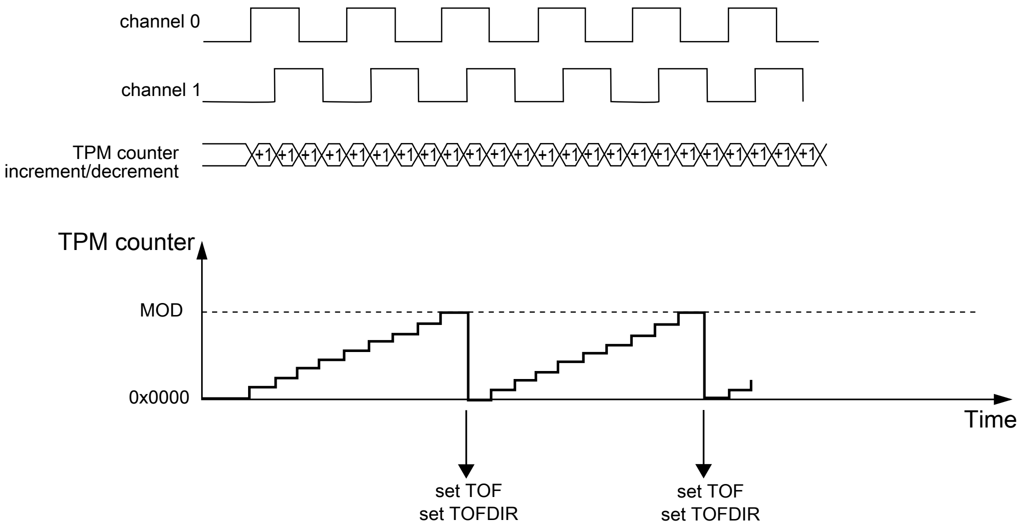


Figure 32-30. TPM Counter overflow in up counting for Quadrature Decoder mode

The following figure shows the TPM counter overflow in down counting. In this case, when the TPM counter changes from zero to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the TPM counter overflow occurred. TOFDIR indicates the counting was down when the TPM counter overflow occurred.

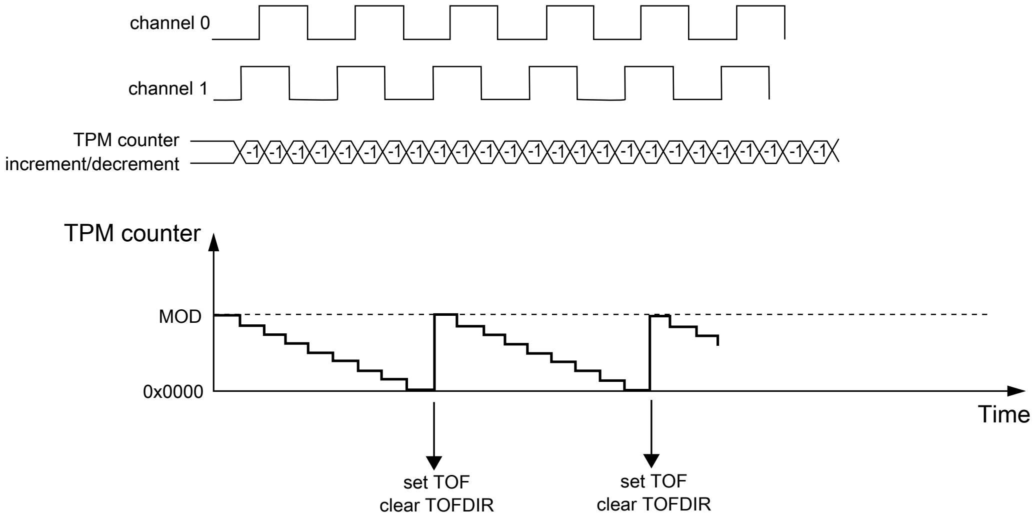


Figure 32-31. TPM counter overflow in down counting for Quadrature Decoder mode

32.4.13 Registers Updated from Write Buffers

32.4.13.1 MOD Register Update

If (CMOD[1:0] = 0:0) then MOD register is updated when MOD register is written.

If (CMOD[1:0] ≠ 0:0), then MOD register is updated according to the CPWMS bit, that is:

- If the selected mode is not CPWM then MOD register is updated after MOD register was written and the TPM counter changes from MOD to zero.
- If the selected mode is CPWM then MOD register is updated after MOD register was written and the TPM counter changes from MOD to (MOD – 1).

32.4.13.2 CnV Register Update

If (CMOD[1:0] = 0:0) then CnV register is updated when CnV register is written.

If (CMOD[1:0] ≠ 0:0), then CnV register is updated according to the selected mode, that is:

- If the selected mode is output compare then CnV register is updated on the next TPM counter increment (end of the prescaler counting) after CnV register was written.

- If the selected mode is EPWM then CnV register is updated after CnV register was written and the TPM counter changes from MOD to zero.
- If the selected mode is CPWM then CnV register is updated after CnV register was written and the TPM counter changes from MOD to (MOD – 1).

32.4.14 DMA

The channel and overflow flags generate a DMA transfer request according to DMA and CHnIE/TOIE bits.

See the following table for more information.

Table 32-3. DMA Transfer Request

DMA	CHnIE/ TOIE	Channel/Overflow DMA Transfer Request	Channel/Overflow Interrupt
0	0	The channel/overflow DMA transfer request is not generated.	The channel/overflow interrupt is not generated.
0	1	The channel/overflow DMA transfer request is not generated.	The channel/overflow interrupt is generated if (CHnF/TOF = 1).
1	0	The channel/overflow DMA transfer request is generated if (CHnF/TOF = 1).	The channel/overflow interrupt is not generated.
1	1	The channel/overflow DMA transfer request is generated if (CHnF/TOF = 1).	The channel/overflow interrupt is generated if (CHnF/TOF = 1).

If DMA = 1, the CHnF/TOF bit can be cleared either by DMA transfer done or writing a one to CHnF/TOF bit (see the following table).

Table 32-4. Clear CHnF/TOF Bit

DMA	How CHnF/TOF Bit Can Be Cleared
0	CHnF/TOF bit is cleared by writing a 1 to CHnF/TOF bit.
1	CHnF/TOF bit is cleared either when the DMA transfer is done or by writing a 1 to CHnF/TOF bit.

32.4.15 Output triggers

The TPM generates output triggers for the counter and each channel that can be used to trigger events in other peripherals.

The counter trigger asserts whenever the TOF is set and remains asserted until the next increment.

Each TPM channel generates both a pre-trigger output and a trigger output. The pre-trigger output asserts whenever the CHnF is set, the trigger output asserts on the first counter increment after the pre-trigger asserts, and then both the trigger and pre-trigger negate on the first counter increment after the trigger asserts.

When (COMBINEn = 1) in output compare modes, the pre-trigger output for both channel (n) and channel (n+1) will assert when CH(n)F is set and will negate when CH(n +1)F is set. The trigger continues to assert on the first counter increment after the pre-trigger asserts and negates at the same time as the pre-trigger negation.

32.4.16 Reset Overview

The TPM is reset whenever any chip reset occurs.

When the TPM exits from reset:

- the TPM counter and the prescaler counter are zero and are stopped (CMOD[1:0] = 0:0);
- the timer overflow interrupt is zero;
- the channels interrupts are zero;
- the channels are in input capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by TPM (ELS(n)B:ELS(n)A = 0:0).

32.4.17 TPM Interrupts

This section describes TPM interrupts.

32.4.17.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

32.4.17.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

Chapter 33
Periodic interrupt timer (PIT)

The periodic interrupt timer (PIT) module consists of an array of timers that can be used to generate interrupts and trigger DMA channels.

33.1 Introduction

The PIT module is an array of timers that can be used to raise interrupts and trigger DMA channels.

33.1.1 Block diagram

The following figure shows the block diagram of the PIT module.

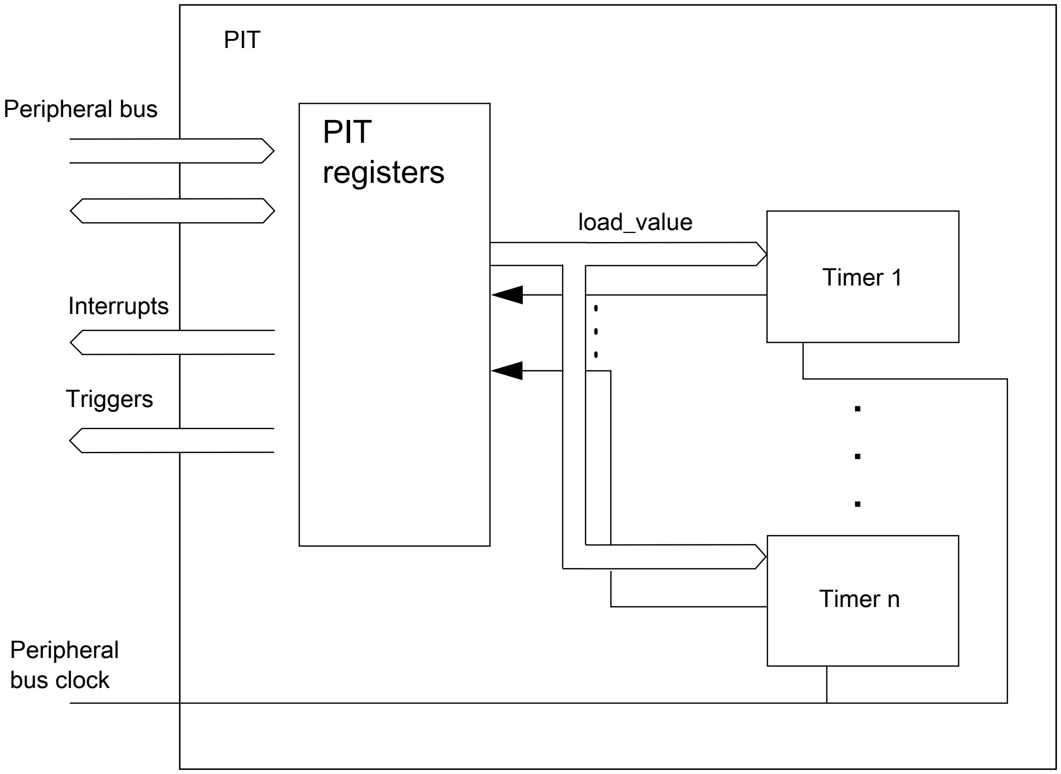


Figure 33-1. Block diagram of the PIT

NOTE

See the chip-specific PIT information for the number of PIT channels used in this MCU.

33.1.2 Features

The main features of this block are:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

33.2 Signal description

The PIT module has no external pins.

33.3 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT module.

- Reserved registers will read as 0, writes will have no effect.
- See the chip-specific PIT information for the number of PIT channels used in this MCU.

PIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0006h	33.3.1/639
4003_70E0	PIT Upper Lifetime Timer Register (PIT_LTMR64H)	32	R	0000_0000h	33.3.2/640
4003_70E4	PIT Lower Lifetime Timer Register (PIT_LTMR64L)	32	R	0000_0000h	33.3.3/641
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	33.3.4/641
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R	0000_0000h	33.3.5/642
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	33.3.6/642
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	33.3.7/643
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	33.3.4/641
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R	0000_0000h	33.3.5/642
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	33.3.6/642
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	33.3.7/643

33.3.1 PIT Module Control Register (PIT_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Access: User read/write

Address: 4003_7000h base + 0h offset = 4003_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Reserved	MDIS	FRZ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

PIT_MCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 MDIS	Module Disable - (PIT section) Disables the standard timers. This field must be enabled before any other setup is done. 0 Clock for standard PIT timers is enabled. 1 Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 Timers continue to run in Debug mode. 1 Timers are stopped in Debug mode.

33.3.2 PIT Upper Lifetime Timer Register (PIT_LTMR64H)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

Access: User read only

Address: 4003_7000h base + E0h offset = 4003_70E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTH																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LTMR64H field descriptions

Field	Description
LTH	Life Timer value Shows the timer value of timer 1. If this register is read at a time t1, LTMR64L shows the value of timer 0 at time t1.

33.3.3 PIT Lower Lifetime Timer Register (PIT_LTMR64L)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

To use LTMR64H and LTMR64L, timer 0 and timer 1 need to be chained. To obtain the correct value, first read LTMR64H and then LTMR64L. LTMR64H will have the value of CVAL1 at the time of the first access, LTMR64L will have the value of CVAL0 at the time of the first access, therefore the application does not need to worry about carry-over effects of the running counter.

Access: User read only

Address: 4003_7000h base + E4h offset = 4003_70E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LTMR64L field descriptions

Field	Description
LTL	Life Timer value Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read.

33.3.4 Timer Load Value Register (PIT_LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Access: User read/write

Address: 4003_7000h base + 100h offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LDVAL*n* field descriptions

Field	Description
TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

33.3.5 Current Timer Value Register (PIT_CVAL*n*)

These registers indicate the current timer position.

Access: User read only

Address: 4003_7000h base + 104h offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TVL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_CVAL*n* field descriptions

Field	Description
TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

33.3.6 Timer Control Register (PIT_TCTRL*n*)

These registers contain the control bits for each timer.

Access: User read/write

Address: 4003_7000h base + 108h offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CHN	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TCTRL*n* field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CHN	Chain Mode When activated, Timer n-1 needs to expire before timer n can decrement by 1. Timer 0 cannot be chained. 0 Timer is not chained. 1 Timer is chained to previous timer. For example, for Channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 Interrupt requests from Timer n are disabled. 1 Interrupt will be requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 Timer n is disabled. 1 Timer n is enabled.

33.3.7 Timer Flag Register (PIT_TFLG*n*)

These registers hold the PIT interrupt flags.

Access: User read/write

Address: 4003_7000h base + 10Ch offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLG*n* field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 Timeout has not yet occurred. 1 Timeout has occurred.

33.4 Functional description

This section provides the functional description of the module.

33.4.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line.

33.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

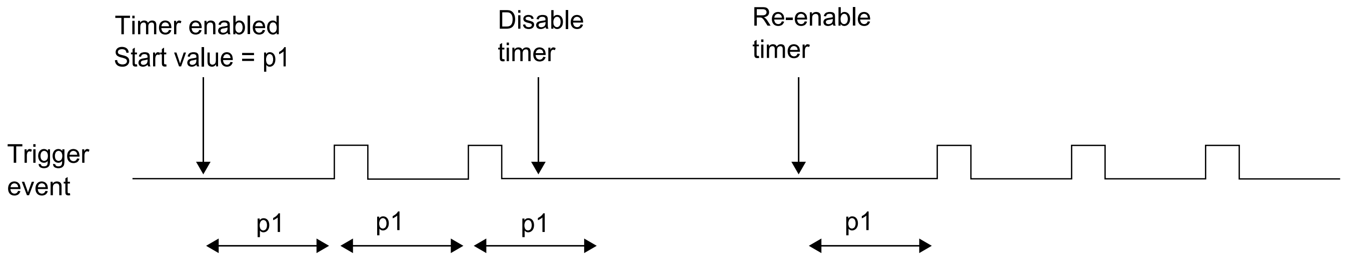


Figure 33-2. Stopping and starting a timer

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

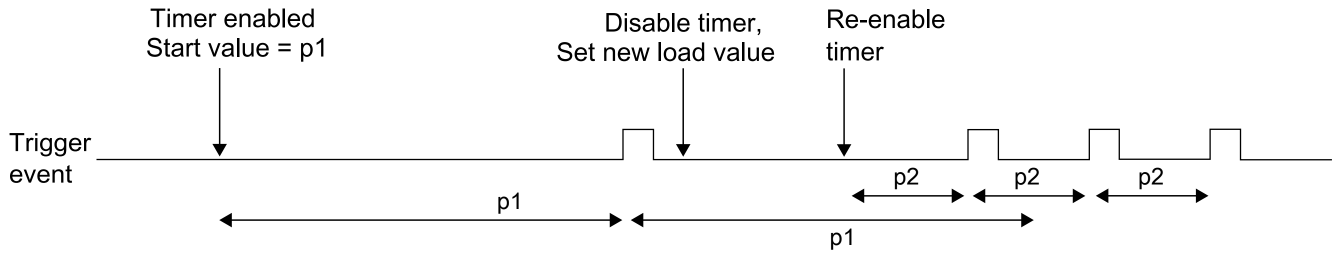


Figure 33-3. Modifying running timer period

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

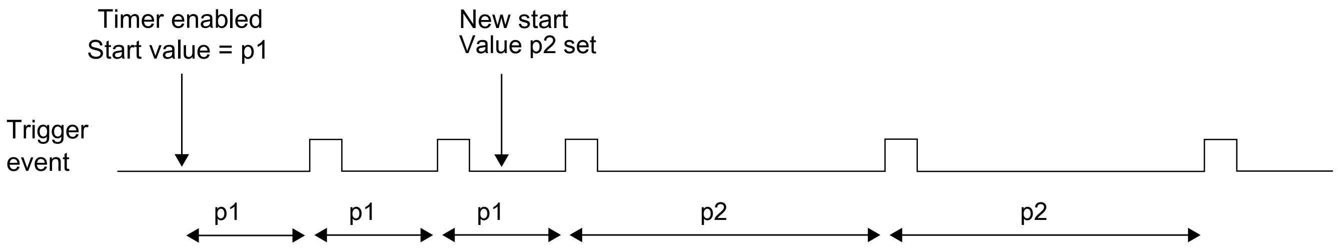


Figure 33-4. Dynamically setting a new load value

33.4.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

33.4.2 Interrupts

All the timers support interrupt generation. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

33.4.3 Chained timers

When a timer has chain mode enabled, it will only count after the previous timer has expired. So if timer n-1 has counted down to 0, counter n will decrement the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

33.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1


// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

33.6 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.
- Timers 1 and 2 are available.
- An interrupt shall be raised every 1 minute.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN]. TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2
```

```
// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

33.7 Example configuration for the lifetime timer

To configure the lifetimer timer, channels 0 and 1 need to be chained together.

First the PIT module needs to be activated by writing a 0 to the MDIS bit in the CTRL register, then the LDVAL registers need to be set to the maximum value.

The timer is a downcounter.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1


// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

To access the lifetime, read first LTMR64H and then LTMR64L.

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

Chapter 34
Low-power timer (LPTMR)

The low-power timer (LPTMR) can be configured to operate as a time counter (with optional prescaler) or as a pulse counter (with optional glitch filter) across all power modes, including the low leakage modes.

34.1 Introduction

The low-power timer (LPTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter, across all power modes, including the low-leakage modes. It can also continue operating through most system reset events, allowing it to be used as a time of day counter.

34.1.1 Features

The features of the LPTMR module include:

- 16-bit time counter or pulse counter with compare
  - Optional interrupt can generate asynchronous wakeup from any low-power mode
  - Hardware trigger output
  - Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
  - Rising-edge or falling-edge

34.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 34-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally in Pulse Counter mode, but counter does not increment in Time Counter mode.

34.2 LPTMR signal descriptions

Table 34-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR0_ALT*n*	I	Pulse Counter Input pin

34.2.1 Detailed signal descriptions

Table 34-3. LPTMR interface—detailed signal descriptions

Signal	I/O	Description	
LPTMR_ALT*n*	I	Pulse Counter Input The LPTMR can select one of the input pins to be used in Pulse Counter mode.	
		State meaning	Assertion—If configured for pulse counter mode with active-high input, then assertion causes the CNR to increment. Deassertion—If configured for pulse counter mode with active-low input, then deassertion causes the CNR to increment.
		Timing	Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.

34.3 Memory map and register definition

LPTMR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Low Power Timer Control Status Register (LPTMR0_CSR)	32	R/W	0000_0000h	34.3.1/651
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	34.3.2/652
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	34.3.3/654
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R/W	0000_0000h	34.3.4/654

34.3.1 Low Power Timer Control Status Register (LPTMRx_CSR)

Address: 4004_0000h base + 0h offset = 4004_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CSR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TCF	Timer Compare Flag TCF is set when the LPTMR is enabled and the CNR equals the CMR and increments. TCF is cleared when the LPTMR is disabled or a logic 1 is written to it. 0 The value of CNR is not equal to CMR and increments. 1 The value of CNR is equal to CMR and increments.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set. 0 Timer interrupt disabled. 1 Timer interrupt enabled.
5–4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. TPS must be altered only when the LPTMR is disabled. The input connections vary by device. See the for information on the connections to these inputs. 00 Pulse counter input 0 is selected.

Table continues on the next page...

LPTMRx_CSR field descriptions (continued)

Field	Description
	01 Pulse counter input 1 is selected. 10 Pulse counter input 2 is selected. 11 Pulse counter input 3 is selected.
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. TPP must be changed only when the LPTMR is disabled. 0 Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1 Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
2 TFC	Timer Free-Running Counter When clear, TFC configures the CNR to reset whenever TCF is set. When set, TFC configures the CNR to reset on overflow. TFC must be altered only when the LPTMR is disabled. 0 CNR is reset whenever TCF is set. 1 CNR is reset on overflow.
1 TMS	Timer Mode Select Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0 Time Counter mode. 1 Pulse Counter mode.
0 TEN	Timer Enable When TEN is clear, it resets the LPTMR internal logic, including the CNR and TCF. When TEN is set, the LPTMR is enabled. While writing 1 to this field, CSR[5:1] must not be altered. 0 LPTMR is disabled and internal logic is reset. 1 LPTMR is enabled.

34.3.2 Low Power Timer Prescale Register (LPTMRx_PSR)

Address: 4004_0000h base + 4h offset = 4004_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									PRESCALE				PBYP	PCS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_PSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–3 PRESCALE	Prescale Value Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. PRESCALE must be altered only when the LPTMR is disabled. 0000 Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001 Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010 Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011 Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100 Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101 Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110 Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111 Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000 Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001 Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010 Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011 Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100 Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101 Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110 Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111 Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.
2 PBYP	Prescaler Bypass When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR. When PBYP is clear, the CNR is clocked by the output of the prescaler/glitch filter. PBYP must be altered only when the LPTMR is disabled. 0 Prescaler/glitch filter is enabled. 1 Prescaler/glitch filter is bypassed.
PCS	Prescaler Clock Select Selects the clock to be used by the LPTMR prescaler/glitch filter. PCS must be altered only when the LPTMR is disabled. The clock connections vary by device. **NOTE:** See the chip configuration details for information on the connections to these inputs.

Table continues on the next page...

LPTMRx_PSR field descriptions (continued)

Field	Description
	00 Prescaler/glitch filter clock 0 selected. 01 Prescaler/glitch filter clock 1 selected. 10 Prescaler/glitch filter clock 2 selected. 11 Prescaler/glitch filter clock 3 selected.

34.3.3 Low Power Timer Compare Register (LPTMRx_CMR)

Address: 4004_0000h base + 8h offset = 4004_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COMPARE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COMPARE	Compare Value When the LPTMR is enabled and the CNR equals the value in the CMR and increments, TCF is set and the hardware trigger asserts until the next time the CNR increments. If the CMR is 0, the hardware trigger will remain asserted until the LPTMR is disabled. If the LPTMR is enabled, the CMR must be altered only when TCF is set.

34.3.4 Low Power Timer Counter Register (LPTMRx_CNR)

NOTE

See LPTMR counter for details on how to read counter value.

Address: 4004_0000h base + Ch offset = 4004_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNTER															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNTER	Counter Value

34.4 Functional description

34.4.1 LPTMR power and reset

The LPTMR remains powered in all power modes, including low-leakage modes. If the LPTMR is not required to remain operating during a low-power mode, then it must be disabled before entering the mode.

The LPTMR is reset only on global Power On Reset (POR) or Low Voltage Detect (LVD). When configuring the LPTMR registers, the CSR must be initially written with the timer disabled, before configuring the PSR and CMR. Then, CSR[TIE] must be set as the last step in the initialization. This ensures the LPTMR is configured correctly and the LPTMR counter is reset to zero following a warm reset.

34.4.2 LPTMR clocking

The LPTMR prescaler/glitch filter can be clocked by one of the four clocks. The clock source must be enabled before the LPTMR is enabled.

NOTE

The clock source selected need to be configured to remain enabled in low-power modes, otherwise the LPTMR will not operate during low-power modes.

In Pulse Counter mode with the prescaler/glitch filter bypassed, the selected input source directly clocks the CNR and no other clock source is required. To minimize power in this case, configure the prescaler clock source for a clock that is not toggling.

NOTE

The clock source or pulse input source selected for the LPTMR should not exceed the frequency f_{LPTMR} defined in the device datasheet.

34.4.3 LPTMR prescaler/glitch filter

The LPTMR prescaler and glitch filter share the same logic which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

34.4.3.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks the CNR. When the LPTMR is enabled, the CNR will increment every 2^2 to 2^{16} prescaler clock cycles. After the LPTMR is enabled, the first increment of the CNR will take an additional one or two prescaler clock cycles due to synchronization logic.

34.4.3.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments the CNR on every clock cycle. When the LPTMR is enabled, the first increment will take an additional one or two prescaler clock cycles due to synchronization logic.

34.4.3.3 Glitch filter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNR. When the LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output will also deassert.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising-edges	The glitch filter output will also assert.

NOTE

The input is only sampled on the rising clock edge.

The CNR will increment each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which the CNR can increment is once every 2^2 to 2^{16} prescaler clock edges. When first enabled, the glitch filter will wait an additional one or two prescaler clock edges due to synchronization logic.

34.4.3.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the CNR every time it asserts. Before the LPTMR is first enabled, the selected input source is forced to be asserted. This prevents the CNR from incrementing if the selected input source is already asserted when the LPTMR is first enabled.

34.4.4 LPTMR compare

When the CNR equals the value of the CMR and increments, the following events occur:

- CSR[TCF] is set.
- LPTMR interrupt is generated if CSR[TIE] is also set.
- LPTMR hardware trigger is generated.
- CNR is reset if CSR[TFC] is clear.

When the LPTMR is enabled, the CMR can be altered only when CSR[TCF] is set. When updating the CMR, the CMR must be written and CSR[TCF] must be cleared before the LPTMR counter has incremented past the new LPTMR compare value.

34.4.5 LPTMR counter

The CNR increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

The CNR is reset when the LPTMR is disabled or if the counter register overflows. If CSR[TFC] is cleared, then the CNR is also reset whenever CSR[TCF] is set.

The CNR continues incrementing when the core is halted in Debug mode when configured for Pulse Counter mode, the CNR will stop incrementing when the core is halted in Debug mode when configured for Time Counter mode.

The CNR cannot be initialized, but can be read at any time. On each read of the CNR, software must first write to the CNR with any value. This will synchronize and register the current value of the CNR into a temporary register. The contents of the temporary register are returned on each read of the CNR.

When reading the CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing, otherwise incorrect data may be returned.

34.4.6 LPTMR hardware trigger

The LPTMR hardware trigger asserts at the same time the CSR[TCF] is set and can be used to trigger hardware events in other peripherals without software intervention. The hardware trigger is always enabled.

When	Then
The CMR is set to 0 with CSR[TFC] clear	The LPTMR hardware trigger will assert on the first compare and does not deassert.
The CMR is set to a nonzero value, or, if CSR[TFC] is set	The LPTMR hardware trigger will assert on each compare and deassert on the following increment of the CNR.

34.4.7 LPTMR interrupt

The LPTMR interrupt is generated whenever CSR[TIE] and CSR[TCF] are set. CSR[TCF] is cleared by disabling the LPTMR or by writing a logic 1 to it.

CSR[TIE] can be altered and CSR[TCF] can be cleared while the LPTMR is enabled.

The LPTMR interrupt is generated asynchronously to the system clock and can be used to generate a wakeup from any low-power mode, including the low-leakage modes, provided the LPTMR is enabled as a wakeup source.

Chapter 35
Real Time Clock (RTC)

The Real Time Clock (RTC) presents a detail description about the features of RTC that operates in one of two modes of operation—chip power-up and chip power-down.

35.1 Introduction

35.1.1 Features

The RTC module features include:

- 32-bit seconds counter with roll-over protection and 32-bit alarm
- 16-bit prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm
- Register write protection
    - Lock register requires POR or software reset to enable write access
- 1 Hz square wave output with optional interrupt

35.1.2 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode.

35.1.3 RTC signal descriptions

Table 35-1. RTC signal descriptions

Signal	Description	I/O
RTC_CLKOUT	1 Hz square-wave output or OSCERCLK	O

35.1.3.1 RTC clock output

The clock to the seconds counter is available on the RTC_CLKOUT signal. It is a 1 Hz square wave output. See RTC_CLKOUT options for details.

35.2 Register definition

All registers must be accessed using 32-bit writes and all register accesses incur three wait states.

Write accesses to any register by non-supervisor mode software, when the supervisor access bit in the control register is clear, will terminate with a bus error.

Read accesses by non-supervisor mode software complete as normal.

Writing to a register protected by the lock register does not generate a bus error, but the write will not complete.

RTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	RTC Time Seconds Register (RTC_TSR)	32	R/W	0000_0000h	35.2.1/660
4003_D004	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	35.2.2/661
4003_D008	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	35.2.3/661
4003_D00C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	35.2.4/662
4003_D010	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	35.2.5/663
4003_D014	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	35.2.6/665
4003_D018	RTC Lock Register (RTC_LR)	32	R/W	0000_00FFh	35.2.7/666
4003_D01C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	35.2.8/667

35.2.1 RTC Time Seconds Register (RTC_TSR)

Address: 4003_D000h base + 0h offset = 4003_D000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TSR field descriptions

Field	Description
TSR	Time Seconds Register When the time counter is enabled, the TSR is read only and increments once a second provided SR[TOF] or SR[TIF] are not set. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TSR can be read or written. Writing to the TSR when the time counter is disabled will clear the SR[TOF] and/or the SR[TIF]. Writing to TSR with zero is supported, but not recommended because TSR will read as zero when SR[TIF] or SR[TOF] are set (indicating the time is invalid).

35.2.2 RTC Time Prescaler Register (RTC_TPR)

Address: 4003_D000h base + 4h offset = 4003_D004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TPR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TPR	Time Prescaler Register When the time counter is enabled, the TPR is read only and increments every 32.768 kHz clock cycle. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TPR can be read or written. The TSR[TSR] increments when bit 14 of the TPR transitions from a logic one to a logic zero.

35.2.3 RTC Time Alarm Register (RTC_TAR)

Address: 4003_D000h base + 8h offset = 4003_D008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TAR field descriptions

Field	Description
TAR	Time Alarm Register When the time counter is enabled, the SR[TAF] is set whenever the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. Writing to the TAR clears the SR[TAF].

35.2.4 RTC Time Compensation Register (RTC_TCR)

Address: 4003_D000h base + Ch offset = 4003_D00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CIC								TCV								CIR								TCR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TCR field descriptions

Field	Description
31–24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second.
23–16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment).
15–8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds. For example, write zero to configure for a compensation interval of one second. This register is double buffered and writes do not take affect until the end of the current compensation interval.
TCR	Time Compensation Register Configures the number of 32.768 kHz clock cycles in each second. This register is double buffered and writes do not take affect until the end of the current compensation interval. 80h Time Prescaler Register overflows every 32896 clock cycles. ... ... FFh Time Prescaler Register overflows every 32769 clock cycles. 00h Time Prescaler Register overflows every 32768 clock cycles. 01h Time Prescaler Register overflows every 32767 clock cycles. .... 7Fh Time Prescaler Register overflows every 32641 clock cycles.

35.2.5 RTC Control Register (RTC_CR)

Address: 4003_D000h base + 10h offset = 4003_D010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	Reserved	SC2P	SC4P	SC8P	SC16P	CLKO	OSCE	0			WPS	UM	SUP	WPE	SWR
W		0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_CR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. It must always be written to 0.
13 SC2P	Oscillator 2pF Load Configure 0 Disable the load. 1 Enable the additional load.
12 SC4P	Oscillator 4pF Load Configure 0 Disable the load. 1 Enable the additional load.

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
11 SC8P	Oscillator 8pF Load Configure 0 Disable the load. 1 Enable the additional load.
10 SC16P	Oscillator 16pF Load Configure 0 Disable the load. 1 Enable the additional load.
9 CLKO	Clock Output 0 The 32 kHz clock is output to other peripherals. 1 The 32 kHz clock is not output to other peripherals.
8 OSCE	Oscillator Enable 0 32.768 kHz oscillator is disabled. 1 32.768 kHz oscillator is enabled. After setting this bit, wait the oscillator startup time before enabling the time counter to allow the 32.768 kHz clock time to stabilize.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 WPS	Wakeup Pin Select The wakeup pin is optional and not available on all devices. 0 Wakeup pin asserts (active low, open drain) if the RTC interrupt asserts or the wakeup pin is turned on. 1 Wakeup pin instead outputs the RTC 32kHz clock, provided the wakeup pin is turned on and the 32kHz clock is output to other peripherals.
3 UM	Update Mode Allows SR[TCE] to be written even when the Status Register is locked. When set, the SR[TCE] can always be written if the SR[TIF] or SR[TOF] are set or if the SR[TCE] is clear. 0 Registers cannot be written when locked. 1 Registers can be written when locked under limited conditions.
2 SUP	Supervisor Access 0 Non-supervisor mode write accesses are not supported and generate a bus error. 1 Non-supervisor mode write accesses are supported.
1 WPE	Wakeup Pin Enable The wakeup pin is optional and not available on all devices. 0 Wakeup pin is disabled. 1 Wakeup pin is enabled and wakeup pin asserts if the RTC interrupt asserts or the wakeup pin is turned on.
0 SWR	Software Reset 0 No effect. 1 Resets all RTC registers except for the SWR bit . The SWR bit is cleared by POR and by software explicitly clearing it.

35.2.6 RTC Status Register (RTC_SR)

Address: 4003_D000h base + 14h offset = 4003_D014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											TCE	0	TAF	TOF	TIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RTC_SR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 TCE	Time Counter Enable When time counter is disabled the TSR register and TPR register are writeable, but do not increment. When time counter is enabled the TSR register and TPR register are not writeable, but increment. 0 Time counter is disabled. 1 Time counter is enabled.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 TAF	Time Alarm Flag Time alarm flag is set when the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. This bit is cleared by writing the TAR register. 0 Time alarm has not occurred. 1 Time alarm has occurred.
1 TOF	Time Overflow Flag Time overflow flag is set when the time counter is enabled and overflows. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. 0 Time overflow has not occurred. 1 Time overflow has occurred and time counter is read as zero.
0 TIF	Time Invalid Flag The time invalid flag is set on POR or software reset. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. 0 Time is valid. 1 Time is invalid and time counter is read as zero.

35.2.7 RTC Lock Register (RTC_LR)

Address: 4003_D000h base + 18h offset = 4003_D018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								1	LRL	SRL	CRL	TCL	1		
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_LR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 LRL	Lock Register Lock After being cleared, this bit can be set only by POR or software reset. 0 Lock Register is locked and writes are ignored. 1 Lock Register is not locked and writes complete as normal.
5 SRL	Status Register Lock After being cleared, this bit can be set only by POR or software reset. 0 Status Register is locked and writes are ignored. 1 Status Register is not locked and writes complete as normal.
4 CRL	Control Register Lock After being cleared, this bit can only be set by POR. 0 Control Register is locked and writes are ignored. 1 Control Register is not locked and writes complete as normal.
3 TCL	Time Compensation Lock After being cleared, this bit can be set only by POR or software reset. 0 Time Compensation Register is locked and writes are ignored. 1 Time Compensation Register is not locked and writes complete as normal.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

35.2.8 RTC Interrupt Enable Register (RTC_IER)

Address: 4003_D000h base + 1Ch offset = 4003_D01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								WPON	Reserved		TSIE	Reserved	TAIE	TOIE	TIIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

RTC_IER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 WPON	Wakeup Pin On The wakeup pin is optional and not available on all devices. Whenever the wakeup pin is enabled and this bit is set, the wakeup pin will assert. 0 No effect. 1 If the wakeup pin is enabled, then the wakeup pin will assert.
6–5 Reserved	This field is reserved.
4 TSIE	Time Seconds Interrupt Enable The seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector. It is generated once a second and requires no software overhead (there is no corresponding status flag to clear). 0 Seconds interrupt is disabled. 1 Seconds interrupt is enabled.
3 Reserved	This field is reserved.
2 TAIE	Time Alarm Interrupt Enable 0 Time alarm flag does not generate an interrupt. 1 Time alarm flag does generate an interrupt.
1 TOIE	Time Overflow Interrupt Enable

Table continues on the next page...

RTC_IER field descriptions (continued)

Field	Description
	0 Time overflow flag does not generate an interrupt. 1 Time overflow flag does generate an interrupt.
0 TIIE	Time Invalid Interrupt Enable 0 Time invalid flag does not generate an interrupt. 1 Time invalid flag does generate an interrupt.

35.3 Functional description

35.3.1 Power, clocking, and reset

The RTC is an always powered block that remains active in all low power modes.

The time counter within the RTC is clocked by a 32.768 kHz clock sourced from an external crystal using the oscillator.

The power-on-reset signal initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers.

35.3.1.1 Oscillator control

The 32.768 kHz crystal oscillator is disabled at POR and must be enabled by software. After enabling the cystal oscillator, wait the oscillator startup time before setting SR[TCE] or using the oscillator clock external to the RTC.

The crystal oscillator includes tunable capacitors that can be configured by software. Do not change the capacitance unless the oscillator is disabled.

35.3.1.2 Software reset

Writing 1 to CR[SWR] forces the equivalent of a POR to the rest of the RTC module. CR[SWR] is not affected by the software reset and must be cleared by software.

35.3.1.3 Supervisor access

When the supervisor access control bit is clear, only supervisor mode software can write to the RTC registers, non-supervisor mode software will generate a bus error. Both supervisor and non-supervisor mode software can always read the RTC registers.

35.3.2 Time counter

The time counter consists of a 32-bit seconds counter that increments once every second and a 16-bit prescaler register that increments once every 32.768 kHz clock cycle.

Reading the time counter (either seconds or prescaler) while it is incrementing may return invalid data due to synchronization of the read data bus. If it is necessary for software to read the prescaler or seconds counter when they could be incrementing, it is recommended that two read accesses are performed and that software verifies that the same data was returned for both reads.

The time seconds register and time prescaler register can be written only when SR[TCE] is clear. Always write to the prescaler register before writing to the seconds register, because the seconds register increments on the falling edge of bit 14 of the prescaler register.

The time prescaler register increments provided SR[TCE] is set, SR[TIF] is clear, SR[TOF] is clear, and the 32.768 kHz clock source is present. After enabling the oscillator, wait the oscillator startup time before setting SR[TCE] to allow time for the oscillator clock output to stabilize.

If the time seconds register overflows then the SR[TOF] will set and the time prescaler register will stop incrementing. Clear SR[TOF] by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TOF] is set.

SR[TIF] is set on POR and software reset and is cleared by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TIF] is set.

35.3.3 Compensation

The compensation logic provides an accurate and wide compensation range and can correct errors as high as 3906 ppm and as low as 0.12 ppm. The compensation factor must be calculated externally to the RTC and supplied by software to the compensation register. The RTC itself does not calculate the amount of compensation that is required, although the 1 Hz clock is output to an external pin in support of external calibration logic.

Crystal compensation can be supported by using firmware and crystal characteristics to determine the compensation amount. Temperature compensation can be supported by firmware that periodically measures the external temperature via ADC and updates the compensation register based on a look-up table that specifies the change in crystal frequency over temperature.

The compensation logic alters the number of 32.768 kHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128. Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used, that is, from once a second to once every 256 seconds.

Updates to the time compensation register will not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

35.3.4 Time alarm

The Time Alarm register (TAR), SR[TAF], and IER[TAIE] allow the RTC to generate an interrupt at a predefined time. The 32-bit TAR is compared with the 32-bit Time Seconds register (TSR) each time it increments. SR[TAF] will set when TAR equals TSR and TSR increments.

SR[TAF] is cleared by writing TAR. This will usually be the next alarm value, although writing a value that is less than TSR, such as 0, will prevent SR[TAF] from setting again. SR[TAF] cannot otherwise be disabled, although the interrupt it generates is enabled or disabled by IER[TAIE].

35.3.5 Update mode

The Update Mode field in the Control register (CR[UM]) configures software write access to the Time Counter Enable (SR[TCE]) field. When CR[UM] is clear, SR[TCE] can be written only when LR[SRL] is set. When CR[UM] is set, SR[TCE] can also be written when SR[TCE] is clear or when SR[TIF] or SR[TOF] are set. This allows the time seconds and prescaler registers to be initialized whenever time is invalidated, while preventing the time seconds and prescaler registers from being changed on the fly. When LR[SRL] is set, CR[UM] has no effect on SR[TCE].

35.3.6 Register lock

The Lock register (LR) can be used to block write accesses to certain registers until the next POR or software reset. Locking the Control register (CR) will disable the software reset. Locking LR will block future updates to LR.

Write accesses to a locked register are ignored and do not generate a bus error.

35.3.7 Interrupt

The RTC interrupt is asserted whenever a status flag and the corresponding interrupt enable bit are both set. It is always asserted on POR, and software reset. The RTC interrupt is enabled at the chip level by enabling the chip-specific RTC clock gate control bit. The RTC interrupt can be used to wakeup the chip from any low-power mode.

The optional RTC seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector that is generated once a second and requires no software overhead (there is no corresponding status flag to clear). It is enabled in the RTC by the time seconds interrupt enable bit and enabled at the chip level by setting the chip-specific RTC clock gate control bit. This interrupt is optional and may not be implemented on all devices.

Chapter 36
Serial Peripheral Interface (SPI)

36.1 Introduction

The serial peripheral interface (SPI) module provides a synchronous serial bus for communication between a chip and an external peripheral device.

36.1.1 Block Diagram

The block diagram of this module is as follows:

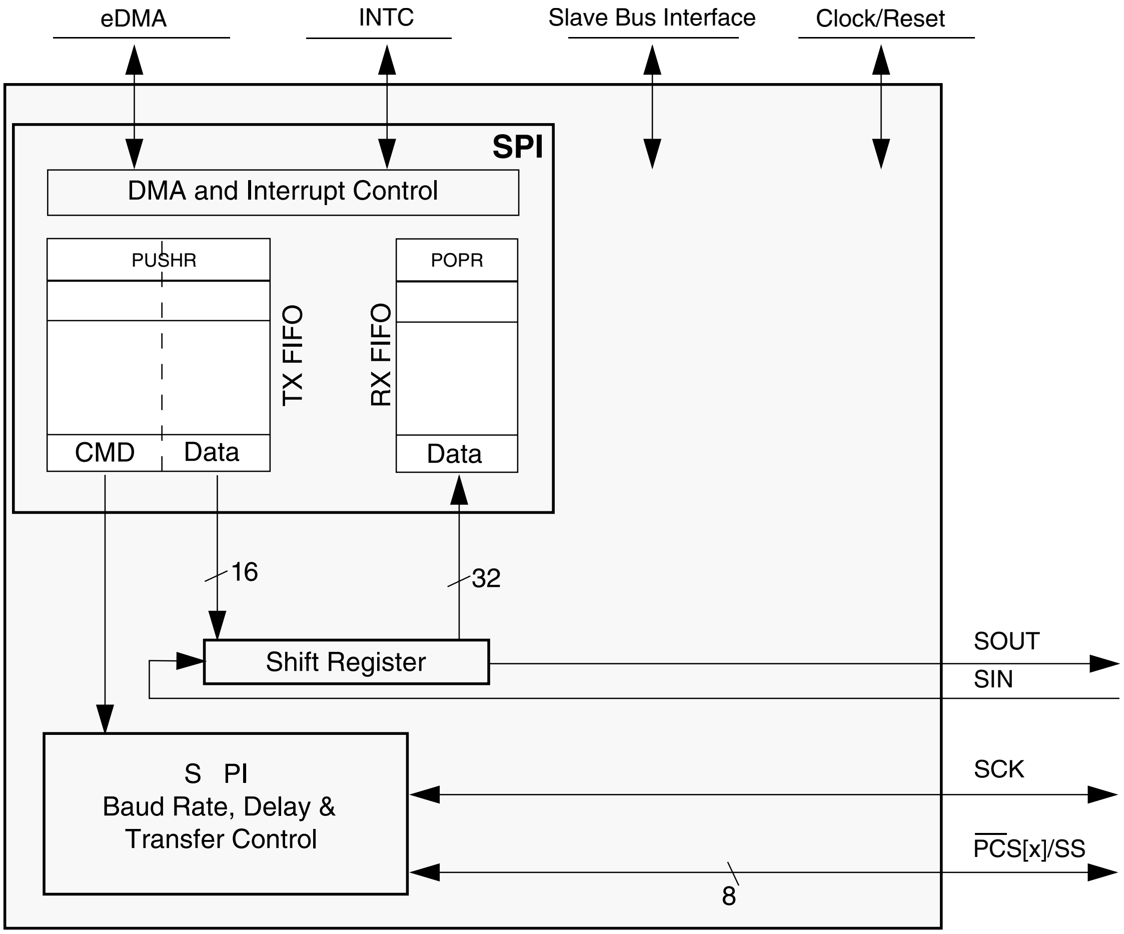


Figure 36-1. SPI Block Diagram

36.1.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of 4 entries
- Buffered receive operation using the receive FIFO (RX FIFO) with depth of 4 entries
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues

- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
    - two transfer attribute registers
    - Serial clock (SCK) with programmable polarity and phase
    - Various programmable delays
    - Programmable serial frame size: 4 to 16 bits
        - SPI frames longer than 16 bits can be supported using the continuous selection format.
    - Continuously held chip select capability
- 4 peripheral chip selects (PCSes), expandable to 16 with external demultiplexer
- Deglitching support for up to 8 peripheral chip selects (PCSes) with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
    - TX FIFO is not full (TFFF)
    - RX FIFO is not empty (RFDF)
- Interrupt conditions:
    - End of Queue reached (EOQF)
    - TX FIFO is not full (TFFF)
    - Transfer of current frame complete (TCF)
    - Attempt to transmit with an empty Transmit FIFO (TFUF)
    - RX FIFO is not empty (RFDF)
    - Frame received while Receive FIFO is full (RFOF)
- Global interrupt request line
- Modified SPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:
    - Support for Stop mode
    - Support for Doze mode

36.1.3 Interface configurations

36.1.3.1 SPI configuration

The Serial Peripheral Interface (SPI) configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic SPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the module. Data transfers between the queues and the module FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, SPI, and external queues in system RAM.

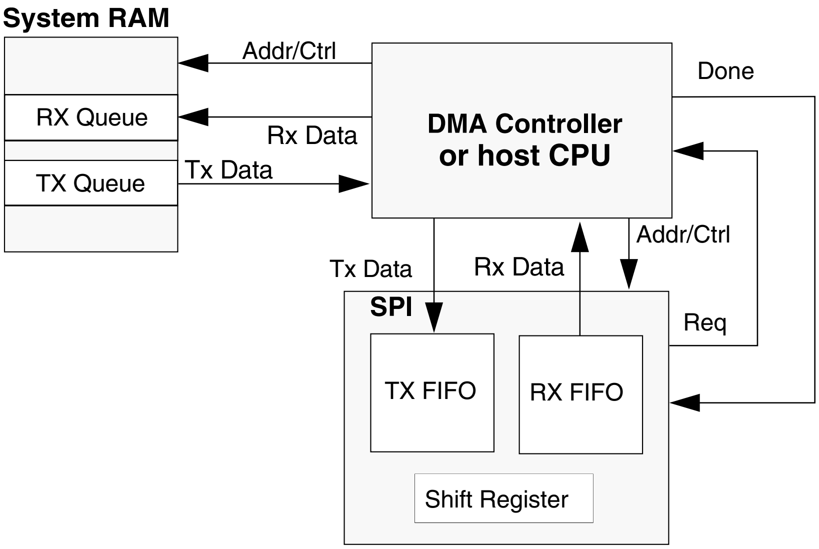


Figure 36-2. SPI with queues and DMA

36.1.4 Modes of Operation

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
    - Master mode

    - Slave mode
    - Module Disable mode
- Chip-specific modes:
    - External Stop mode
    - Debug mode

The module enters module-specific modes when the host writes a module register. The chip-specific modes are controlled by signals external to the module. The chip-specific modes are modes that a chip may enter in parallel to the block-specific modes.

36.1.4.1 Master Mode

Master mode allows the module to initiate and control serial communication. In this mode, these signals are controlled by the module and configured as outputs:

- SCK
- SOUT
- PCS[*x*]

36.1.4.2 Slave Mode

Slave mode allows the module to communicate with SPI bus masters. In this mode, the module responds to externally controlled serial transfers. The SCK signal and the PCS[0]/$\overline{\text{SS}}$ signals are configured as inputs and driven by an SPI bus master.

36.1.4.3 Module Disable Mode

The Module Disable mode can be used for chip power management. The clock to the non-memory mapped logic in the module can be stopped while in the Module Disable mode.

36.1.4.4 External Stop Mode

External Stop mode is used for chip power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the protocol clock to the module may be shut off.

36.1.4.5 Debug Mode

Debug mode is used for system development and debugging. The MCR[FRZ] bit controls module behavior in the Debug mode:

- If the bit is set, the module stops all serial transfers, when the chip is in debug mode.
- If the bit is cleared, the chip debug mode has no effect on the module.

36.2 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

Table 36-1. Module signal descriptions

Signal	Master mode	Slave mode	I/O
PCS0/$\overline{\text{SS}}$	Peripheral Chip Select 0 (O)	Slave Select (I)	I/O
PCS[1:3]	Peripheral Chip Selects 1–3	(Unused)	O
SCK	Serial Clock (O)	Serial Clock (I)	I/O
SIN	Serial Data In	Serial Data In	I
SOUT	Serial Data Out	Serial Data Out	O

36.2.1 PCS0/$\overline{\text{SS}}$—Peripheral Chip Select/Slave Select

Master mode: Peripheral Chip Select 0 (O)—Selects an SPI slave to receive data transmitted from the module.

Slave mode: Slave Select (I)—Selects the module to receive data transmitted from an SPI master.

36.2.2 PCS1–PCS3—Peripheral Chip Selects 1–3

Master mode: Peripheral Chip Selects 1–3 (O)—Select an SPI slave to receive data transmitted by the module.

Slave mode: Unused

36.2.3 SCK—Serial Clock

Master mode: Serial Clock (O)—Supplies a clock signal from the module to SPI slaves.

Slave mode: Serial Clock (I)—Supplies a clock signal to the module from an SPI master.

36.2.4 SIN—Serial Input

Master mode: Serial Input (I)—Receives serial data.

Slave mode: Serial Input (I)—Receives serial data.

36.2.5 SOUT—Serial Output

Master mode: Serial Output (O)—Transmits serial data.

Slave mode: Serial Output (O)—Transmits serial data.

NOTE

Serial Data Out output buffers are controlled through SIU (or SIUL) and cannot be controlled through the module.

36.3 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Write access to the POPR and RXFRn also results in a transfer error.

SPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C000	Module Configuration Register (SPI0_MCR)	32	R/W	0000_4001h	36.3.1/682
4002_C008	Transfer Count Register (SPI0_TCR)	32	R/W	0000_0000h	36.3.2/685

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C00C	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR0)	32	R/W	7800_0000h	36.3.3/686
4002_C00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	36.3.4/690
4002_C010	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR1)	32	R/W	7800_0000h	36.3.3/686
4002_C02C	Status Register (SPI0_SR)	32	R/W	0200_0000h	36.3.5/692
4002_C030	DMA/Interrupt Request Select and Enable Register (SPI0_RSER)	32	R/W	0000_0000h	36.3.6/695
4002_C034	PUSH TX FIFO Register In Master Mode (SPI0_PUSHR)	32	R/W	0000_0000h	36.3.7/697
4002_C034	PUSH TX FIFO Register In Slave Mode (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	36.3.8/699
4002_C038	POP RX FIFO Register (SPI0_POPR)	32	R	0000_0000h	36.3.9/699
4002_C03C	Transmit FIFO Registers (SPI0_TXFR0)	32	R	0000_0000h	36.3.10/ 700
4002_C040	Transmit FIFO Registers (SPI0_TXFR1)	32	R	0000_0000h	36.3.10/ 700
4002_C044	Transmit FIFO Registers (SPI0_TXFR2)	32	R	0000_0000h	36.3.10/ 700
4002_C048	Transmit FIFO Registers (SPI0_TXFR3)	32	R	0000_0000h	36.3.10/ 700
4002_C07C	Receive FIFO Registers (SPI0_RXFR0)	32	R	0000_0000h	36.3.11/ 700
4002_C080	Receive FIFO Registers (SPI0_RXFR1)	32	R	0000_0000h	36.3.11/ 700
4002_C084	Receive FIFO Registers (SPI0_RXFR2)	32	R	0000_0000h	36.3.11/ 700
4002_C088	Receive FIFO Registers (SPI0_RXFR3)	32	R	0000_0000h	36.3.11/ 700
4002_D000	Module Configuration Register (SPI1_MCR)	32	R/W	0000_4001h	36.3.1/682
4002_D008	Transfer Count Register (SPI1_TCR)	32	R/W	0000_0000h	36.3.2/685
4002_D00C	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR0)	32	R/W	7800_0000h	36.3.3/686
4002_D00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI1_CTAR0_SLAVE)	32	R/W	7800_0000h	36.3.4/690
4002_D010	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR1)	32	R/W	7800_0000h	36.3.3/686
4002_D02C	Status Register (SPI1_SR)	32	R/W	0200_0000h	36.3.5/692
4002_D030	DMA/Interrupt Request Select and Enable Register (SPI1_RSER)	32	R/W	0000_0000h	36.3.6/695
4002_D034	PUSH TX FIFO Register In Master Mode (SPI1_PUSHR)	32	R/W	0000_0000h	36.3.7/697
4002_D034	PUSH TX FIFO Register In Slave Mode (SPI1_PUSHR_SLAVE)	32	R/W	0000_0000h	36.3.8/699

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_D038	POP RX FIFO Register (SPI1_POPR)	32	R	0000_0000h	36.3.9/699
4002_D03C	Transmit FIFO Registers (SPI1_TXFR0)	32	R	0000_0000h	36.3.10/ 700
4002_D040	Transmit FIFO Registers (SPI1_TXFR1)	32	R	0000_0000h	36.3.10/ 700
4002_D044	Transmit FIFO Registers (SPI1_TXFR2)	32	R	0000_0000h	36.3.10/ 700
4002_D048	Transmit FIFO Registers (SPI1_TXFR3)	32	R	0000_0000h	36.3.10/ 700
4002_D07C	Receive FIFO Registers (SPI1_RXFR0)	32	R	0000_0000h	36.3.11/ 700
4002_D080	Receive FIFO Registers (SPI1_RXFR1)	32	R	0000_0000h	36.3.11/ 700
4002_D084	Receive FIFO Registers (SPI1_RXFR2)	32	R	0000_0000h	36.3.11/ 700
4002_D088	Receive FIFO Registers (SPI1_RXFR3)	32	R	0000_0000h	36.3.11/ 700

36.3.1 Module Configuration Register (SPIx_MCR)

Contains bits to configure various attributes associated with the module operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the module is in the Running state.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSTR	CONT_SCKE	DCONF		FRZ	MTFE	Reserved	ROOE	Reserved				PCSIS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOZE	MDIS	DIS_TXF	DIS_RXF	0	0	SMPL_PT		0					Reserved	Reserved	HALT
W					CLR_TXF	CLR_RXF										
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SPIx_MCR field descriptions

Field	Description
31 MSTR	Master/Slave Mode Select Enables either Master mode (if supported) or Slave mode (if supported) operation. 0 Enables Slave mode 1 Enables Master mode
30 CONT_SCKE	Continuous SCK Enable Enables the Serial Communication Clock (SCK) to run continuously. 0 Continuous SCK disabled. 1 Continuous SCK enabled.
29–28 DCONF	SPI Configuration. Selects among the different configurations of the module. 00 SPI 01 Reserved 10 Reserved 11 Reserved
27 FRZ	Freeze Enables transfers to be stopped on the next frame boundary when the device enters Debug mode. 0 Do not halt serial transfers in Debug mode. 1 Halt serial transfers in Debug mode.
26 MTFE	Modified Transfer Format Enable Enables a modified transfer format to be used. 0 Modified SPI transfer format disabled. 1 Modified SPI transfer format enabled.
25 Reserved	This field is reserved.
24 ROOE	Receive FIFO Overflow Overwrite Enable In the RX FIFO overflow condition, configures the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register. 0 Incoming data is ignored. 1 Incoming data is shifted into the shift register.
23–20 Reserved	Always write the reset value to this field. This field is reserved.
19–16 PCSIS	Peripheral Chip Select x Inactive State Determines the inactive state of PCSx. Refer to the chip-specific SPI information for the number of PCS signals used in this chip. **NOTE:** The effect of this bit only takes place when module is enabled. Ensure that this bit is configured correctly before enabling the DSPI interface.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	0 The inactive state of PCSx is low. 1 The inactive state of PCSx is high.
15 DOZE	Doze Enable Provides support for an externally controlled Doze mode power-saving mechanism. 0 Doze mode has no effect on the module. 1 Doze mode disables the module.
14 MDIS	Module Disable Allows the clock to be stopped to the non-memory mapped logic in the module effectively putting it in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default reset value of 1. When the module is used in Slave Mode, it is recommended to leave this bit 0, because a slave doesn't have control over master transactions. 0 Enables the module clocks. 1 Allows external logic to disable the module clocks.
13 DIS_TXF	Disable Transmit FIFO When the TX FIFO is disabled, the transmit part of the module operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared. 0 TX FIFO is enabled. 1 TX FIFO is disabled.
12 DIS_RXF	Disable Receive FIFO When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered SPI. This bit can only be written when the MDIS bit is cleared. 0 RX FIFO is enabled. 1 RX FIFO is disabled.
11 CLR_TXF	Clear TX FIFO Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero. 0 Do not clear the TX FIFO counter. 1 Clear the TX FIFO counter.
10 CLR_RXF	CLR_RXF Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero. 0 Do not clear the RX FIFO counter. 1 Clear the RX FIFO counter.
9–8 SMPL_PT	Sample Point Controls when the module master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0. 00 0 protocol clock cycles between SCK edge and SIN sample 01 1 protocol clock cycle between SCK edge and SIN sample

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	10 2 protocol clock cycles between SCK edge and SIN sample 11 Reserved
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 Reserved	This field is reserved.
0 HALT	Halt The HALT bit starts and stops frame transfers. See Start and Stop of Module transfers 0 Start transfers. 1 Stop transfers.

36.3.2 Transfer Count Register (SPIx_TCR)

TCR contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the module is in the Running state.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPI_TCNT																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TCR field descriptions

Field	Description
31–16 SPI_TCNT	SPI Transfer Counter Counts the number of SPI transfers the module makes. The SPI_TCNT field increments every time the last bit of an SPI frame is transmitted. A value written to SPI_TCNT presets the counter to that value. SPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

36.3.3 Clock and Transfer Attributes Register (In Master Mode) (SPIx_CTAR*n*)

CTAR registers are used to define different transfer attributes. Do not write to the CTAR registers while the module is in the Running state.

In Master mode, the CTAR registers define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In slave mode, a subset of the bitfields in CTAR0 are used to set the slave transfer attributes.

When the module is configured as an SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR registers is used. When the module is configured as an SPI bus slave, it uses the CTAR0 register.

Address: Base address + Ch offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DBR	FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSSCK				ASC				DT				BR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n* field descriptions

Field	Description
31 DBR	Double Baud Rate Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the Serial Communications Clock (SCK). When the DBR bit is set, the duty cycle of the Serial Communications Clock (SCK) depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate.

Table 36-2. SPI SCK Duty Cycle

DBR	CPHA	PBR	SCK Duty Cycle
0	any	any	50/50
1	0	00	50/50
1	0	01	33/66
1	0	10	40/60

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Table 36-2. SPI SCK Duty Cycle (continued)

DBR	CPHA	PBR	SCK Duty Cycle
1	0	11	43/57
1	1	00	50/50
1	1	01	66/33
1	1	10	60/40
1	1	11	57/43

Field	Description
	0 The baud rate is computed normally with a 50/50 duty cycle. 1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.
30–27 FMSZ	Frame Size The number of bits transferred per frame is equal to the FMSZ value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4.
26 CPOL	Clock Polarity Selects the inactive state of the Serial Communications Clock (SCK). This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the module can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge. **NOTE:** In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranted. 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24 LSBFE	LSB First Specifies whether the LSB or MSB of the frame is transferred first. 0 Data is transferred MSB first. 1 Data is transferred LSB first.
23–22 PCSSCK	PCS to SCK Delay Prescaler Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. Refer PCS to SCK Delay (t_{CSC}) for more details.

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	00 PCS to SCK Prescaler value is 1. 01 PCS to SCK Prescaler value is 3. 10 PCS to SCK Prescaler value is 5. 11 PCS to SCK Prescaler value is 7.
21–20 PASC	After SCK Delay Prescaler Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay. Refer After SCK Delay (t_{ASC}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
19–18 PDT	Delay after Transfer Prescaler Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is only used in master mode. See the DT field description for details on how to compute the Delay after Transfer. Refer Delay after Transfer (t_{DT}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
17–16 PBR	Baud Rate Prescaler Selects the prescaler value for the baud rate. This field is used only in master mode. The baud rate is the frequency of the SCK. The protocol clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate. 00 Baud Rate Prescaler value is 2. 01 Baud Rate Prescaler value is 3. 10 Baud Rate Prescaler value is 5. 11 Baud Rate Prescaler value is 7.
15–12 CSSCK	PCS to SCK Delay Scaler Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the protocol clock period, and it is computed according to the following equation: $t_{CSC} = (1/f_P) \times PCSSCK \times CSSCK$. The following table lists the delay scaler values.

Table 36-3. Delay Scaler Encoding

Field Value	Delay Scaler Value
0000	2
0001	4
0010	8

Table continues on the next page...

SPIx_CTARn field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<b>Table 36-3. Delay Scaler Encoding (continued)</b>
<table>
<tr><th>Field Value</th><th>Delay Scaler Value</th></tr>
<tr><td>0011</td><td>16</td></tr>
<tr><td>0100</td><td>32</td></tr>
<tr><td>0101</td><td>64</td></tr>
<tr><td>0110</td><td>128</td></tr>
<tr><td>0111</td><td>256</td></tr>
<tr><td>1000</td><td>512</td></tr>
<tr><td>1001</td><td>1024</td></tr>
<tr><td>1010</td><td>2048</td></tr>
<tr><td>1011</td><td>4096</td></tr>
<tr><td>1100</td><td>8192</td></tr>
<tr><td>1101</td><td>16384</td></tr>
<tr><td>1110</td><td>32768</td></tr>
<tr><td>1111</td><td>65536</td></tr>
</table>
Refer PCS to SCK Delay (t_{CSC}) for more details.
</td></tr>
<tr><td>11–8
ASC</td><td>After SCK Delay Scaler

Selects the scaler value for the After SCK Delay. This field is used only in master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:

t_{ASC} = (1/f_P) x PASC x ASC

See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. Refer After SCK Delay (t_{ASC}) for more details.</td></tr>
<tr><td>7–4
DT</td><td>Delay After Transfer Scaler

Selects the Delay after Transfer Scaler. This field is used only in master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame.

In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period, The Delay after Transfer is a multiple of the protocol clock period, and it is computed according to the following equation:

t_{DT} = (1/f_P) x PDT x DT

See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values.</td></tr>
<tr><td>BR</td><td>Baud Rate Scaler

Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled protocol clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation:

SCK baud rate = (f_P /PBR) x [(1+DBR)/BR]

The following table lists the baud rate scaler values.</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>

Table 36-4. Baud Rate Scaler

<table>
<tr><th>CTARn[BR]</th><th>Baud Rate Scaler Value</th></tr>
<tr><td>0000</td><td>2</td></tr>
<tr><td>0001</td><td>4</td></tr>
<tr><td>0010</td><td>6</td></tr>
<tr><td>0011</td><td>8</td></tr>
<tr><td>0100</td><td>16</td></tr>
<tr><td>0101</td><td>32</td></tr>
<tr><td>0110</td><td>64</td></tr>
<tr><td>0111</td><td>128</td></tr>
<tr><td>1000</td><td>256</td></tr>
<tr><td>1001</td><td>512</td></tr>
<tr><td>1010</td><td>1024</td></tr>
<tr><td>1011</td><td>2048</td></tr>
<tr><td>1100</td><td>4096</td></tr>
<tr><td>1101</td><td>8192</td></tr>
<tr><td>1110</td><td>16384</td></tr>
<tr><td>1111</td><td>32768</td></tr>
</table>

</td></tr>
</table>

36.3.4 Clock and Transfer Attributes Register (In Slave Mode) (SPIx_CTAR*n*_SLAVE)

When the module is configured as an SPI bus slave, the CTAR0 register is used.

Address: Base address + Ch offset + (0d × i), where i=0d to 0d

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2" colspan="4">FMSZ</td><td rowspan="2">CPOL</td><td rowspan="2">CPHA</td><td colspan="2">0</td><td rowspan="2">Reserved</td><td rowspan="2" colspan="6">Reserved</td></tr>
<tr><td>W</td><td colspan="2"></td></tr>
<tr><td>Reset</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td rowspan="2" colspan="16">Reserved</td></tr>
<tr><td>W</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

SPIx_CTAR*n*_SLAVE field descriptions

Field	Description
31 Reserved	Always write the reset value to this field. This field is reserved.
30–27 FMSZ	Frame Size The number of bits transfered per frame is equal to the FMSZ field value plus 1. Note that the minimum valid value of frame size is 4.
26 CPOL	Clock Polarity Selects the inactive state of the Serial Communications Clock (SCK). **NOTE:** In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranted. 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved.
Reserved	This field is reserved.

36.3.5 Status Register (SPIx_SR)

SR contains status and flag bits. The bits reflect the status of the module and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0	0	0	0	0	RFOF	0	RFDF	0
W	w1c	w1c		w1c	w1c		w1c						w1c		w1c	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_SR field descriptions

Field	Description
31 TCF	Transfer Complete Flag Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it. 0 Transfer not complete. 1 Transfer complete.
30 TXRXS	TX and RX Status Reflects the run status of the module. 0 Transmit and receive operations are disabled (The module is in Stopped state). 1 Transmit and receive operations are enabled (The module is in Running state).
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
28 EOQF	End of Queue Flag Indicates that the last entry in a queue has been transmitted when the module is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared. 0 EOQ is not set in the executing command. 1 EOQ is set in the executing SPI command.
27 TFUF	Transmit FIFO Underflow Flag Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for SPI blocks operating in Slave mode and SPI configuration. TFUF is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it. 0 No TX FIFO underflow. 1 TX FIFO underflow has occurred.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 TFFF	Transmit FIFO Fill Flag Provides a method for the module to request more entries to be added to the TX FIFO. The TFFF bit is set while the TX FIFO is not full. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request. **NOTE:** The reset value of this bit is 0 when the module is disabled,(MCR[MDIS]=1). 0 TX FIFO is full. 1 TX FIFO is not full.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 RFOF	Receive FIFO Overflow Flag Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it. 0 No Rx FIFO overflow. 1 Rx FIFO overflow has occurred.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 RFDF	Receive FIFO Drain Flag

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
	Provides a method for the module to request that entries be removed from the RX FIFO. The bit is set while the RX FIFO is not empty. The RFDF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller when the RX FIFO is empty. 0 RX FIFO is empty. 1 RX FIFO is not empty.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TXCTR	TX FIFO Counter Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHR is written. The TXCTR is decremented every time an SPI command is executed and the SPI data is transferred to the shift register.
11–8 TXNXTPTR	Transmit Next Pointer Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register.
7–4 RXCTR	RX FIFO Counter Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO.
POPNXTPTR	Pop Next Pointer Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read.

36.3.6 DMA/Interrupt Request Select and Enable Register (SPIx_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the module is in the Running state.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCF_RE	Reserved	Reserved	EOQF_RE	TFUF_RE	Reserved	TFFF_RE	TFFF_DIRS	Reserved	Reserved	Reserved	Reserved	RFOF_RE	Reserved	RFDF_RE	RFDF_DIRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	0													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RSER field descriptions

Field	Description
31 TCF_RE	Transmission Complete Request Enable Enables TCF flag in the SR to generate an interrupt request. 0 TCF interrupt requests are disabled. 1 TCF interrupt requests are enabled.
30 Reserved	Always write the reset value to this field. This field is reserved.
29 Reserved	Always write the reset value to this field. This field is reserved.
28 EOQF_RE	Finished Request Enable Enables the EOQF flag in the SR to generate an interrupt request. 0 EOQF interrupt requests are disabled. 1 EOQF interrupt requests are enabled.
27 TFUF_RE	Transmit FIFO Underflow Request Enable Enables the TFUF flag in the SR to generate an interrupt request.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
	0 TFUF interrupt requests are disabled. 1 TFUF interrupt requests are enabled.
26 Reserved	Always write the reset value to this field. This field is reserved.
25 TFFF_RE	Transmit FIFO Fill Request Enable Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request. 0 TFFF interrupts or DMA requests are disabled. 1 TFFF interrupts or DMA requests are enabled.
24 TFFF_DIRS	Transmit FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request. 0 TFFF flag generates interrupt requests. 1 TFFF flag generates DMA requests.
23 Reserved	Always write the reset value to this field. This field is reserved.
22 Reserved	Always write the reset value to this field. This field is reserved.
21 Reserved	Always write the reset value to this field. This field is reserved.
20 Reserved	Always write the reset value to this field. This field is reserved.
19 RFOF_RE	Receive FIFO Overflow Request Enable Enables the RFOF flag in the SR to generate an interrupt request. 0 RFOF interrupt requests are disabled. 1 RFOF interrupt requests are enabled.
18 Reserved	Always write the reset value to this field. This field is reserved.
17 RFDF_RE	Receive FIFO Drain Request Enable Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 RFDF interrupt or DMA requests are disabled. 1 RFDF interrupt or DMA requests are enabled.
16 RFDF_DIRS	Receive FIFO Drain DMA or Interrupt Request Select

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
	Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 Interrupt request. 1 DMA request.
15 Reserved	Always write the reset value to this field. This field is reserved.
14 Reserved	Always write the reset value to this field. This field is reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

36.3.7 PUSH TX FIFO Register In Master Mode (SPIx_PUSHR)

Specifies data to be transferred to the TX FIFO. An 8- or 16-bit write access transfers all 32 bits to the TX FIFO. In Master mode, the register transfers 16 bits of data and 16 bits of command information. A read access of PUSHR returns the topmost TX FIFO entry.

When the module is disabled, writing to this register does not update the FIFO. Therefore, any reads performed while the module is disabled return the last PUSHR write performed while the module was still enabled.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CONT	CTAS			EOQ	CTCNT	Reserved		Reserved				PCS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR field descriptions

Field	Description
31 CONT	Continuous Peripheral Chip Select Enable Selects a continuous selection format. The bit is used in SPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers. 0 Return PCSn signals to their inactive state between transfers. 1 Keep PCSn signals asserted between transfers.
30–28 CTAS	Clock and Transfer Attributes Select Selects which CTAR to use in master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chip specific section for details to determine how many CTARs this device has. You should not program a value in this field for a register that is not present. 000 CTAR0 001 CTAR1 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
27 EOQ	End Of Queue Host software uses this bit to signal to the module that the current SPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set. 0 The SPI data is not the last data to transfer. 1 The SPI data is the last data to transfer.
26 CTCNT	Clear Transfer Counter Clears the TCNT field in the TCR register. The TCNT field is cleared before the module starts transmitting the current SPI frame. 0 Do not clear the TCR[TCNT] field. 1 Clear the TCR[TCNT] field.
25–24 Reserved	Always write the reset value to this field. This field is reserved.
23–20 Reserved	Always write the reset value to this field. This field is reserved.
19–16 PCS	Select which PCS signals are to be asserted for the transfer. Refer to the chip-specific SPI information for the number of PCS signals used in this chip. 0 Negate the PCS[x] signal 1 Assert the PCS[x] signal.
TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

36.3.8 PUSH TX FIFO Register In Slave Mode (SPIx_PUSHR_SLAVE)

Specifies data to be transferred to the TX FIFO in slave mode. An 8- or 16-bit write access to PUSHR transfers the 16-bit TXDATA field to the TX FIFO.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR_SLAVE field descriptions

Field	Description
31–16 Reserved	This field is reserved.
TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

36.3.9 POP RX FIFO Register (SPIx_POPR)

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_POPR field descriptions

Field	Description
RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points.

36.3.10 Transmit FIFO Registers (SPIx_TXFR*n*)

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO.

Address: Base address + 3Ch offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCMD_TXDATA																TXDATA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TXFR*n* field descriptions

Field	Description
31–16 TXCMD_ TXDATA	Transmit Command or Transmit Data In Master mode the TXCMD field contains the command that sets the transfer attributes for the SPI data. In Slave mode, this field is reserved.
TXDATA	Transmit Data Contains the SPI data to be shifted out.

36.3.11 Receive FIFO Registers (SPIx_RXFR*n*)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFR registers are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO.

Address: Base address + 7Ch offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RXFR*n* field descriptions

Field	Description
RXDATA	Receive Data Contains the received SPI data.

SPIx_RXFR*n* field descriptions (continued)

Field	Description

36.4 Functional description

The module supports full-duplex, synchronous serial communications between chips and peripheral devices. The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes.

The module has the following configuration

- The SPI Configuration in which the module operates as a basic SPI or a queued SPI.

The DCONF field in the Module Configuration Register (MCR) determines the module Configuration. SPI configuration is selected when DCONF within SPIx_MCR is 0b00.

The CTARn registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the SPI command.

See Clock and Transfer Attributes Register (In Master Mode) (SPI_CTAR*n*) for information on the fields of CTAR registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the Transfer Control Flag(TCF) bit in the Shift Register(SR) is set to indicate a completed frame transfer.

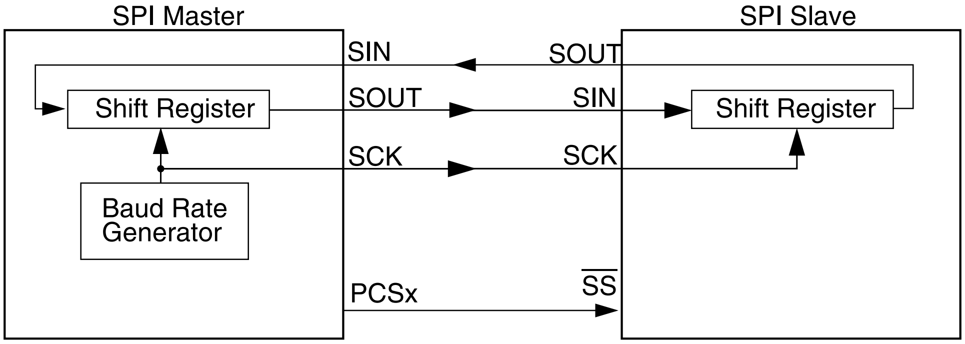


Figure 36-3. Serial protocol overview

Generally, more than one slave device can be connected to the module master. 4 Peripheral Chip Select (PCS) signals of the module masters can be used to select which of the slaves to communicate with. Refer to the chip specific section for details on the number of PCS signals used in this chip.

The SPI configuration shares transfer protocol and timing properties which are described independently of the configuration in Transfer formats. The transfer rate and delay settings are described in Module baud rate and clock delay generation.

36.4.1 Start and Stop of module transfers

The module has two operating states: Stopped and Running. Both the states are independent of it's configuration. The default state of the module is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the module without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of module. The bit is set if the module is in Running state.

The module starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear
- Chip is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The module stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- Chip in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

36.4.2 Serial Peripheral Interface (SPI) configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in SPI configuration when the DCONF field in the MCR is 0b00. The SPI frames can be 32 bits long. The host CPU or a DMA controller transfers the SPI data from the external to the module RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the module. The operation of FIFO buffers is described in the following sections:

- Transmit First In First Out (TX FIFO) buffering mechanism
- Transmit First In First Out (TX FIFO) buffering mechanism
- Receive First In First Out (RX FIFO) buffering mechanism

The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two block-specific modes—Master mode and Slave mode.In Master mode the module initiates and controls the transfer according to the fields of the executing SPI Command. In Slave mode, the module responds only to transfers initiated by a bus master external to it and the SPI command field space is reserved.

36.4.2.1 Master mode

In SPI Master mode, the module initiates the serial transfers by controlling the SCK and the PCS signals. The executing SPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert. The command field also contains various bits that help with queue management and transfer protocol. See PUSH TX FIFO Register In Master Mode (SPI_PUSHR) for details on the SPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the Serial Out (SOUT) pin. In SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis.

36.4.2.2 Slave mode

In SPI Slave mode the module responds to transfers initiated by an SPI bus master. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an SPI master. The SPI Slave mode transfer attributes are set in the CTAR0. The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

36.4.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO or RX FIFO. The module operates as a double-buffered simplified SPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately. Setting the MCR[DIS_TXF] bit disables the TX FIFO, and setting the MCR[DIS_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO is disabled:

- SR[TFFF], SR[TFUF] and SR[TXCTR] behave as if there is a one-entry FIFO
- The contents of TXFRs, SR[TXNXTPTR] are undefined

Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

36.4.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. The TX FIFO holds 4 words, each consisting of SPI data. The number of entries in the TX FIFO is device-specific. SPI data is added to the TX FIFO by writing to the Data Field of module PUSH FIFO Register (PUSHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the module Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to PUSHR[TXDATA] or SPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXNXTPTR field is incremented every time SPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

36.4.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO by writing to the PUSHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See Transmit FIFO Fill Interrupt or DMA Request for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

36.4.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting SPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR_TXF bit in MCR.

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See Transmit FIFO Underflow Interrupt Request for details.

36.4.2.5 Receive First In First Out (RX FIFO) buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds 4 received SPI data frames. The number of entries in the RX FIFO is device-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. SPI data are removed (popped) from the RX FIFO by reading the module POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the module's Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or SPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received SPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

36.4.2.5.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the shift register. While the RX FIFO is not full, SPI frames from the shift register are transferred to the RX FIFO. Every time an SPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

36.4.2.5.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the module POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

36.4.3 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the system clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

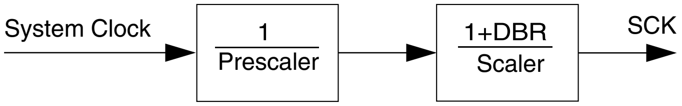


Figure 36-4. Communications clock prescalers and scalers

36.4.3.1 Baud rate generator

The baud rate is the frequency of the SCK. The protocol clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 36-5. Baud rate computation example

f_P	PBR	Prescaler	BR	Scaler	DBR	Baud rate
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

NOTE

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

36.4.3.2 PCS to SCK Delay (t_{CSC})

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See Figure 36-5 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

Table 36-6. PCS to SCK delay computation example

f_{SYS}	PCSSCK	Prescaler	CSSCK	Scaler	PCS to SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 µs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

36.4.3.3 After SCK Delay (t_{ASC})

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See Figure 36-5 and Figure 36-6 for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR*x* registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

Table 36-7. After SCK Delay computation example

f_P	PASC	Prescaler	ASC	Scaler	After SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

36.4.3.4 Delay after Transfer (t_{DT})

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 36-5 for an illustration of the Delay after Transfer. The PDT and DT fields in the CTAR*x* registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

Table 36-8. Delay after Transfer computation example

f_P	PDT	Prescaler	DT	Scaler	Delay after Transfer
100 MHz	0b01	3	0b1110	32768	0.98 ms

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the t_{DT} delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode, the delay is fixed at 1 SCK period.

36.4.4 Transfer formats

The SPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used.

The module supports four different transfer formats:

- Classic SPI with CPHA=0
- Classic SPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The module can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic SPI Format and Modified Transfer Format.

In the interface configurations, the module provides the option of keeping the PCS signals asserted between frames. See Continuous Selection Format for details.

36.4.4.1 Classic SPI Transfer Format (CPHA = 0)

The transfer format shown in following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

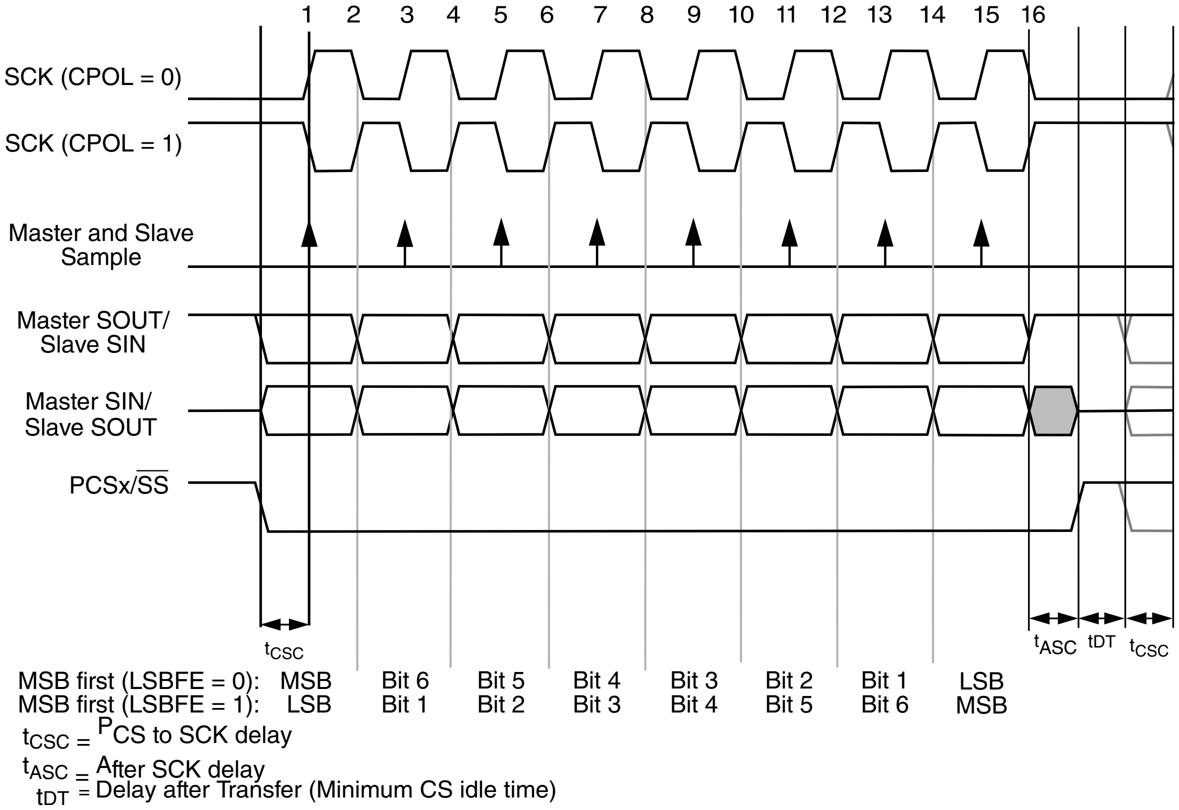

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SCK (CPOL = 0)
SCK (CPOL = 1)
Master and Slave Sample
Master SOUT/ Slave SIN
Master SIN/ Slave SOUT
PCSx/$\overline{SS}$
t_{CSC}
t_{ASC} t_{DT} t_{CSC}
MSB first (LSBFE = 0): MSB Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 LSB
MSB first (LSBFE = 1): LSB Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 MSB
t_{CSC} = PCS to SCK delay
t_{ASC} = After SCK delay
t_{DT} = Delay after Transfer (Minimum CS idle time)

Figure 36-5. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the t_{CSC} delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signals. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

36.4.4.2 Classic SPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges.

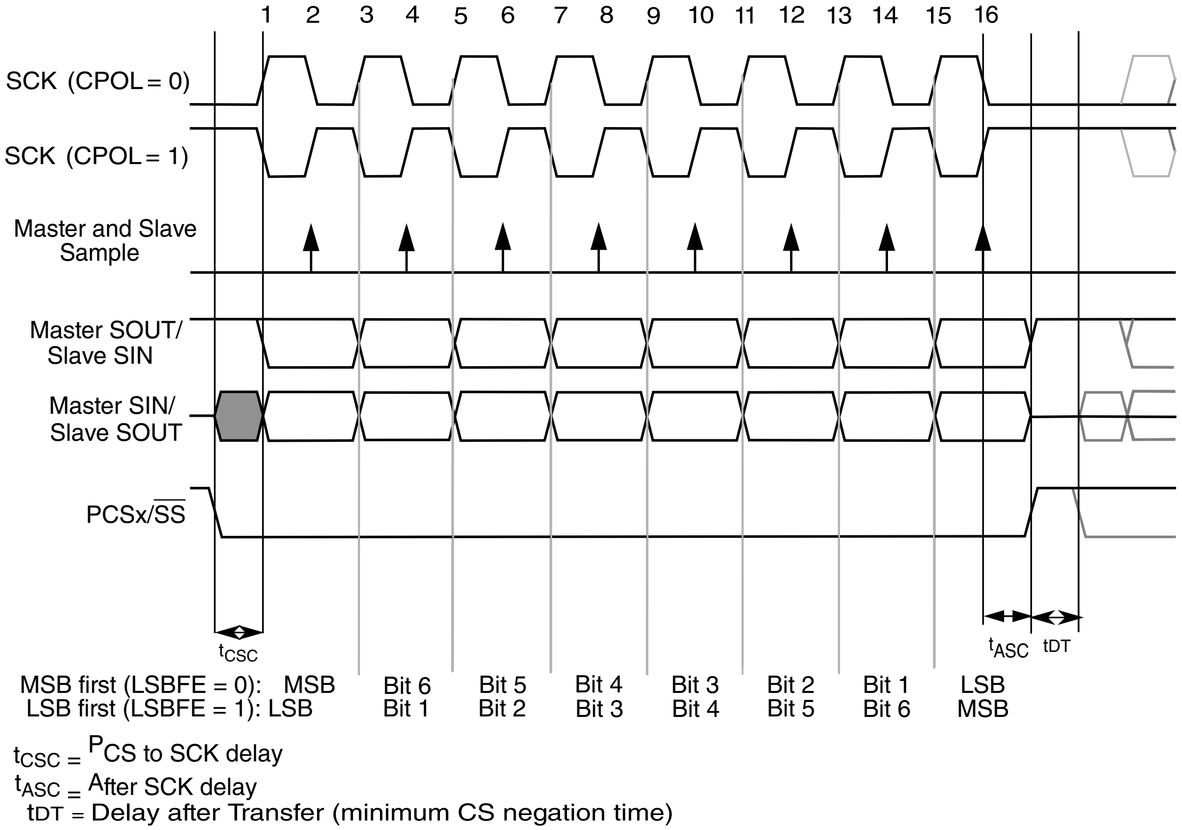

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SCK (CPOL = 0)
SCK (CPOL = 1)
Master and Slave Sample
Master SOUT/ Slave SIN
Master SIN/ Slave SOUT
PCSx/$\overline{SS}$
t_{CSC}
t_{ASC}
tDT
MSB first (LSBFE = 0): MSB Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 LSB
LSB first (LSBFE = 1): LSB Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 MSB
t_{CSC} = PCS to SCK delay
t_{ASC} = After SCK delay
tDT = Delay after Transfer (minimum CS negation time)

Figure 36-6. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)

The master initiates the transfer by asserting the PCS signal to the slave. After the t_{CSC} delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin. The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signal. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

36.4.4.3 Modified SPI Transfer Format (MTFE = 1, CPHA = 0)

In this Modified Transfer Format both the master and the slave sample later in the SCK period than in Classic SPI mode to allow the logic to tolerate more delays in device pads and board traces. These delays become a more significant fraction of the SCK period as the SCK period decreases with increasing baud rates.

The master and the slave place data on the SOUT pins at the assertion of the PCS signal. After the PCS to SCK delay has elapsed the first SCK edge is generated. The slave samples the master SOUT signal on every odd numbered SCK edge. The DSPI in the slave mode when the MTFE bit is set also places new data on the slave SOUT on every odd numbered clock edge. Regular external slave, configured with CPHA=0 format drives its SOUT output at every even numbered SCK clock edge.

The DSPI master places its second data bit on the SOUT line one protocol clock after odd numbered SCK edge if the protocol clock frequency to SCK frequency ratio is higher than three. If this ratio is below four the master changes SOUT at odd numbered SCK edge. The point where the master samples the SIN is selected by the DSPI_MCR[SMPL_PT] field. The master sample point can be delayed by one or two protocol clock cycles. The SMPL_PT field should be set to 0 if the protocol to SCK frequency ratio is less than 4. However if this ratio is less than 4, the actual sample point is delayed by one protocol clock cycle automatically by the design.

The following timing diagrams illustrate the DSPI operation with MTFE=1. Timing delays shown are:

- T_{csc} - PCS to SCK assertion delay
- T_{acs} - After SCK PCS negation delay
- T_{su_ms} - master SIN setup time
- T_{hd_ms} - master SIN hold time
- T_{vd_sl} - slave data output valid time, time between slave data output SCK driving edge and data becomes valid.
- T_{su_sl} - data setup time on slave data input
- T_{hd_sl} - data hold time on slave data input
- T_{sys} - protocol clock period.

The following figure shows the modified transfer format for CPHA = 0 and Fsys/Fsck = 4. Only the condition where CPOL = 0 is illustrated. Solid triangles show the data sampling clock edges. The two possible slave behavior are shown.

- Signal, marked "SOUT of Ext Slave", presents regular SPI slave serial output.
- Signal, marked "SOUT of DSPI Slave", presents DSPI in the slave mode with MTFE bit set.

Other MTFE = 1 diagrams show DSPI SIN input as being driven by a regular external SPI slave, configured according DSPI master CPHA programming.

Note

In the following diagrams, f_{sys} represents the protocol clock frequency from which the Baud frequency f_{sck} is derived.

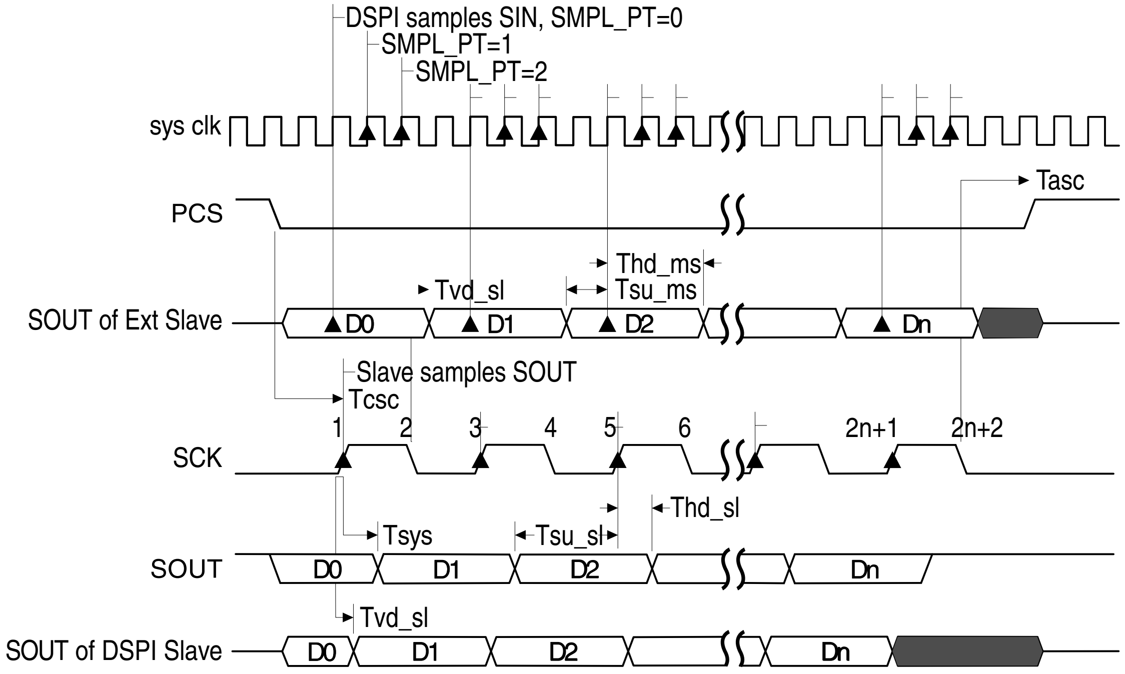


Figure 36-7. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/4$)

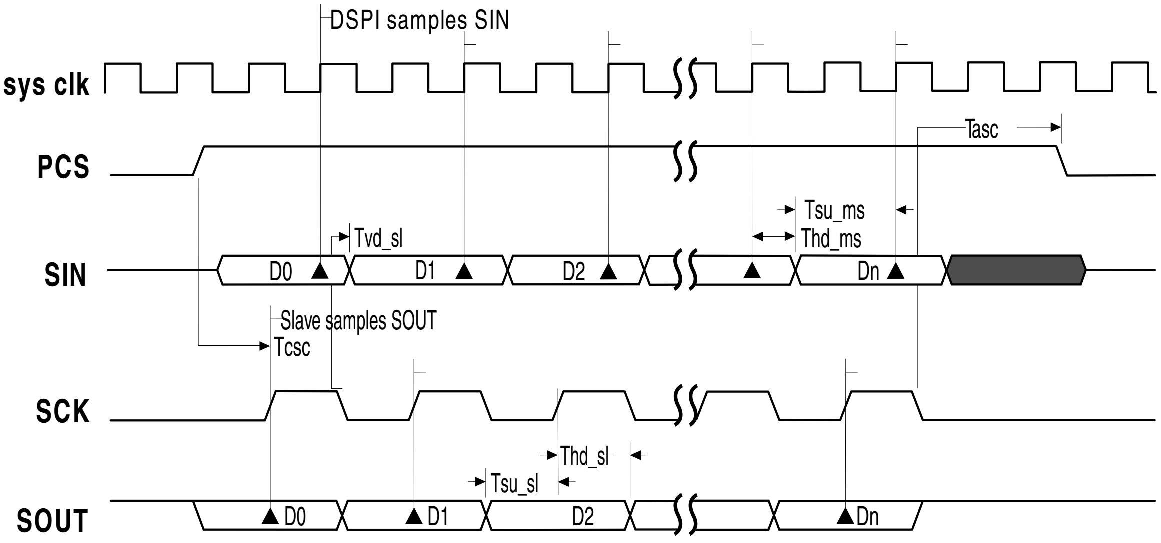


Figure 36-8. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/2$)

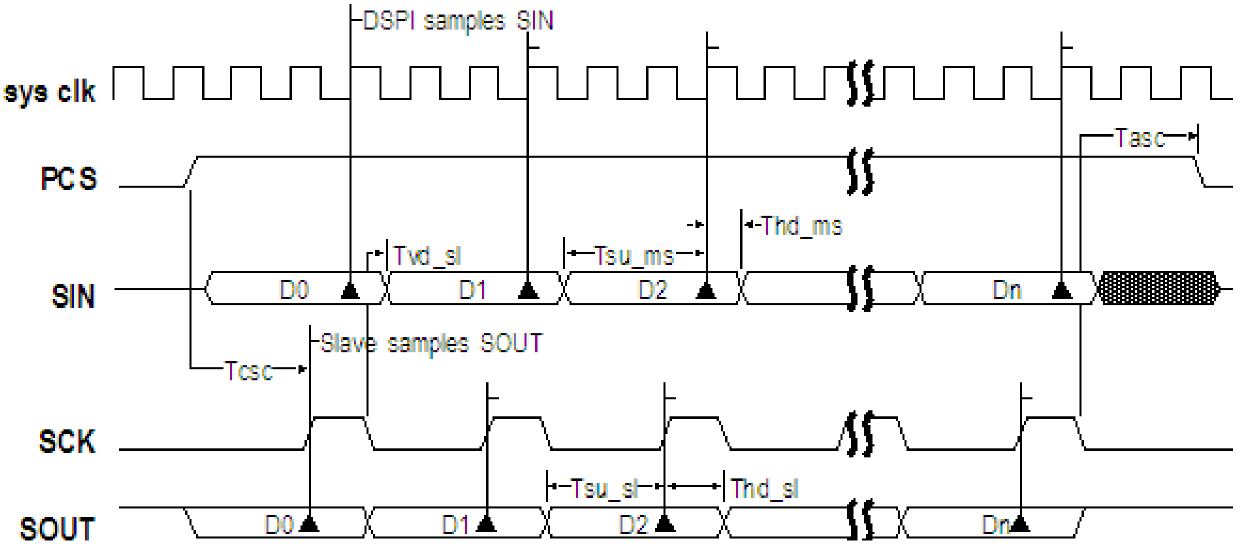


Figure 36-9. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/3$)

36.4.4.4 Modified SPI Transfer Format (MTFE = 1, CPHA = 1)

The following figures show the Modified Transfer Format for CPHA = 1. Only the condition, where CPOL = 0 is shown. At the start of a transfer the DSPI asserts the PCS signal to the slave device. After the PCS to SCK delay has elapsed the master and the slave put data on their SOUT pins at the first edge of SCK . The slave samples the master SOUT signal on the even numbered edges of SCK. The master samples the slave SOUT signal on the odd numbered SCK edges starting with the third SCK edge. The slave samples the last bit on the last edge of the SCK. The master samples the last slave SOUT bit one half SCK cycle after the last edge of SCK. No clock edge will be visible on the master SCK pin during the sampling of the last bit. **The SCK to PCS delay and the After SCK delay must be greater or equal to half of the SCK period.**

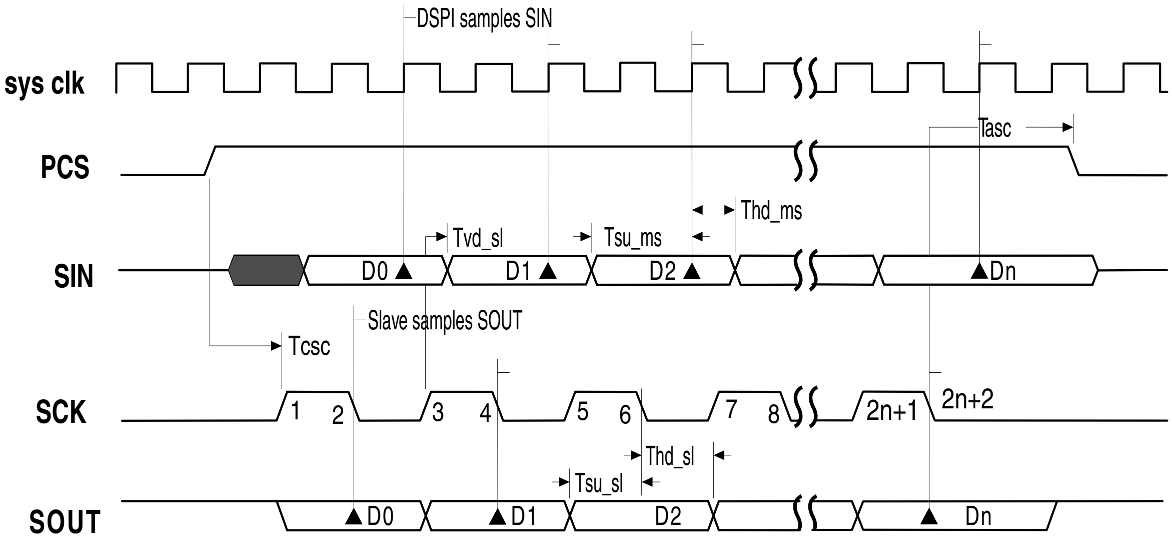


Figure 36-10. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/2$)

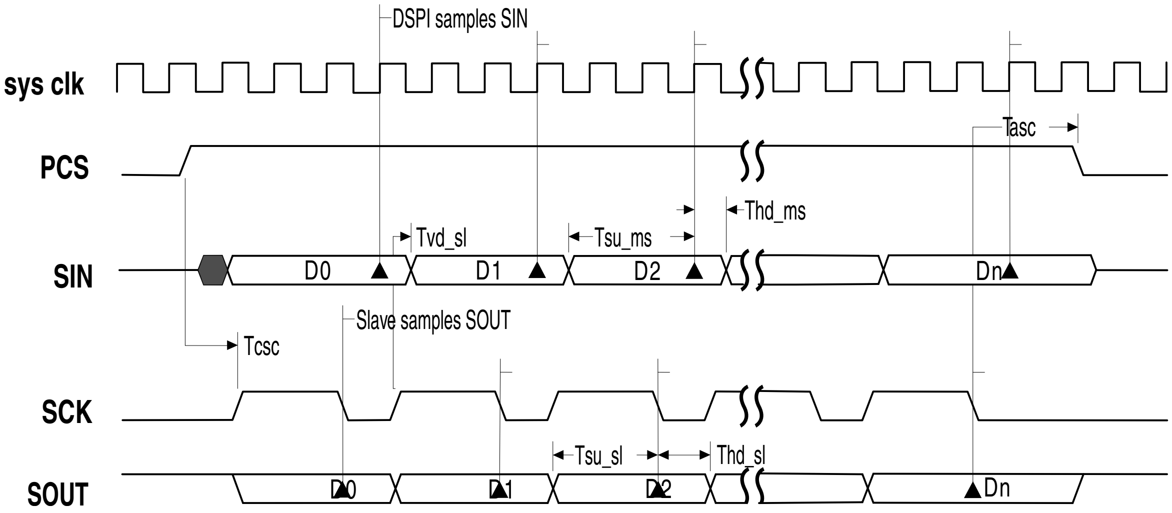


Figure 36-11. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/3$)

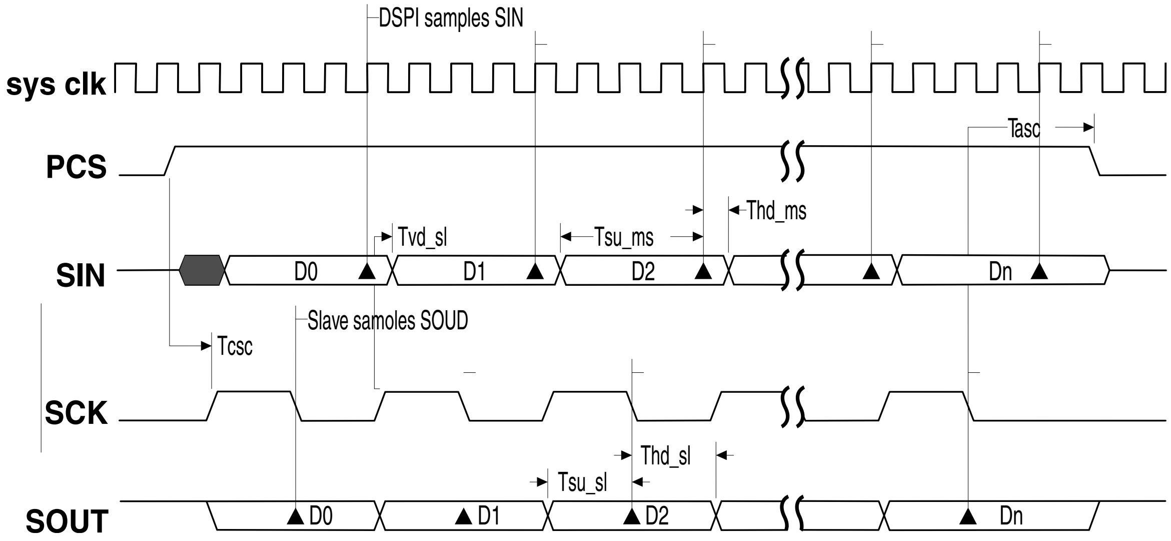
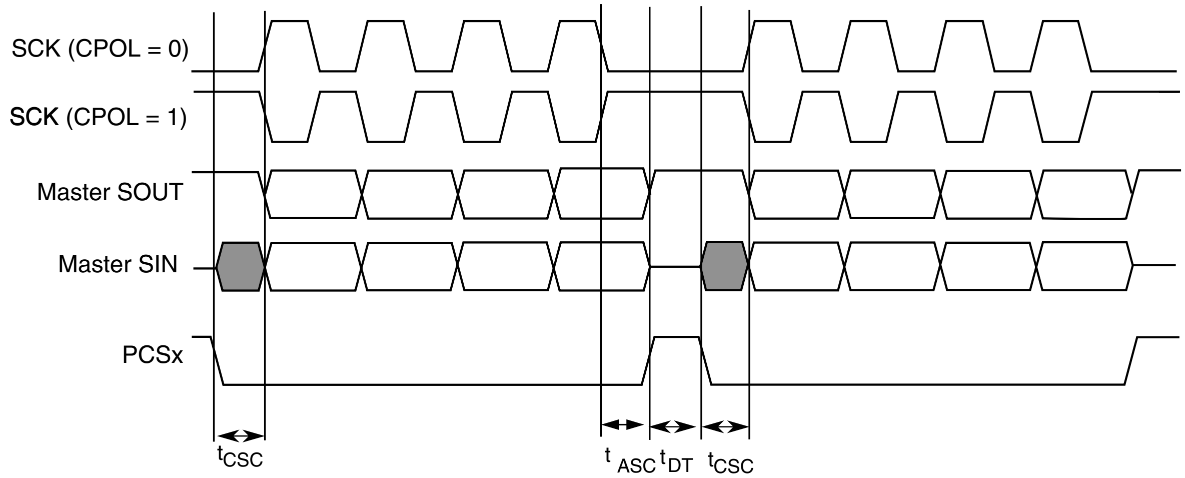


Figure 36-12. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/4$)

36.4.4.5 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle the following case. The Continuous Selection Format is enabled for the SPI configuration by setting the CONT bit in the SPI command.

When the CONT bit = 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.



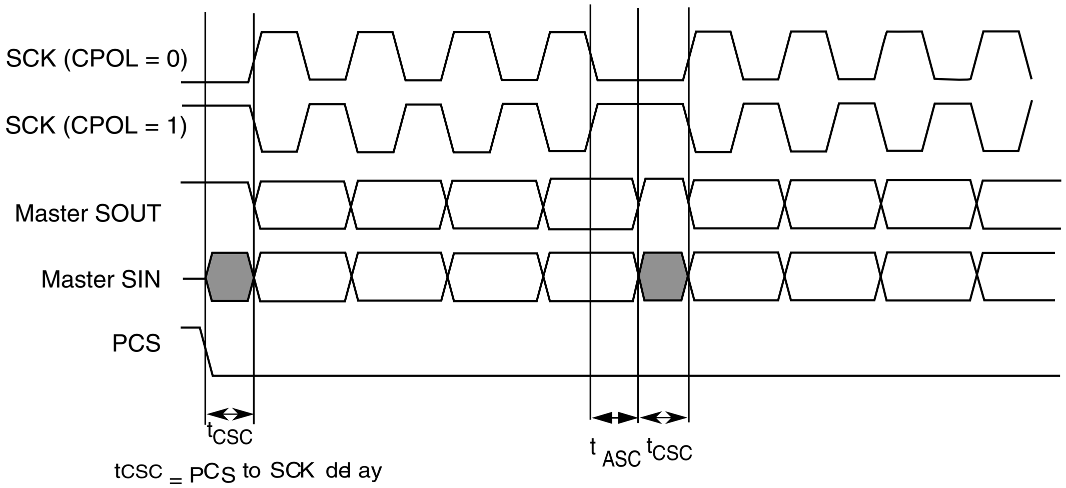
t_{CSC} = PCS to SCK dela

t_{ASC} = After SCK delay

t_{DT} = Delay after Transfer (minimum CS negation time)

Figure 36-13. Example of non-continuous format (CPHA=1, CONT=0)

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers (t_{DT}) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.



tCSC = PCS to SCK del ay

t ASC – After SCK delav

Figure 36-14. Example of continuous transfer (CPHA=1, CONT=1)

When using the module with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.

- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- PUSHR[CONT] must be deasserted before asserting MCR[HALT] in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

NOTE

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

36.4.5 Continuous Serial Communications Clock

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK only if MCR[HALT] bit is low. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer (t_{DT}) is fixed to one SCK cycle. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

NOTE

In Continuous SCK mode, for the SPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR_TXF] field before initiating transfer.

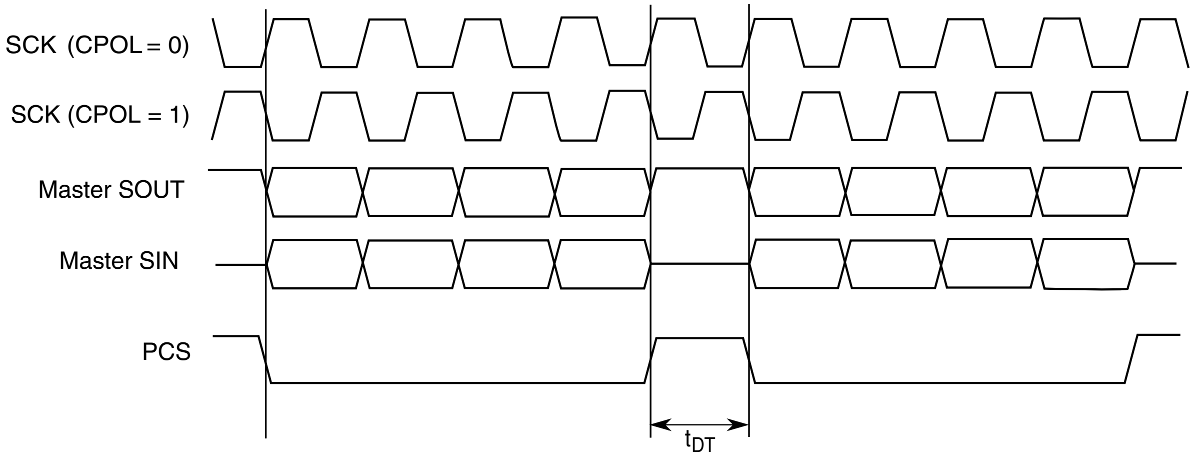


Figure 36-15. Continuous SCK Timing Diagram (CONT=0)

If the CONT bit in the TX FIFO entry is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.
- Continuous SCK with CONT bit set and entering Stopped state (refer to Start and Stop of module transfers).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

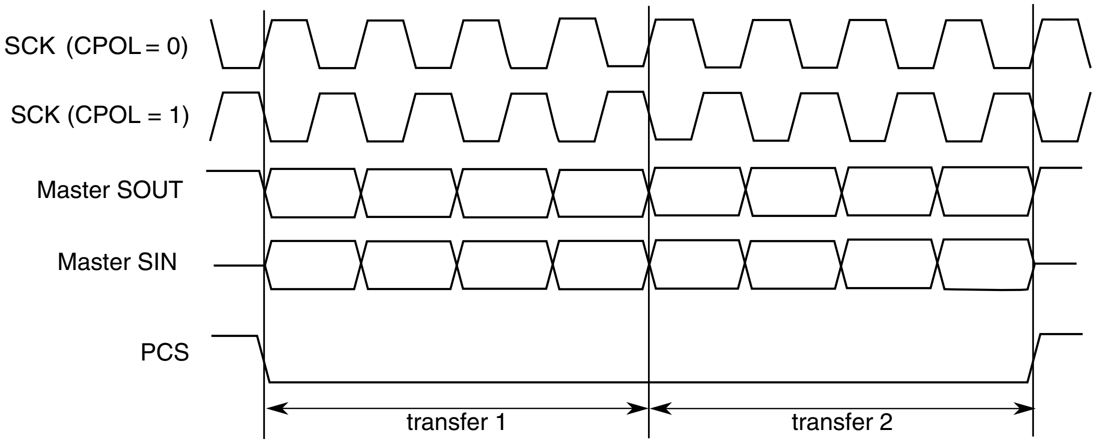


Figure 36-16. Continuous SCK timing diagram (CONT=1)

36.4.6 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the module is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the $\overline{\mathrm{SS}}$ signal is asserted and any time when transmit data is ready and $\overline{\mathrm{SS}}$ signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the $\overline{\mathrm{SS}}$ negates before that last SCK edge, the data from shift register is lost.

36.4.7 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

Table 36-9. Interrupt and DMA request conditions

Condition	Flag	Interrupt	DMA
End of Queue (EOQ)	EOQF	Yes	-
TX FIFO Fill	TFFF	Yes	Yes
Transfer Complete	TCF	Yes	-
TX FIFO Underflow	TFUF	Yes	-

Table continues on the next page...

Table 36-9. Interrupt and DMA request conditions (continued)

Condition	Flag	Interrupt	DMA
RX FIFO Drain	RFDF	Yes	Yes
RX FIFO Overflow	RFOF	Yes	-

Each condition has a flag bit in the module Status Register (SR) and a Request Enable bit in the DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of RSER register.

The module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

36.4.7.1 End Of Queue interrupt request

The End Of Queue (EOQ) interrupt request indicates that the end of a transmit queue is reached. The module generates the interrupt request when EOQ interrupt requests are enabled (RSER[EOQF_RE]) and the EOQ bit in the executing SPI command is 1.

The module generates the interrupt request when the last bit of the SPI frame with EOQ bit set is transmitted.

36.4.7.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF_RE bit in the RSER is set. The TFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

NOTE

TFFF flag clears automatically when DMA is used to fill TX FIFO.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

36.4.7.3 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF_RE bit is set in the RSER.

36.4.7.4 Transmit FIFO Underflow Interrupt Request

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in Slave mode and SPI configuration . The TFUF bit is set when the TX FIFO of the module is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF_RE bit in the RSER is set, an interrupt request is generated.

36.4.7.5 Receive FIFO Drain Interrupt or DMA Request

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF_RE bit in the RSER is set. The RFDF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

36.4.7.6 Receive FIFO Overflow Interrupt Request

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

36.4.8 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

36.4.8.1 Stop mode (External Stop mode)

This module supports the Stop mode protocol. When a request is made to enter External Stop mode, the module acknowledges the request . If a serial transfer is in progress, then this module waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, this module's memory-mapped logic is not accessible. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

36.4.8.2 Module Disable mode

Module Disable mode is a block-specific mode that the module can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware.

When the MDIS bit is set, the module negates the Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, it is said to have entered Module Disable Mode. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state.If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHR Register does not change the state of the TX FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS_TXF and DIS_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

36.5 Initialization/application information

This section describes how to initialize the module.

36.5.1 How to manage queues

The queues are not part of the module, but it includes features in support of queue management. Queues are primarily supported in SPI configuration.

1. When module executes last command word from a queue, the EOQ bit in the command word is set to indicate it that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the module in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.
7. Modify DMA descriptor of TX and RX channels for new queues
8. Flush TX FIFO by writing a 1 to the CLR_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to SPI_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the module TX FIFO, and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

36.5.2 Switching Master and Slave mode

When changing modes in the module, follow the steps below to guarantee proper operation.

1. Halt it by setting MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to the CLR_TXF and CLR_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable it by clearing MCR[HALT].

36.5.3 Initializing Module in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the module is enabled by clearing MCR[HALT]. It should be ensured that module Slave is enabled before enabling it's Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

36.5.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz protocol frequency and the double baud rate DBR bit is cleared.

Table 36-10. Baud rate values (bps)

		Baud rate divider prescaler values			
		2	3	5	7
Baud Rate Scaler Values	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
	32768	1.53k	1.02k	610	436

36.5.5 Delay settings

The following table shows the values for the Delay after Transfer (t_{DT}) and CS to SCK Delay (T_{CSC}) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz protocol frequency.

NOTE

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 36-11. Delay values

		Delay prescaler values			
		1	3	5	7
Delay scaler values	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 µs
	32	320.0 ns	960.0 ns	1.6 µs	2.2 µs
	64	640.0 ns	1.9 µs	3.2 µs	4.5 µs
	128	1.3 µs	3.8 µs	6.4 µs	9.0 µs
	256	2.6 µs	7.7 µs	12.8 µs	17.9 µs
	512	5.1 µs	15.4 µs	25.6 µs	35.8 µs
	1024	10.2 µs	30.7 µs	51.2 µs	71.7 µs
	2048	20.5 µs	61.4 µs	102.4 µs	143.4 µs
	4096	41.0 µs	122.9 µs	204.8 µs	286.7 µs
	8192	81.9 µs	245.8 µs	409.6 µs	573.4 µs
	16384	163.8 µs	491.5 µs	819.2 µs	1.1 ms
	32768	327.7 µs	983.0 µs	1.6 ms	2.3 ms
	65536	655.4 µs	2.0 ms	3.3 ms	4.6 ms

36.5.6 Calculation of FIFO pointer addresses

Complete visibility of the FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX

FIFO is chosen for the illustration, but the concepts carry over. See Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism for details on the FIFO operation.

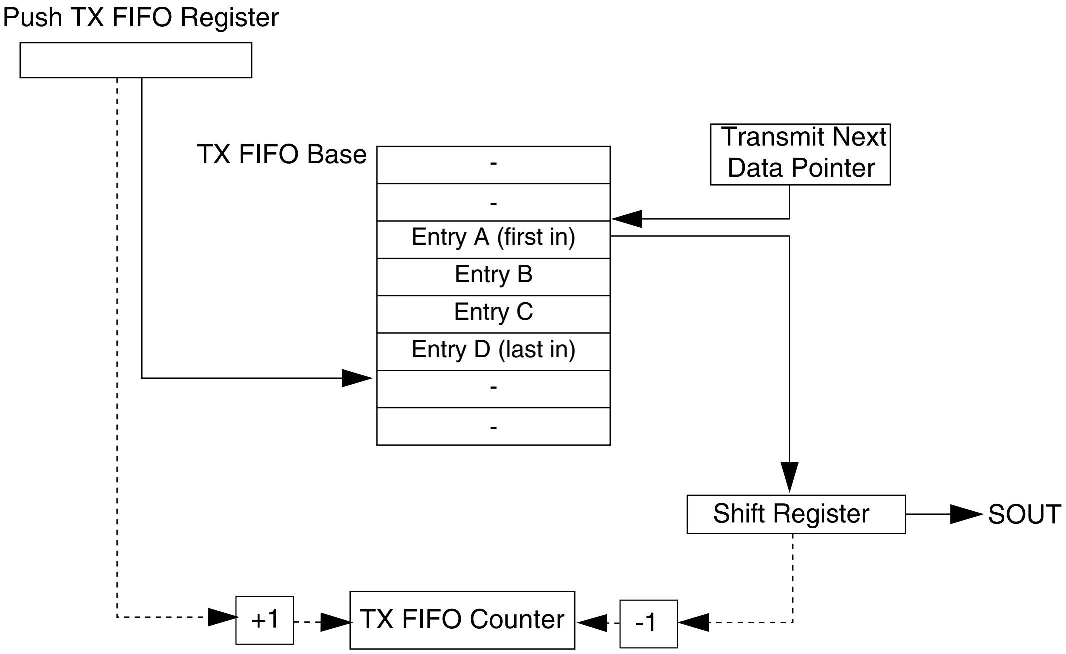


Figure 36-17. TX FIFO pointers and counter

36.5.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1)\text{mod}(\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

36.5.6.2 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + \left(4 \times \text{POPNXTPTR}\right)$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times \left(\text{RXCTR} + \text{POPNXTPTR} - 1\right)\text{mod}\left(\text{RXFIFOdepth}\right)$$

RX FIFO Base - Base address of RX FIFO
RXCTR - RX FIFO counter
POPNXTPTR - Pop Next Pointer
RX FIFO Depth - Receive FIFO depth, implementation specific

Chapter 37
Inter-Integrated Circuit (I2C)

The inter-integrated circuit (I^{2}C, I2C, or IIC) module provides a method of communication between a number of devices.

37.1 Introduction

The inter-integrated circuit (I^{2}C, I2C, or IIC) module provides a method of communication between a number of devices.

The interface is designed to operate up to at least 400 kbit/s with maximum bus loading and timing. The I2C device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF. The I2C module also complies with the *System Management Bus (SMBus) Specification, version 2*.

37.1.1 Features

The I2C module has the following features:

- Compatible with *The I^{2}C-Bus Specification*
- Multimaster operation
- Software programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- START and STOP signal generation and detection
- Repeated START signal generation and detection
- Acknowledge bit generation and detection

- Bus busy detection
- General call recognition
- 10-bit address extension
- Support for *System Management Bus (SMBus) Specification, version 2*
- Programmable input glitch filter
- Low power mode wakeup on slave address match
- Range slave address support
- DMA support
- Double buffering support to achieve higher baud rate

37.1.2 Modes of operation

The I2C module's operation in various low power modes is as follows:

- Run mode: This is the basic mode of operation. To conserve power in this mode, disable the module.
- Wait mode: The module continues to operate when the core is in Wait mode and can provide a wakeup interrupt.
- Stop mode: The module is inactive in Stop mode for reduced power consumption, except that address matching is enabled in Stop mode. The STOP instruction does not affect the I2C module's register states.

37.1.3 Block diagram

The following figure is a functional block diagram of the I2C module.

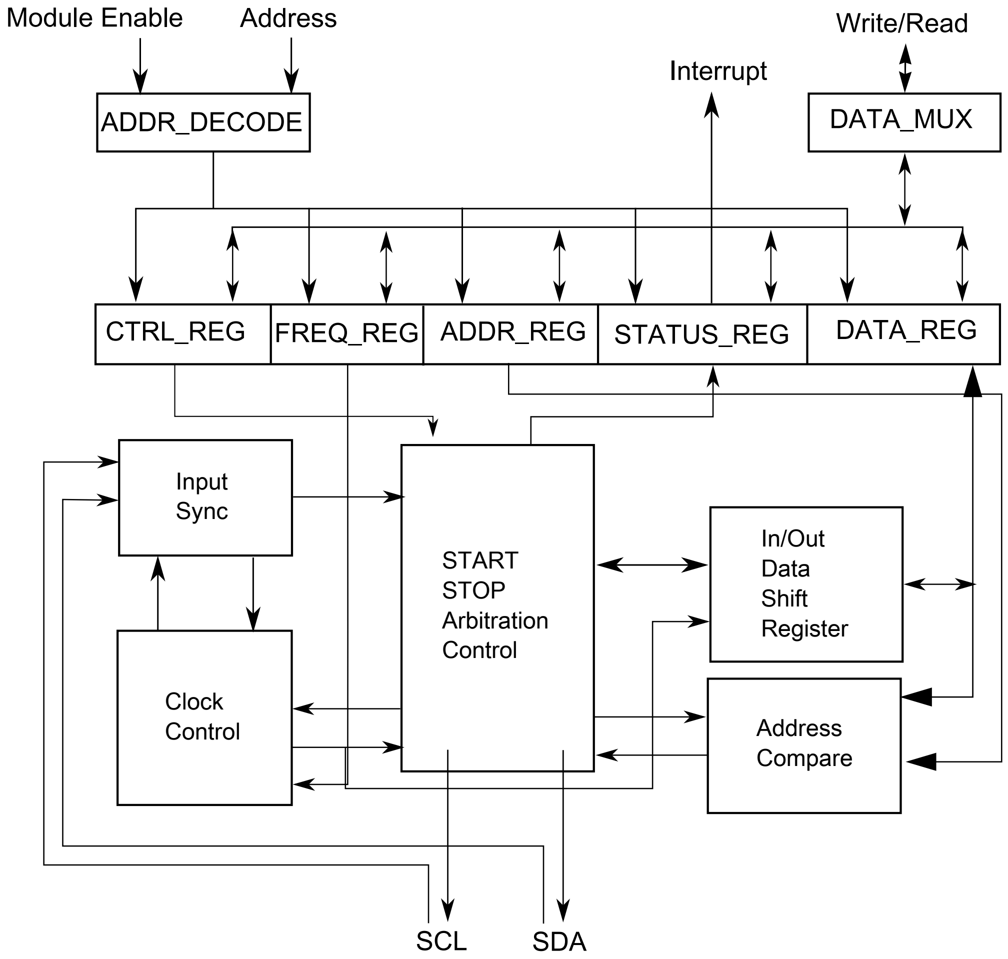


Figure 37-1. I2C Functional block diagram

37.2 I^{2}C signal descriptions

The signal properties of I^{2}C are shown in the table found here.

Table 37-1. I^{2}C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock line of the I^{2}C system.	I/O
SDA	Bidirectional serial data line of the I^{2}C system.	I/O

37.3 Memory map/register definition

This section describes in detail all I2C registers accessible to the end user.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_6000	I2C Address Register 1 (I2C0_A1)	8	R/W	00h	37.3.1/733
4006_6001	I2C Frequency Divider register (I2C0_F)	8	R/W	00h	37.3.2/733
4006_6002	I2C Control Register 1 (I2C0_C1)	8	R/W	00h	37.3.3/734
4006_6003	I2C Status register (I2C0_S)	8	R/W	80h	37.3.4/736
4006_6004	I2C Data I/O register (I2C0_D)	8	R/W	00h	37.3.5/738
4006_6005	I2C Control Register 2 (I2C0_C2)	8	R/W	00h	37.3.6/738
4006_6006	I2C Programmable Input Glitch Filter Register (I2C0_FLT)	8	R/W	00h	37.3.7/739
4006_6007	I2C Range Address register (I2C0_RA)	8	R/W	00h	37.3.8/741
4006_6008	I2C SMBus Control and Status register (I2C0_SMB)	8	R/W	00h	37.3.9/741
4006_6009	I2C Address Register 2 (I2C0_A2)	8	R/W	C2h	37.3.10/ 743
4006_600A	I2C SCL Low Timeout Register High (I2C0_SLTH)	8	R/W	00h	37.3.11/ 743
4006_600B	I2C SCL Low Timeout Register Low (I2C0_SLTL)	8	R/W	00h	37.3.12/ 744
4006_600C	I2C Status register 2 (I2C0_S2)	8	R/W	01h	37.3.13/ 744
4006_7000	I2C Address Register 1 (I2C1_A1)	8	R/W	00h	37.3.1/733
4006_7001	I2C Frequency Divider register (I2C1_F)	8	R/W	00h	37.3.2/733
4006_7002	I2C Control Register 1 (I2C1_C1)	8	R/W	00h	37.3.3/734
4006_7003	I2C Status register (I2C1_S)	8	R/W	80h	37.3.4/736
4006_7004	I2C Data I/O register (I2C1_D)	8	R/W	00h	37.3.5/738
4006_7005	I2C Control Register 2 (I2C1_C2)	8	R/W	00h	37.3.6/738
4006_7006	I2C Programmable Input Glitch Filter Register (I2C1_FLT)	8	R/W	00h	37.3.7/739
4006_7007	I2C Range Address register (I2C1_RA)	8	R/W	00h	37.3.8/741
4006_7008	I2C SMBus Control and Status register (I2C1_SMB)	8	R/W	00h	37.3.9/741
4006_7009	I2C Address Register 2 (I2C1_A2)	8	R/W	C2h	37.3.10/ 743
4006_700A	I2C SCL Low Timeout Register High (I2C1_SLTH)	8	R/W	00h	37.3.11/ 743
4006_700B	I2C SCL Low Timeout Register Low (I2C1_SLTL)	8	R/W	00h	37.3.12/ 744
4006_700C	I2C Status register 2 (I2C1_S2)	8	R/W	01h	37.3.13/ 744

37.3.1 I2C Address Register 1 (I2Cx_A1)

This register contains the slave address to be used by the I2C module.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	AD[7:1]							0
Write								▨
Reset	0	0	0	0	0	0	0	0

I2Cx_A1 field descriptions

Field	Description
7–1 AD[7:1]	Address Contains the primary slave address used by the I2C module when it is addressed as a slave. This field is used in the 7-bit address scheme and the lower seven bits in the 10-bit address scheme.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

37.3.2 I2C Frequency Divider register (I2Cx_F)

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read Write	MULT		ICR					
Reset	0	0	0	0	0	0	0	0

I2Cx_F field descriptions

Field	Description
7–6 MULT	Multiplier Factor Defines the multiplier factor (mul). This factor is used along with the SCL divider to generate the I2C baud rate. 00 mul = 1 01 mul = 2 10 mul = 4 11 Reserved
ICR	ClockRate Prescales the I2C module clock for bit rate selection. This field and the MULT field determine the I2C baud rate, the SDA hold time, the SCL start hold time, and the SCL stop hold time. For a list of values corresponding to each ICR setting, see I2C divider and hold values. The SCL divider multiplied by multiplier factor (mul) determines the I2C baud rate. `I2C baud rate = I2C module clock speed (Hz)/(mul × SCL divider)`

Table continues on the next page...

I2Cx_F field descriptions (continued)

Field	Description
	The SDA hold time is the delay from the falling edge of SCL (I2C clock) to the changing of SDA (I2C data).

SDA hold time = I2C module clock period (s) × mul × SDA hold value

The SCL start hold time is the delay from the falling edge of SDA (I2C data) while SCL is high (start condition) to the falling edge of SCL (I2C clock).

SCL start hold time = I2C module clock period (s) × mul × SCL start hold value

The SCL stop hold time is the delay from the rising edge of SCL (I2C clock) to the rising edge of SDA (I2C data) while SCL is high (stop condition).

SCL stop hold time = I2C module clock period (s) × mul × SCL stop hold value

For example, if the I2C module clock speed is 8 MHz, the following table shows the possible hold time values with different ICR and MULT selections to achieve an I^{2}C baud rate of 100 kbit/s.

MULT	ICR	Hold times (μs)		
		SDA	SCL Start	SCL Stop
2h	00h	3.500	3.000	5.500
1h	07h	2.500	4.000	5.250
1h	0Bh	2.250	4.000	5.250
0h	14h	2.125	4.250	5.125
0h	18h	1.125	4.750	5.125

37.3.3 I2C Control Register 1 (I2Cx_C1)

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read	IICEN	IICIE	MST	TX	TXAK	0	WUEN	DMAEN
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_C1 field descriptions

Field	Description
7 IICEN	I2C Enable Enables I2C module operation. 0 Disabled 1 Enabled
6 IICIE	I2C Interrupt Enable Enables I2C interrupt requests.

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
5 MST	Master Mode Select When MST is changed from 0 to 1, a START signal is generated on the bus and master mode is selected. When this bit changes from 1 to 0, a STOP signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	Transmit Mode Select Selects the direction of master and slave transfers. In master mode this bit must be set according to the type of transfer required. Therefore, for address cycles, this bit is always set. When addressed as a slave this bit must be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	Transmit Acknowledge Enable Specifies the value driven onto the SDA during data acknowledge cycles for both master and slave receivers. The value of SMB[FACK] affects NACK/ACK generation. **NOTE:** SCL is held low until TXAK is written. 0 An acknowledge signal is sent to the bus on the following receiving byte (if FACK is cleared) or the current receiving byte (if FACK is set). 1 No acknowledge signal is sent to the bus on the following receiving data byte (if FACK is cleared) or the current receiving data byte (if FACK is set).
2 RSTA	Repeat START Writing 1 to this bit generates a repeated START condition provided it is the current master. This bit will always be read as 0. Attempting a repeat at the wrong time results in loss of arbitration.
1 WUEN	Wakeup Enable The I2C module can wake the MCU from low power mode with no peripheral bus running when slave address matching occurs. 0 Normal operation. No interrupt generated when address matching in low power mode. 1 Enables the wakeup function in low power mode.
0 DMAEN	DMA Enable Enables or disables the DMA function. 0 All DMA signalling disabled. 1 DMA transfer is enabled. While SMB[FACK] = 0, the following conditions trigger the DMA request: • a data byte is received, and either address or data is transmitted. (ACK/NACK is automatic) • the first byte received matches the A1 register or is a general call address.

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
	If any address matching occurs, S[IAAS] and S[TCF] are set. If the direction of transfer is known from master to slave, then it is not required to check S[SRW]. With this assumption, DMA can also be used in this case. In other cases, if the master reads data from the slave, then it is required to rewrite the C1 register operation. With this assumption, DMA cannot be used. When FACK = 1, an address or a data byte is transmitted.

37.3.4 I2C Status register (I2Cx_S)

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read	TCF	IAAS	BUSY	ARBL	RAM	SRW	IICIF	RXAK
Write				w1c			w1c	
Reset	1	0	0	0	0	0	0	0

I2Cx_S field descriptions

Field	Description
7 TCF	Transfer Complete Flag Acknowledges a byte transfer; TCF is set on the completion of a byte transfer. This bit is valid only during or immediately following a transfer to or from the I2C module. TCF is cleared by reading the I2C data register in receive mode or by writing to the I2C data register in transmit mode. **NOTE:** In the buffer mode, TCF is cleared automatically by internal reading or writing the data register I2C_D, with no need waiting for manually reading/writing the I2C data register in Rx/Tx mode. 0 Transfer in progress 1 Transfer complete
6 IAAS	Addressed As A Slave This bit is set by one of the following conditions: • The calling address matches the programmed primary slave address in the A1 register, or matches the range address in the RA register (which must be set to a nonzero value and under the condition I2C_C2[RMEN] = 1). • C2[GCAEN] is set and a general call is received. • SMB[SIICAEN] is set and the calling address matches the second programmed slave address. • ALERTEN is set and an SMBus alert response address is received • RMEN is set and an address is received that is within the range between the values of the A1 and RA registers. IAAS sets before the ACK bit. The CPU must check the SRW bit and set TX/RX accordingly. Writing the C1 register with any value clears this bit. 0 Not addressed 1 Addressed as a slave
5 BUSY	Bus Busy

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	Indicates the status of the bus regardless of slave or master mode. This bit is set when a START signal is detected and cleared when a STOP signal is detected. 0 Bus is idle 1 Bus is busy
4 ARBL	Arbitration Lost This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software, by writing 1 to it. 0 Standard bus operation. 1 Loss of arbitration.
3 RAM	Range Address Match This bit is set to 1 by any of the following conditions, if I2C_C2[RMEN] = 1: • Any nonzero calling address is received that matches the address in the RA register. • The calling address is within the range of values of the A1 and RA registers. **NOTE:** For the RAM bit to be set to 1 correctly, C1[IICIE] must be set to 1. Writing the C1 register with any value clears this bit to 0. 0 Not addressed 1 Addressed as a slave
2 SRW	Slave Read/Write When addressed as a slave, SRW indicates the value of the R/$\overline{W}$ command bit of the calling address sent to the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IICIF	Interrupt Flag This bit sets when an interrupt is pending. This bit must be cleared by software by writing 1 to it, such as in the interrupt routine. One of the following events can set this bit: • One byte transfer, including ACK/NACK bit, completes if FACK is 0. An ACK or NACK is sent on the bus by writing 0 or 1 to TXAK after this bit is set in receive mode. • One byte transfer, excluding ACK/NACK bit, completes if FACK is 1. • Match of slave address to calling address including primary slave address, range slave address, alert response address, second slave address, or general call address. • Arbitration lost • In SMBus mode, any timeouts except SCL and SDA high timeouts • I2C bus stop or start detection if the SSIE bit in the Input Glitch Filter register is 1 **NOTE:** To clear the I2C bus stop or start detection interrupt: In the interrupt service routine, first clear the STOPF or STARTF bit in the Input Glitch Filter register by writing 1 to it, and then clear the IICIF bit. If this sequence is reversed, the IICIF bit is asserted again. 0 No interrupt pending 1 Interrupt pending
0 RXAK	Receive Acknowledge

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	0 Acknowledge signal was received after the completion of one byte of data transmission on the bus 1 No acknowledge signal detected

37.3.5 I2C Data I/O register (I2Cx_D)

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read Write	DATA							
Reset	0	0	0	0	0	0	0	0

I2Cx_D field descriptions

Field	Description
DATA	Data In master transmit mode, when data is written to this register, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data. **NOTE:** When making the transition out of master receive mode, switch the I2C mode before reading the Data register to prevent an inadvertent initiation of a master receive data transfer. In slave mode, the same functions are available after an address match occurs. The C1[TX] bit must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For example, if the I2C module is configured for master transmit but a master receive is desired, reading the Data register does not initiate the receive. Reading the Data register returns the last byte received while the I2C module is configured in master receive or slave receive mode. The Data register does not reflect every byte that is transmitted on the I2C bus, and neither can software verify that a byte has been written to the Data register correctly by reading it back. In master transmit mode, the first byte of data written to the Data register following assertion of MST (start bit) or assertion of RSTA (repeated start bit) is used for the address transfer and must consist of the calling address (in bits 7-1) concatenated with the required R/$\overline{W}$ bit (in position bit 0).

37.3.6 I2C Control Register 2 (I2Cx_C2)

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read Write	GCAEN	ADEXT	HDRS	SBRC	RMEN	AD[10:8]		
Reset	0	0	0	0	0	0	0	0

I2Cx_C2 field descriptions

Field	Description
7 GCAEN	General Call Address Enable Enables general call address. 0 Disabled 1 Enabled
6 ADEXT	Address Extension Controls the number of bits used for the slave address. 0 7-bit address scheme 1 10-bit address scheme
5 HDRS	High Drive Select Controls the drive capability of the I2C pads. 0 Normal drive mode 1 High drive mode
4 SBRC	Slave Baud Rate Control Enables independent slave mode baud rate at maximum frequency, which forces clock stretching on SCL in very fast I2C modes. To a slave, an example of a "very fast" mode is when the master transfers at 40 kbit/s but the slave can capture the master's data at only 10 kbit/s. 0 The slave baud rate follows the master baud rate and clock stretching may occur 1 Slave baud rate is independent of the master baud rate
3 RMEN	Range Address Matching Enable This bit controls the slave address matching for addresses between the values of the A1 and RA registers. When this bit is set, a slave address matching occurs for any address greater than the value of the A1 register and less than or equal to the value of the RA register. 0 Range mode disabled. No address matching occurs for an address within the range of values of the A1 and RA registers. 1 Range mode enabled. Address matching occurs when a slave receives an address within the range of values of the A1 and RA registers.
AD[10:8]	Slave Address Contains the upper three bits of the slave address in the 10-bit address scheme. This field is valid only while the ADEXT bit is set.

37.3.7 I2C Programmable Input Glitch Filter Register (I2Cx_FLT)

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read	SHEN	STOPF	SSIE	STARTF	FLT			
Write		w1c		w1c				
Reset	0	0	0	0	0	0	0	0

I2Cx_FLT field descriptions

Field	Description
7 SHEN	Stop Hold Enable Set this bit to hold off entry to stop mode when any data transmission or reception is occurring. The following scenario explains the holdoff functionality: 1. The I2C module is configured for a basic transfer, and the SHEN bit is set to 1. 2. A transfer begins. 3. The MCU signals the I2C module to enter stop mode. 4. The byte currently being transferred, including both address and data, completes its transfer. 5. The I2C slave or master acknowledges that the in-transfer byte completed its transfer and acknowledges the request to enter stop mode. 6. After receiving the I2C module's acknowledgment of the request to enter stop mode, the MCU determines whether to shut off the I2C module's clock. If the SHEN bit is set to 1 and the I2C module is in an idle or disabled state when the MCU signals to enter stop mode, the module immediately acknowledges the request to enter stop mode. If SHEN is cleared to 0 and the overall data transmission or reception that was suspended by stop mode entry was incomplete: To resume the overall transmission or reception after the MCU exits stop mode, software must reinitialize the transfer by resending the address of the slave. If the I2C Control Register 1's IICIE bit was set to 1 before the MCU entered stop mode, system software will receive the interrupt triggered by the I2C Status Register's TCF bit after the MCU wakes from the stop mode. 0 Stop holdoff is disabled. The MCU's entry to stop mode is not gated. 1 Stop holdoff is enabled.
6 STOPF	I2C Bus Stop Detect Flag Hardware sets this bit when the I2C bus's stop status is detected. The STOPF bit must be cleared by writing 1 to it. **NOTE:** The stop flag is only for the matched slave devices, therefore the master will not respond for it. 0 No stop happens on I2C bus 1 Stop detected on I2C bus
5 SSIE	I2C Bus Stop or Start Interrupt Enable This bit enables the interrupt for I2C bus stop or start detection. **NOTE:** To clear the I2C bus stop or start detection interrupt: In the interrupt service routine, first clear the STOPF or STARTF bit by writing 1 to it, and then clear the IICIF bit in the status register. If this sequence is reversed, the IICIF bit is asserted again. 0 Stop or start detection interrupt is disabled 1 Stop or start detection interrupt is enabled
4 STARTF	I2C Bus Start Detect Flag Hardware sets this bit when the I2C bus's start status is detected. The STARTF bit must be cleared by writing 1 to it. 0 No start happens on I2C bus 1 Start detected on I2C bus
FLT	I2C Programmable Filter Factor

Table continues on the next page...

I2Cx_FLT field descriptions (continued)

Field	Description
	Controls the width of the glitch, in terms of I2C module clock cycles, that the filter must absorb. For any glitch whose size is less than or equal to this width setting, the filter does not allow the glitch to pass. 0h No filter/bypass 1-Fh Filter glitches up to width of *n* I2C module clock cycles, where *n*=1-15d

37.3.8 I2C Range Address register (I2Cx_RA)

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read	RAD							0
Write								▨
Reset	0	0	0	0	0	0	0	0

I2Cx_RA field descriptions

Field	Description
7–1 RAD	Range Slave Address This field contains the slave address to be used by the I2C module. The field is used in the 7-bit address scheme. If I2C_C2[RMEN] is set to 1, any nonzero value write enables this register. This register value can be considered as a maximum boundary in the range matching mode.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

37.3.9 I2C SMBus Control and Status register (I2Cx_SMB)

NOTE

When the SCL and SDA signals are held high for a length of time greater than the high timeout period, the SHTF1 flag sets. Before reaching this threshold, while the system is detecting how long these signals are being held high, a master assumes that the bus is free. However, the SHTF1 bit is set to 1 in the bus transmission process with the idle bus state.

NOTE

When the TCKSEL bit is set, there is no need to monitor the SHTF1 bit because the bus speed is too high to match the protocol of SMBus.

Address: Base address + 8h offset

Bit	7	6	5	4	3	2	1	0
Read	FACK	ALERTEN	SIICAEN	TCKSEL	SLTF	SHTF1	SHTF2	SHTF2IE
Write					w1c	▨	w1c	
Reset	0	0	0	0	0	0	0	0

I2Cx_SMB field descriptions

Field	Description
7 FACK	Fast NACK/ACK Enable For SMBus packet error checking, the CPU must be able to issue an ACK or NACK according to the result of receiving data byte. 0 An ACK or NACK is sent on the following receiving data byte 1 Writing 0 to TXAK after receiving a data byte generates an ACK. Writing 1 to TXAK after receiving a data byte generates a NACK. **NOTE:** Enable I2C_S2[DFEN] in the master receive mode.
6 ALERTEN	SMBus Alert Response Address Enable Enables or disables SMBus alert response address matching. **NOTE:** After the host responds to a device that used the alert response address, you must use software to put the device's address on the bus. The alert protocol is described in the SMBus specification. 0 SMBus alert response address matching is disabled 1 SMBus alert response address matching is enabled
5 SIICAEN	Second I2C Address Enable Enables or disables SMBus device default address. 0 I2C address register 2 matching is disabled 1 I2C address register 2 matching is enabled
4 TCKSEL	Timeout Counter Clock Select Selects the clock source of the timeout counter. 0 Timeout counter counts at the frequency of the I2C module clock / 64 1 Timeout counter counts at the frequency of the I2C module clock
3 SLTF	SCL Low Timeout Flag This bit is set when the SLT register (consisting of the SLTH and SLTL registers) is loaded with a non-zero value (LoValue) and an SCL low timeout occurs. Software clears this bit by writing a logic 1 to it. **NOTE:** The low timeout function is disabled when the SLT register's value is 0. 0 No low timeout occurs 1 Low timeout occurs
2 SHTF1	SCL High Timeout Flag 1 This read-only bit sets when SCL and SDA are held high more than clock × LoValue / 512, which indicates the bus is free. This bit is cleared automatically.

Table continues on the next page...

I2Cx_SMB field descriptions (continued)

Field	Description
	0 No SCL high and SDA high timeout occurs 1 SCL high and SDA high timeout occurs
1 SHTF2	SCL High Timeout Flag 2 This bit sets when SCL is held high and SDA is held low more than clock × LoValue / 512. Software clears this bit by writing 1 to it. 0 No SCL high and SDA low timeout occurs 1 SCL high and SDA low timeout occurs
0 SHTF2IE	SHTF2 Interrupt Enable Enables SCL high and SDA low timeout interrupt. 0 SHTF2 interrupt is disabled 1 SHTF2 interrupt is enabled

37.3.10 I2C Address Register 2 (I2Cx_A2)

Address: Base address + 9h offset

Bit	7	6	5	4	3	2	1	0
Read	SAD							0
Write								▨
Reset	1	1	0	0	0	0	1	0

I2Cx_A2 field descriptions

Field	Description
7–1 SAD	SMBus Address Contains the slave address used by the SMBus. This field is used on the device default address or other related addresses.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

37.3.11 I2C SCL Low Timeout Register High (I2Cx_SLTH)

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[15:8]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTH field descriptions

Field	Description
SSLT[15:8]	SSLT[15:8] Most significant byte of SCL low timeout value that determines the timeout period of SCL low.

37.3.12 I2C SCL Low Timeout Register Low (I2Cx_SLTL)

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[7:0]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTL field descriptions

Field	Description
SSLT[7:0]	SSLT[7:0] Least significant byte of SCL low timeout value that determines the timeout period of SCL low.

37.3.13 I2C Status register 2 (I2Cx_S2)

Address: Base address + Ch offset

Bit	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	DFEN	ERROR	EMPTY
Write							w1c	
Reset	0	0	0	0	0	0	0	1

I2Cx_S2 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 DFEN	Double Buffer Enable

Table continues on the next page...

I2Cx_S2 field descriptions (continued)

Field	Description
	Enables or disables the double buffer mode. In the double buffer mode, the clock stretch is disabled. 0 Disables the double buffer mode; clock stretch is enabled. 1 Enables the double buffer mode; clock stretch is disabled. In the slave mode, the I2C will not hold bus between data transfers.
1 ERROR	Error flag Indicates if there are read or write errors with the Tx and Rx buffers. 0 The buffer is not full and all write/read operations have no errors. 1 There are 3 or more write/read errors during the data transfer phase (when the Empty flag is not set and the buffer is busy).
0 EMPTY	Empty flag Indicates if the Tx or Rx buffer is empty. 0 Tx or Rx buffer is not empty and cannot be written to, that is new data cannot be loaded into the buffer. 1 Tx or Rx buffer is empty and can be written to, that is new data can be loaded into the buffer.

37.4 Functional description

This section provides a comprehensive functional description of the I2C module.

37.4.1 I2C protocol

The I2C bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers.

All devices connected to it must have open drain or open collector outputs. A logic AND function is exercised on both lines with external pull-up resistors. The value of these resistors depends on the system.

Normally, a standard instance of communication is composed of four parts:

1. START signal
2. Slave address transmission
3. Data transfer
4. STOP signal

The STOP signal should not be confused with the CPU STOP instruction. The following figure illustrates I2C bus system communication.

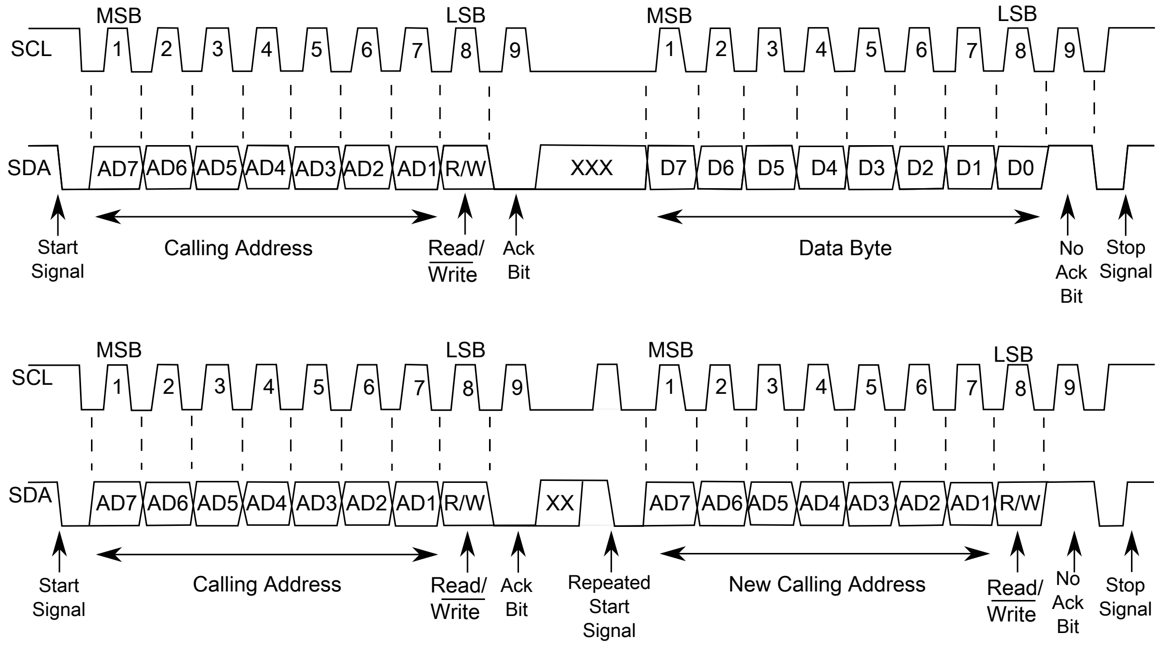


Figure 37-2. I2C bus transmission signals

37.4.1.1 START signal

The bus is free when no master device is engaging the bus (both SCL and SDA are high). When the bus is free, a master may initiate communication by sending a START signal. A START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer—each data transfer might contain several bytes of data—and brings all slaves out of their idle states.

37.4.1.2 Slave address transmission

Immediately after the START signal, the first byte of a data transfer is the slave address transmitted by the master. This address is a 7-bit calling address followed by an R/$\overline{\text{W}}$ bit. The R/$\overline{\text{W}}$ bit tells the slave the desired direction of data transfer.

- 1 = Read transfer: The slave transmits data to the master
- 0 = Write transfer: The master transmits data to the slave

Only the slave with a calling address that matches the one transmitted by the master responds by sending an acknowledge bit. The slave sends the acknowledge bit by pulling SDA low at the ninth clock.

No two slaves in the system can have the same address. If the I2C module is the master, it must not transmit an address that is equal to its own slave address. The I2C module cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the I2C module reverts to slave mode and operates correctly even if it is being addressed by another master.

37.4.1.3 Data transfers

When successful slave addressing is achieved, data transfer can proceed on a byte-by-byte basis in the direction specified by the R/$\overline{W}$ bit sent by the calling master.

All transfers that follow an address cycle are referred to as data transfers, even if they carry subaddress information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low. Data must be held stable while SCL is high. There is one clock pulse on SCL for each data bit, and the MSB is transferred first. Each data byte is followed by a ninth (acknowledge) bit, which is signaled from the receiving device by pulling SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the ninth bit, the slave must leave SDA high. The master interprets the failed acknowledgement as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets it as an end to data transfer and releases the SDA line.

In the case of a failed acknowledgement by either the slave or master, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a STOP signal.
- Commences a new call by generating a repeated START signal.

37.4.1.4 STOP signal

The master can terminate the communication by generating a STOP signal to free the bus. A STOP signal is defined as a low-to-high transition of SDA while SCL is asserted.

37.4.1.5 Repeated START signal

The master may generate a START signal followed by a calling command without generating a STOP signal first. This action is called a repeated START. The master uses a repeated START to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.The master needs to send a NACK signal before sending repeated-START in the buffering mode.

37.4.1.6 Arbitration procedure

The I2C bus is a true multimaster bus that allows more than one master to be connected on it.

If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The bus clock's low period is equal to the longest clock low period, and the high period is equal to the shortest one among the masters.

The relative priority of the contending masters is determined by a data arbitration procedure. A bus master loses arbitration if it transmits logic level 1 while another master transmits logic level 0. The losing masters immediately switch to slave receive mode and stop driving SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, hardware sets a status bit to indicate the loss of arbitration.

37.4.1.7 Clock synchronization

Because wire AND logic is performed on SCL, a high-to-low transition on SCL affects all devices connected on the bus. The devices start counting their low period and, after a device's clock has gone low, that device holds SCL low until the clock reaches its high state. However, the change of low to high in this device clock might not change the state of SCL if another device clock is still within its low period. Therefore, the synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time; see the following diagram. When all applicable devices have counted off their low period, the synchronized clock SCL is released and pulled high. Afterward there is no difference between the device clocks and the state of SCL, and all devices start counting their high periods. The first device to complete its high period pulls SCL low again.

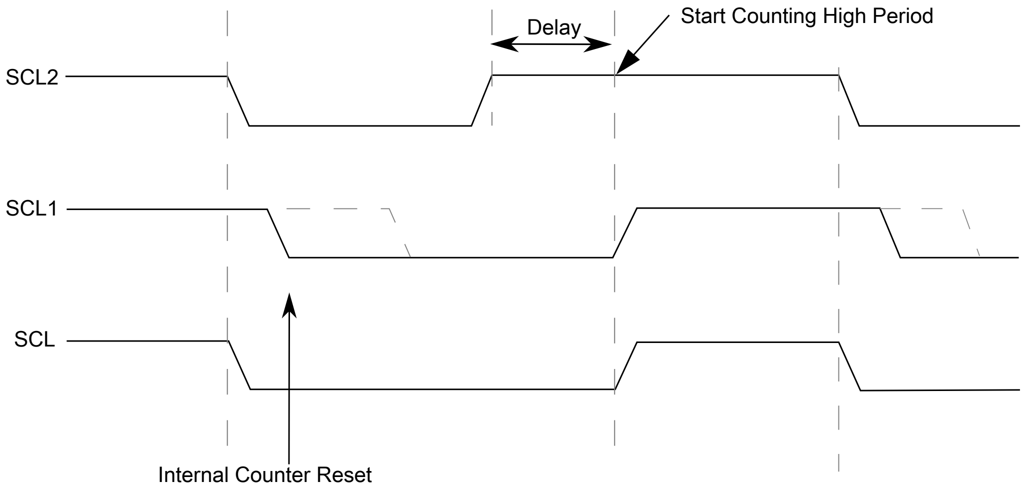


Figure 37-3. I2C clock synchronization

37.4.1.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. A slave device may hold SCL low after completing a single byte transfer (9 bits). In this case, it halts the bus clock and forces the master clock into wait states until the slave releases SCL.

37.4.1.9 Clock stretching

The clock synchronization mechanism can be used by slaves to slow down the bit rate of a transfer. After the master drives SCL low, a slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal's low period is stretched. In other words, the SCL bus signal's low period is increased to be the same length as the slave's SCL low period.

37.4.1.10 I2C divider and hold values

NOTE

For some cases on some devices, the SCL divider value may vary by ±2 or ±4 when ICR's value ranges from 00h to 0Fh. These potentially varying SCL divider values are highlighted in the following table. For the actual SCL divider values for your device, see the chip-specific details about the I2C module.

Table 37-2. I2C divider and hold values

ICR (hex)	SCL divider	SDA hold value	SCL hold (start) value	SCL hold (stop) value		ICR (hex)	SCL divider (clocks)	SDA hold (clocks)	SCL hold (start) value	SCL hold (stop) value
00	20	7	6	11		20	160	17	78	81
01	22	7	7	12		21	192	17	94	97
02	24	8	8	13		22	224	33	110	113
03	26	8	9	14		23	256	33	126	129
04	28	9	10	15		24	288	49	142	145
05	30	9	11	16		25	320	49	158	161
06	34	10	13	18		26	384	65	190	193
07	40	10	16	21		27	480	65	238	241
08	28	7	10	15		28	320	33	158	161
09	32	7	12	17		29	384	33	190	193
0A	36	9	14	19		2A	448	65	222	225
0B	40	9	16	21		2B	512	65	254	257
0C	44	11	18	23		2C	576	97	286	289
0D	48	11	20	25		2D	640	97	318	321
0E	56	13	24	29		2E	768	129	382	385
0F	68	13	30	35		2F	960	129	478	481
10	48	9	18	25		30	640	65	318	321
11	56	9	22	29		31	768	65	382	385
12	64	13	26	33		32	896	129	446	449
13	72	13	30	37		33	1024	129	510	513
14	80	17	34	41		34	1152	193	574	577
15	88	17	38	45		35	1280	193	638	641
16	104	21	46	53		36	1536	257	766	769
17	128	21	58	65		37	1920	257	958	961
18	80	9	38	41		38	1280	129	638	641
19	96	9	46	49		39	1536	129	766	769
1A	112	17	54	57		3A	1792	257	894	897
1B	128	17	62	65		3B	2048	257	1022	1025
1C	144	25	70	73		3C	2304	385	1150	1153
1D	160	25	78	81		3D	2560	385	1278	1281
1E	192	33	94	97		3E	3072	513	1534	1537
1F	240	33	118	121		3F	3840	513	1918	1921

37.4.2 10-bit address

For 10-bit addressing, 0x11110 is used for the first 5 bits of the first address byte. Various combinations of read/write formats are possible within a transfer that includes 10-bit addressing.

37.4.2.1 Master-transmitter addresses a slave-receiver

The transfer direction is not changed. When a 10-bit address follows a START condition, each slave compares the first 7 bits of the first byte of the slave address (11110XX) with its own address and tests whether the eighth bit (R/$\overline{W}$ direction bit) is 0. It is possible that more than one device finds a match and generates an acknowledge (A1). Each slave that finds a match compares the 8 bits of the second byte of the slave address with its own address, but only one slave finds a match and generates an acknowledge (A2). The matching slave remains addressed by the master until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

Table 37-3. Master-transmitter addresses slave-receiver with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	R/$\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Data	A	...	Data	A/A	P

After the master-transmitter has sent the first byte of the 10-bit address, the slave-receiver sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

37.4.2.2 Master-receiver addresses a slave-transmitter

The transfer direction is changed after the second R/$\overline{W}$ bit. Up to and including acknowledge bit A2, the procedure is the same as that described for a master-transmitter addressing a slave-receiver. After the repeated START condition (Sr), a matching slave remembers that it was addressed before. This slave then checks whether the first seven bits of the first byte of the slave address following Sr are the same as they were after the START condition (S), and it tests whether the eighth (R/$\overline{W}$) bit is 1. If there is a match, the slave considers that it has been addressed as a transmitter and generates acknowledge A3. The slave-transmitter remains addressed until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

After a repeated START condition (Sr), all other slave devices also compare the first seven bits of the first byte of the slave address with their own addresses and test the eighth ($R/\overline{W}$) bit. However, none of them are addressed because $R/\overline{W}$ = 1 (for 10-bit devices), or the 11110XX slave address (for 7-bit devices) does not match.

Table 37-4. Master-receiver addresses a slave-transmitter with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Sr	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 1	A3	Data	A	...	Data	A	P

After the master-receiver has sent the first byte of the 10-bit address, the slave-transmitter sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

37.4.3 Address matching

All received addresses can be requested in 7-bit or 10-bit address format.

- AD[7:1] in Address Register 1, which contains the I2C primary slave address, always participates in the address matching process. It provides a 7-bit address.
- If the ADEXT bit is set, AD[10:8] in Control Register 2 participates in the address matching process. It extends the I2C primary slave address to a 10-bit address.

Additional conditions that affect address matching include:

- If the GCAEN bit is set, general call participates the address matching process.
- If the ALERTEN bit is set, alert response participates the address matching process.
- If the SIICAEN bit is set, Address Register 2 participates in the address matching process.
- If the RMEN bit is set, when the Range Address register is programmed to a nonzero value, any address within the range of values of Address Register 1 (excluded) and the Range Address register (included) participates in the address matching process. The Range Address register must be programmed to a value greater than the value of Address Register 1.

When the I2C module responds to one of these addresses, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the Data register after the first byte transfer to determine that the address is matched.

37.4.4 System management bus specification

SMBus provides a control bus for system and power management related tasks. A system can use SMBus to pass messages to and from devices instead of tripping individual control lines.

Removing the individual control lines reduces pin count. Accepting messages ensures future expandability. With the system management bus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status.

37.4.4.1 Timeouts

The $T_{TIMEOUT,MIN}$ parameter allows a master or slave to conclude that a defective device is holding the clock low indefinitely or a master is intentionally trying to drive devices off the bus. The slave device must release the bus (stop driving the bus and let SCL and SDA float high) when it detects any single clock held low longer than $T_{TIMEOUT,MIN}$. Devices that have detected this condition must reset their communication and be able to receive a new START condition within the timeframe of $T_{TIMEOUT,MAX}$.

SMBus defines a clock low timeout, $T_{TIMEOUT}$, of 35 ms, specifies $T_{LOW:SEXT}$ as the cumulative clock low extend time for a slave device, and specifies $T_{LOW:MEXT}$ as the cumulative clock low extend time for a master device.

37.4.4.1.1 SCL low timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than a timeout value condition. Devices that have detected the timeout condition must reset the communication. When the I2C module is an active master, if it detects that SMBCLK low has exceeded the value of $T_{TIMEOUT,MIN}$, it must generate a stop condition within or after the current data byte in the transfer process. When the I2C module is a slave, if it detects the $T_{TIMEOUT,MIN}$ condition, it resets its communication and is then able to receive a new START condition.

37.4.4.1.2 SCL high timeout

When the I2C module has determined that the SMBCLK and SMBDAT signals have been high for at least $T_{HIGH:MAX}$, it assumes that the bus is idle.

A HIGH timeout occurs after a START condition appears on the bus but before a STOP condition appears on the bus. Any master detecting this scenario can assume the bus is free when either of the following occurs:

- SHTF1 rises.
- The BUSY bit is high and SHTF1 is high.

When the SMBDAT signal is low and the SMBCLK signal is high for a period of time, another kind of timeout occurs. The time period must be defined in software. SHTF2 is used as the flag when the time limit is reached. This flag is also an interrupt resource, so it triggers IICIF.

37.4.4.1.3 CSMBCLK TIMEOUT MEXT and CSMBCLK TIMEOUT SEXT

The following figure illustrates the definition of the timeout intervals $T_{LOW:SEXT}$ and $T_{LOW:MEXT}$. When in master mode, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:MEXT}$ within a byte, where each byte is defined as START-to-ACK, ACK-to-ACK, or ACK-to-STOP. When CSMBCLK TIMEOUT MEXT occurs, SMBus MEXT rises and also triggers the SLTF.

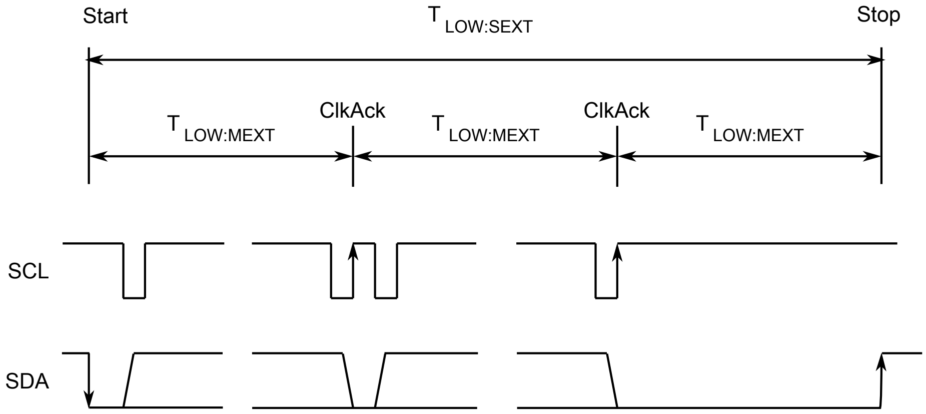


Figure 37-4. Timeout measurement intervals

A master is allowed to abort the transaction in progress to any slave that violates the $T_{LOW:SEXT}$ or $T_{TIMEOUT,MIN}$ specifications. To abort the transaction, the master issues a STOP condition at the conclusion of the byte transfer in progress. When a slave, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:SEXT}$ during any message from the initial START to the STOP. When CSMBCLK TIMEOUT SEXT occurs, SEXT rises and also triggers SLTF.

NOTE

CSMBCLK TIMEOUT SEXT and CSMBCLK TIMEOUT MEXT are optional functions that are implemented in the second step.

37.4.4.2 FAST ACK and NACK

To improve reliability and communication robustness, implementation of packet error checking (PEC) by SMBus devices is optional for SMBus devices but required for devices participating in and only during the address resolution protocol (ARP) process. The PEC is a CRC-8 error checking byte, calculated on all the message bytes. The PEC is appended to the message by the device that supplied the last data byte. If the PEC is present but not correct, a NACK is issued by the receiver. Otherwise an ACK is issued. To calculate the CRC-8 by software, this module can hold the SCL line low after receiving the eighth SCL (8th bit) if this byte is a data byte. So software can determine whether an ACK or NACK should be sent to the bus by setting or clearing the TXAK bit if the FACK (fast ACK/NACK enable) bit is enabled.

SMBus requires a device always to acknowledge its own address, as a mechanism to detect the presence of a removable device (such as a battery or docking station) on the bus. In addition to indicating a slave device busy condition, SMBus uses the NACK mechanism to indicate the reception of an invalid command or invalid data. Because such a condition may occur on the last byte of the transfer, SMBus devices are required to have the ability to generate the not acknowledge after the transfer of each byte and before the completion of the transaction. This requirement is important because SMBus does not provide any other resend signaling. This difference in the use of the NACK signaling has implications on the specific implementation of the SMBus port, especially in devices that handle critical system data such as the SMBus host and the SBS components.

NOTE

In the last byte of master receive slave transmit mode, the master must send a NACK to the bus, so FACK must be switched off before the last byte transmits.

37.4.5 Resets

The I2C module is disabled after a reset. The I2C module cannot cause a core reset.

37.4.6 Interrupts

The I2C module generates an interrupt when any of the events in the table found here occur, provided that the IICIE bit is set.

The interrupt is driven by the IICIF bit (of the I2C Status Register) and masked with the IICIE bit (of the I2C Control Register 1). The IICIF bit must be cleared (by software) by writing 1 to it in the interrupt routine. The SMBus timeouts interrupt is driven by SLTF and masked with the IICIE bit. The SLTF bit must be cleared by software by writing 1 to it in the interrupt routine. You can determine the interrupt type by reading the Status Register.

NOTE

In master receive mode, the FACK bit must be set to zero before the last byte transfer.

Table 37-5. Interrupt summary

Interrupt source	Status	Flag	Local enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration lost	ARBL	IICIF	IICIE
I^{2}C bus stop detection	STOPF	IICIF	IICIE & SSIE
I^{2}C bus start detection	STARTF	IICIF	IICIE & SSIE
SMBus SCL low timeout	SLTF	IICIF	IICIE
SMBus SCL high SDA low timeout	SHTF2	IICIF	IICIE & SHTF2IE
Wakeup from stop or wait mode	IAAS	IICIF	IICIE & WUEN

37.4.6.1 Byte transfer interrupt

The Transfer Complete Flag (TCF) bit is set at the falling edge of the ninth clock to indicate the completion of a byte and acknowledgement transfer. When FACK is enabled, TCF is then set at the falling edge of eighth clock to indicate the completion of byte.

37.4.6.2 Address detect interrupt

When the calling address matches the programmed slave address (I2C Address Register) or when the GCAEN bit is set and a general call is received, the IAAS bit in the Status Register is set. The CPU is interrupted, provided the IICIE bit is set. The CPU must check the SRW bit and set its Tx mode accordingly.

37.4.6.3 Stop Detect Interrupt

When the stop status is detected on the I^{2}C bus, the STOPF bit is set to 1. The CPU is interrupted, provided the IICIE and SSIE bits are both set to 1.

37.4.6.4 Exit from low-power/stop modes

The slave receive input detect circuit and address matching feature are still active on low power modes (wait and stop). An asynchronous input matching slave address or general call address brings the CPU out of low power/stop mode if the interrupt is not masked. Therefore, TCF and IAAS both can trigger this interrupt.

37.4.6.5 Arbitration lost interrupt

The I2C is a true multimaster bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, the relative priority of the contending masters is determined by a data arbitration procedure. The I2C module asserts the arbitration-lost interrupt when it loses the data arbitration process and the ARBL bit in the Status Register is set.

Arbitration is lost in the following circumstances:

1. SDA is sampled as low when the master drives high during an address or data transmit cycle.
2. SDA is sampled as low when the master drives high during the acknowledge bit of a data receive cycle.
3. A START cycle is attempted when the bus is busy.
4. A repeated START cycle is requested in slave mode.
5. A STOP condition is detected when the master did not request it.

The ARBL bit must be cleared (by software) by writing 1 to it.

37.4.6.6 Timeout interrupt in SMBus

When the IICIE bit is set, the I2C module asserts a timeout interrupt (outputs SLTF and SHTF2) upon detection of any of the mentioned timeout conditions, with one exception. The SCL high and SDA high TIMEOUT mechanism must not be used to influence the timeout interrupt output, because this timeout indicates an idle condition on the bus. SHTF1 rises when it matches the SCL high and SDA high TIMEOUT and falls automatically just to indicate the bus status. The SHTF2's timeout period is the same as that of SHTF1, which is short compared to that of SLTF, so another control bit, SHTF2IE, is added to enable or disable it.

37.4.7 Programmable input glitch filter

An I2C glitch filter has been added outside legacy I2C modules but within the I2C package. This filter can absorb glitches on the I2C clock and data lines for the I2C module.

The width of the glitch to absorb can be specified in terms of the number of (half) I2C module clock cycles. A single Programmable Input Glitch Filter control register is provided. Effectively, any down-up-down or up-down-up transition on the data line that occurs within the number of clock cycles programmed in this register is ignored by the I2C module. The programmer must specify the size of the glitch (in terms of I2C module clock cycles) for the filter to absorb and not pass.

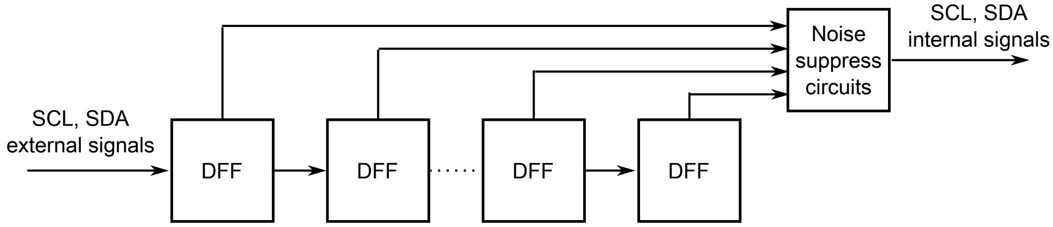


Figure 37-5. Programmable input glitch filter diagram

37.4.8 Address matching wake-up

When a primary, range, or general call address match occurs when the I2C module is in slave receive mode, the MCU wakes from a low power mode where no peripheral bus is running.

Data sent on the bus that is the same as a target device address might also wake the target MCU.

After the address matching IAAS bit is set, an interrupt is sent at the end of address matching to wake the core. The IAAS bit must be cleared after the clock recovery.

NOTE

After the system recovers and is in Run mode, restart the I2C module if it is needed to transfer packets. To avoid I2C transfer problems resulting from the situation, firmware should prevent the MCU execution of a STOP instruction when the I2C module is in the middle of a transfer unless the Stop mode holdoff feature is used during this period (set FLT[SHEN] to 1).

NOTE

After I2C address matching wake-up, the master must wait a time long enough for the slave ISR to finish running and resend start or repeat start signals.

For the SRW bit to function properly, it only supports Address +Write to wake up by I2C address matching. Before entering the next low power mode, Address+Write must be sent to change the SRW status.

37.4.9 DMA support

If the DMAEN bit is cleared and the IICIE bit is set, an interrupt condition generates an interrupt request.

If the DMAEN bit is set and the IICIE bit is set, an interrupt condition generates a DMA request instead. DMA requests are generated by the transfer complete flag (TCF).

If the DMAEN bit is set, only the TCF initiates a DMA request. All other events generate CPU interrupts.

NOTE

Before the last byte of master receive mode, TXAK must be set to send a NACK after the last byte's transfer. Therefore, the DMA must be disabled before the last byte's transfer.

NOTE

In 10-bit address mode transmission, the addresses to send occupy 2–3 bytes. During this transfer period, the DMA must be disabled because the C1 register is written to send a repeat start or to change the transfer direction.

37.4.10 Double buffering mode

In the double buffering mode, the data transfer is processed byte by byte. However, the data can be transferred without waiting for the interrupt or the polling to finish. This means the write/read I2C_D operation will not block the data transfer, as the hardware has already finished the internal write or read. The benefit is that the baud rate is able to achieve higher speed.

There are several items to consider as follows:

- When initiating a double buffering transfer at Tx side, the user can write 2 values to the I2C_D buffer before transfer. However, that is allowed only at one time per package frame (due to the buffer depth, and because two-times writes in each ISR are not allowed). The second write to the I2C_D buffer must wait for the Empty flag. On the other hand, at Rx side the user can read twice in a one-byte transfer (if needed).

NOTE

> Check Empty flag before write to I2C_D.
>
> Write twice to the I2C_D buffer ONLY after the address matching byte. Do not write twice (Address+Data) before START or at the beginning of I2C transfer, especially when the baud rate is very slow.

- To write twice in one frame, during the next-to-last ISR, do a dummy read from the I2C_D buffer at Tx side (or the TCF will stay high, because the TCF is cleared by write/read operation). In the next-to-last ISR, do not send data again (the buffer data will be under running).
- To keep new ISRs software-compatible with previous ISRs, the write/read I2C_D operation will not block the internal-hardware-released SCL/SDA signals. At the ACK phase, the bus is released to accept the next byte if the master can send the clock immediately.
- On the slave side, two-times writes to the I2C_D buffer may be limited by the master's clock and START/repeated-START signal. This is not currently supported, and the master's START/repeated-START signal will break data transfers. To release the bus, do a dummy read or write to the I2C_D buffer again. It is suggested to send repeated-START/START during intervals as before.
- The master receive should send a NACK in the next-to-last ISR, if it wants to do the STOP or the repeated-START work. The transmitting slave which receives the NACK, will switch to receive mode, and do a dummy read to release SCL and SDA signals.

37.5 Initialization/application information

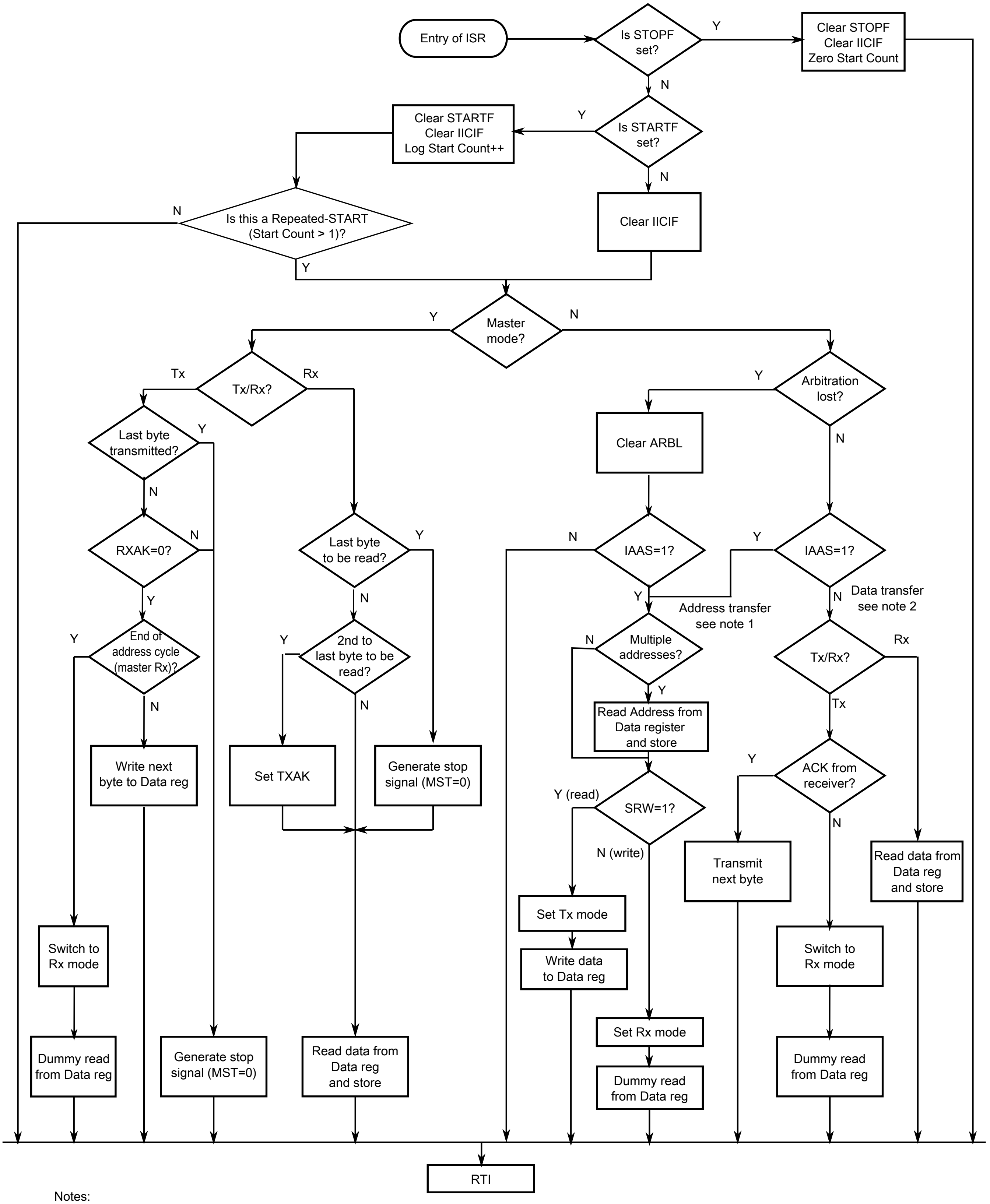
Module Initialization (Slave)

1. Write: Control Register 2
   - to enable or disable general call
   - to select 10-bit or 7-bit addressing mode
2. Write: Address Register 1 to set the slave address
3. Write: Control Register 1 to enable the I2C module and interrupts
4. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
5. Initialize RAM variables used to achieve the routine shown in the following figure

Module Initialization (Master)

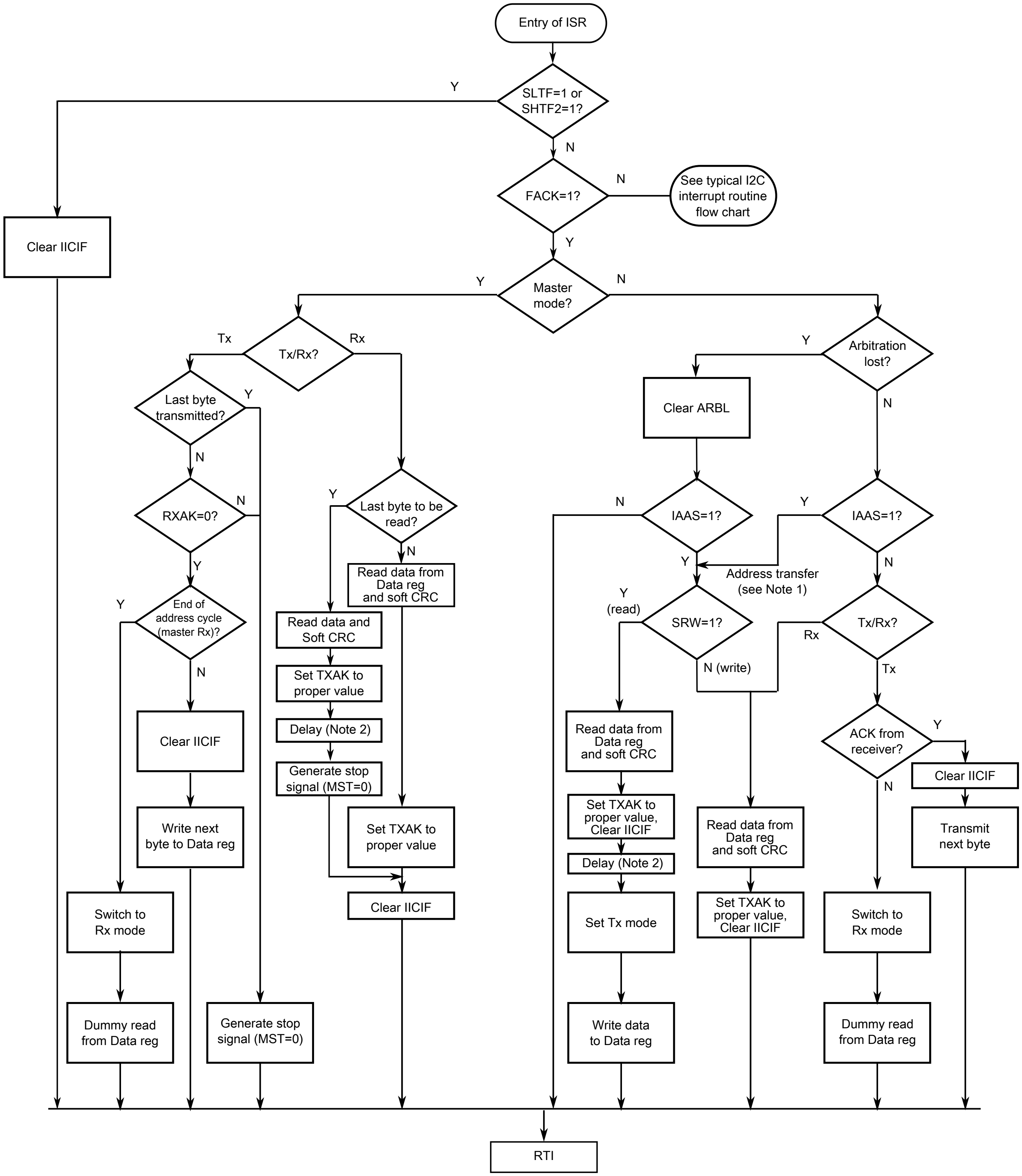
1. Write: Frequency Divider register to set the I2C baud rate (see example in description of ICR)
2. Write: Control Register 1 to enable the I2C module and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in the following figure
5. Write: Control Register 1 to enable TX
6. Write: Control Register 1 to enable MST (master mode)
7. Write: Data register with the address of the target slave (the LSB of this byte determines whether the communication is master receive or transmit)

The routine shown in the following figure encompasses both master and slave I2C operations. For slave operation, an incoming I2C message that contains the proper address begins I2C communication. For master operation, communication must be initiated by writing the Data register. An example of an I2C driver which implements many of the steps described here is available in AN4342: Using the Inter-Integrated Circuit on ColdFire+ and Kinetis .



Notes:
1. If general call is enabled, check to determine if the received address is a general call address (0x00).
If the received address is a general call address, the general call must be handled by user software.
2. When 10-bit addressing addresses a slave, the slave sees an interrupt following the first byte of the extended address.
Ensure that for this interrupt, the contents of the Data register are ignored and not treated as a valid data transfer.

Figure 37-6. Typical I2C interrupt routine



Notes:

1. If general call or SIICAEN is enabled, check to determine if the received address is a general call address (0x00) or an SMBus device default address. In either case, they must be handled by user software.
2. In receive mode, one bit time delay may be needed before the stop signal generation, to wait for the possible longest time period (in worst case) of the 9th SCL cycle.

Figure 37-7. Typical I2C SMBus interrupt routine

Chapter 38
Universal Asynchronous Receiver/Transmitter (UART0)

38.1 Introduction

38.1.1 Features

Features of the LPUART module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Programmable baud rates (13-bit modulo divider) with configurable oversampling ratio from 4x to 32x
- Transmit and receive baud rate can operate asynchronous to the bus clock:
    - Baud rate can be configured independently of the bus clock frequency
- Interrupt, DMA or polled operation:
    - Transmit data register empty and transmission complete
    - Receive data register full
    - Receive overrun, parity error, framing error, and noise error
    - Idle receiver detect
    - Active edge on receive pin
    - Break detect supporting LIN
    - Receive data match
- Hardware parity generation and checking
- Programmable 8-bit, 9-bit or 10-bit character length
- Programmable 1-bit or 2-bit stop bits
- Three receiver wakeup methods:
    - Idle line wakeup
    - Address mark wakeup
    - Receive data match
- Automatic address matching to reduce ISR overhead:
    - Address mark matching

- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output and receiver input polarity
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse width

38.1.2 Modes of operation

38.1.2.1 Wait mode

The LPUART can be configured to Stop in Wait modes, when the DOZEEN bit is set. The transmitter and receiver will finish transmitting/receiving the current word.

38.1.3 Signal Descriptions

Signal	Description	I/O
LPUART_TX	Transmit data. This pin is normally an output, but is an input (tristated) in single wire mode whenever the transmitter is disabled or transmit direction is configured for receive data.	I/O
LPUART_RX	Receive data.	I
LPUART_CTS	Clear to send.	I
LPUART_RTS	Request to send.	O

38.1.4 Block diagram

The following figure shows the transmitter portion of the LPUART.

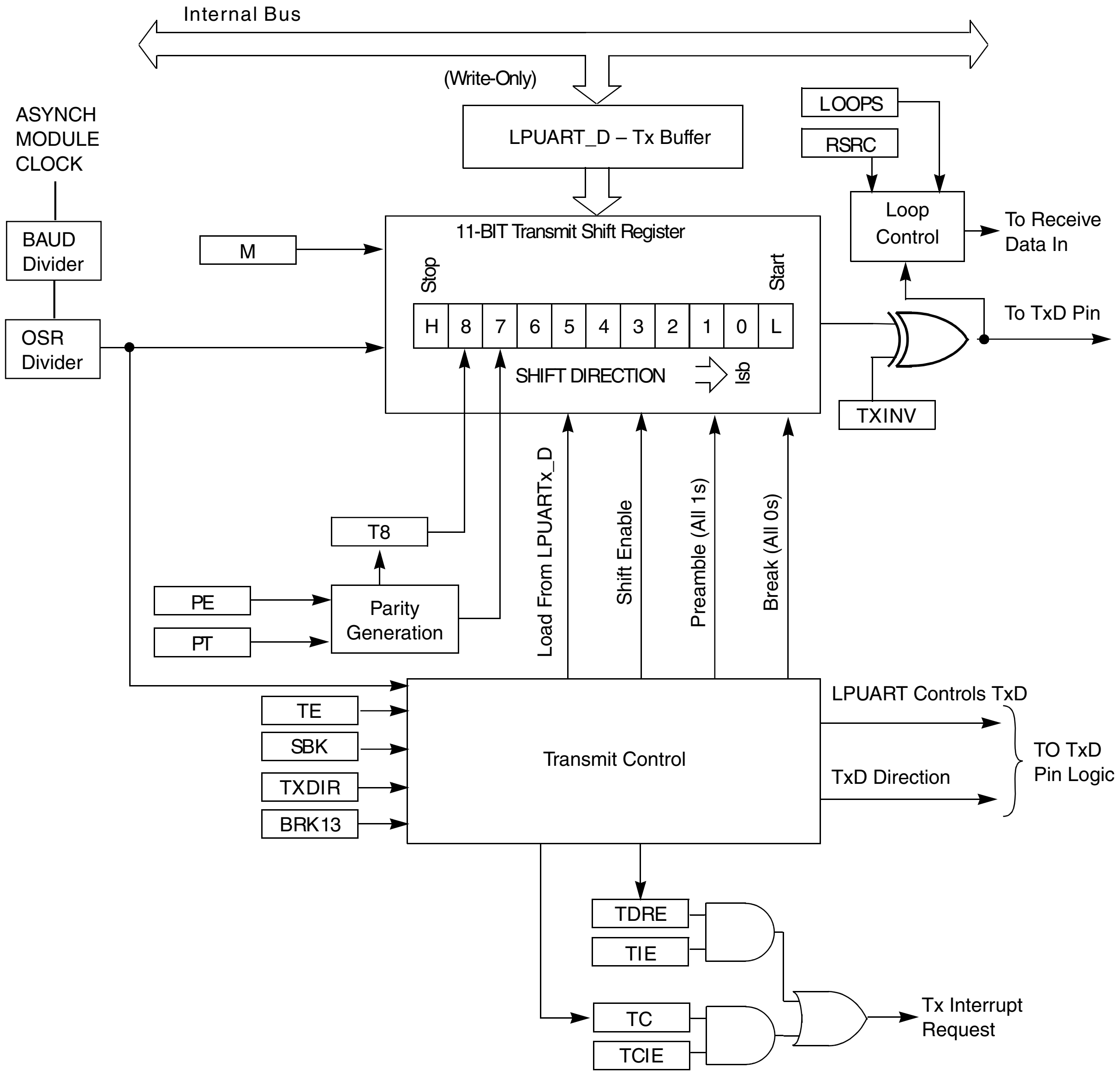


Figure 38-1. LPUART transmitter block diagram

The following figure shows the receiver portion of the LPUART.

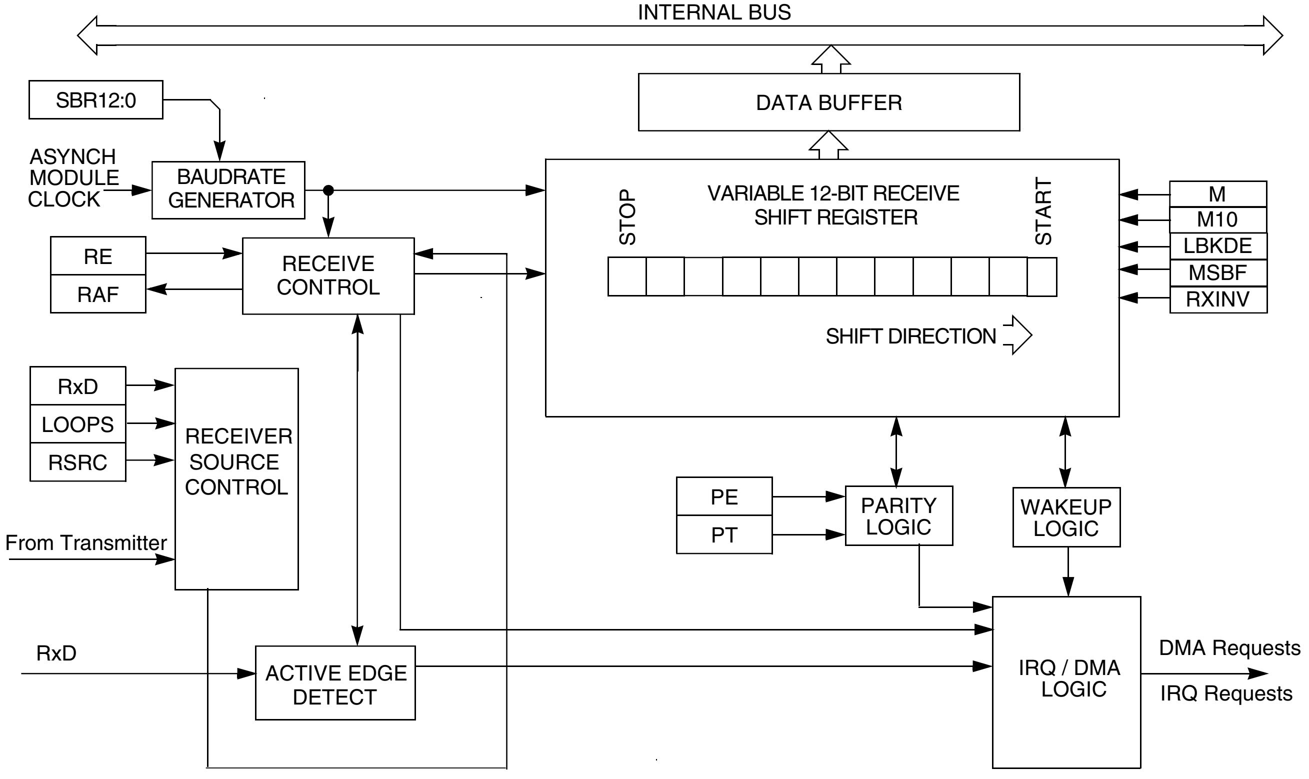


Figure 38-2. LPUART receiver block diagram

38.2 Register definition

The LPUART includes registers to control baud rate, select LPUART options, report LPUART status, and for transmit/receive data. Access to an address outside the valid memory map will generate a bus error.

LPUART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_4000	LPUART Baud Rate Register (LPUART0_BAUD)	32	R/W	0F00_0004h	38.2.1/769
4005_4004	LPUART Status Register (LPUART0_STAT)	32	R/W	00C0_0000h	38.2.2/771
4005_4008	LPUART Control Register (LPUART0_CTRL)	32	R/W	0000_0000h	38.2.3/775
4005_400C	LPUART Data Register (LPUART0_DATA)	32	R/W	0000_0000h	38.2.4/779
4005_4010	LPUART Match Address Register (LPUART0_MATCH)	32	R/W	0000_0000h	38.2.5/780
4005_4014	LPUART Modem IrDA Register (LPUART0_MODIR)	32	R/W	0000_0000h	38.2.6/781

38.2.1 LPUART Baud Rate Register (LPUARTx_BAUD)

Address: 4005_4000h base + 0h offset = 4005_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAEN1	MAEN2	M10	OSR					TDMAE	0	RDMAE	0	0		BOTHEDGE	RESYNCDIS
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LBKDIE	RXEDGIE	SBNS	SBR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

LPUARTx_BAUD field descriptions

Field	Description
31 MAEN1	Match Address Mode Enable 1 0 Normal operation. 1 Enables automatic address matching or data matching mode for MATCH[MA1].
30 MAEN2	Match Address Mode Enable 2 0 Normal operation. 1 Enables automatic address matching or data matching mode for MATCH[MA2].
29 M10	10-bit Mode select The M10 bit causes a tenth bit to be part of the serial transmission. This bit should only be changed when the transmitter and receiver are both disabled. 0 Receiver and transmitter use 8-bit or 9-bit data characters. 1 Receiver and transmitter use 10-bit data characters.
28–24 OSR	Oversampling Ratio This field configures the oversampling ratio for the receiver between 4x (00011) and 32x (11111). Writing an invalid oversampling ratio (for example, a value not between 4x and 32x) will default to an oversampling ratio of 16 (01111). The OSR field should only be changed when the transmitter and receiver are both disabled. Note that the oversampling ratio = OSR + 1.
23 TDMAE	Transmitter DMA Enable TDMAE configures the transmit data register empty flag, LPUART_STAT[TDRE], to generate a DMA request. 0 DMA request disabled. 1 DMA request enabled.

Table continues on the next page...

LPUARTx_BAUD field descriptions (continued)

Field	Description
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 RDMAE	Receiver Full DMA Enable RDMAE configures the receiver data register full flag, LPUART_STAT[RDRF], to generate a DMA request. 0 DMA request disabled. 1 DMA request enabled.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 BOTHEDGE	Both Edge Sampling Enables sampling of the received data on both edges of the baud rate clock, effectively doubling the number of times the receiver samples the input data for a given oversampling ratio. This bit must be set for oversampling ratios between x4 and x7 and is optional for higher oversampling ratios. This bit should only be changed when the receiver is disabled. 0 Receiver samples input data using the rising edge of the baud rate clock. 1 Receiver samples input data using the rising and falling edge of the baud rate clock.
16 RESYNCDIS	Resynchronization Disable When set, disables the resynchronization of the received data word when a data one followed by data zero transition is detected. This bit should only be changed when the receiver is disabled. 0 Resynchronization during received data word is supported 1 Resynchronization during received data word is disabled
15 LBKDIE	LIN Break Detect Interrupt Enable LBKDIE enables the LIN break detect flag, LBKDIF, to generate interrupt requests. 0 Hardware interrupts from LPUART_STAT[LBKDIF] disabled (use polling). 1 Hardware interrupt requested when LPUART_STAT[LBKDIF] flag is 1.
14 RXEDGIE	RX Input Active Edge Interrupt Enable Enables the receive input active edge, RXEDGIF, to generate interrupt requests. Changing CTRL[LOOP] or CTRL[RSRC] when RXEDGIE is set can cause the RXEDGIF to set. 0 Hardware interrupts from LPUART_STAT[RXEDGIF] disabled (use polling). 1 Hardware interrupt requested when LPUART_STAT[RXEDGIF] flag is 1.
13 SBNS	Stop Bit Number Select SBNS determines whether data characters are one or two stop bits. This bit should only be changed when the transmitter and receiver are both disabled. 0 One stop bit. 1 Two stop bits.
SBR	Baud Rate Modulo Divisor. The 13 bits in SBR[12:0] set the modulo divide rate for the baud rate generator. When SBR is 1 - 8191, the baud rate equals "baud clock / ((OSR+1) × SBR)". The 13-bit baud rate setting [SBR12:SBR0] must

Table continues on the next page...

LPUARTx_BAUD field descriptions (continued)

Field	Description
	only be updated when the transmitter and receiver are both disabled (LPUART_CTRL[RE] and LPUART_CTRL[TE] are both 0).

38.2.2 LPUART Status Register (LPUARTx_STAT)

Address: 4005_4000h base + 4h offset = 4005_4004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	w1c	w1c										w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_STAT field descriptions

Field	Description
31 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character has been detected. 1 LIN break character has been detected.
30 RXEDGIF	LPUART_RX Pin Active Edge Interrupt Flag

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
	RXEDGIF is set when an active edge, falling if RXINV = 0, rising if RXINV=1, on the LPUART_RX pin occurs. RXEDGIF is cleared by writing a 1 to it. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
29 MSBF	MSB First Setting this bit reverses the order of the bits that are transmitted and received on the wire. This bit does not affect the polarity of the bits, the location of the parity bit or the location of the start or stop bits. This bit should only be changed when the transmitter and receiver are both disabled. 0 LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0. 1 MSB (bit9, bit8, bit7 or bit6) is the first bit that is transmitted following the start bit depending on the setting of CTRL[M], CTRL[PE] and BAUD[M10]. Further, the first bit received after the start bit is identified as bit9, bit8, bit7 or bit6 depending on the setting of CTRL[M] and CTRL[PE].
28 RXINV	Receive Data Inversion Setting this bit reverses the polarity of the received data input. **NOTE:** Setting RXINV inverts the LPUART_RX input for all cases: data bits, start and stop bits, break, and idle. 0 Receive data not inverted. 1 Receive data inverted.
27 RWUID	Receive Wake Up Idle Detect For RWU on idle character, RWUID controls whether the idle character that wakes up the receiver sets the IDLE bit. For address match wakeup, RWUID controls if the IDLE bit is set when the address does not match. This bit should only be changed when the receiver is disabled. 0 During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character. During address match wakeup, the IDLE bit does not get set when an address does not match. 1 During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character. During address match wakeup, the IDLE bit does get set when an address does not match.
26 BRK13	Break Character Generation Length BRK13 selects a longer transmitted break character length. Detection of a framing error is not affected by the state of this bit. This bit should only be changed when the transmitter is disabled. 0 Break character is transmitted with length of 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 13 (if M10 = 1, SNBS = 1). 1 Break character is transmitted with length of 13 bit times (if M = 0, SBNS = 0) or 14 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 15 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 16 (if M10 = 1, SNBS = 1).
25 LBKDE	LIN Break Detection Enable LBKDE selects a longer break character detection length. While LBKDE is set, receive data is not stored in the receive data buffer.

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
	0 Break character is detected at length 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 13 (if M10 = 1, SNBS = 1). 1 Break character is detected at length of 11 bit times (if M = 0, SBNS = 0) or 12 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 14 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 15 (if M10 = 1, SNBS = 1).
24 RAF	Receiver Active Flag RAF is set when the receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. 0 LPUART receiver idle waiting for a start bit. 1 LPUART receiver active (LPUART_RX input not idle).
23 TDRE	Transmit Data Register Empty Flag When the transmit FIFO is enabled, TDRE will set when the number of datawords in the transmit FIFO (LPUART_DATA) is equal to or less than the number indicated by LPUART_WATER[TXWATER]). To clear TDRE, write to the LPUART data register (LPUART_DATA) until the number of words in the transmit FIFO is greater than the number indicated by LPUART_WATER[TXWATER]. When the transmit FIFO is disabled,TDRE will set when the transmit data register (LPUART_DATA) is empty. To clear TDRE, write to the LPUART data register (LPUART_DATA). TDRE is not affected by a character that is in the process of being transmitted, it is updated at the start of each transmitted character. 0 Transmit data buffer full. 1 Transmit data buffer empty.
22 TC	Transmission Complete Flag TC is cleared when there is a transmission in progress or when a preamble or break character is loaded. TC is set when the transmit buffer is empty and no data, preamble, or break character is being transmitted. When TC is set, the transmit data output signal becomes idle (logic 1). TC is cleared by writing to LPUART_DATA to transmit new data, queuing a preamble by clearing and then setting LPUART_CTRL[TE], queuing a break character by writing 1 to LPUART_CTRL[SBK]. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
21 RDRF	Receive Data Register Full Flag When the receive FIFO is enabled, RDRF is set when the number of datawords in the receive buffer is greater than the number indicated by LPUART_WATER[RXWATER]. To clear RDRF, read LPUART_DATA until the number of datawords in the receive data buffer is equal to or less than the number indicated by LPUART_WATER[RXWATER]. When the receive FIFO is disabled, RDRF is set when the receive buffer (LPUART_DATA) is full. To clear RDRF, read the LPUART_DATA register. A character that is in the process of being received does not cause a change in RDRF until the entire character is received. Even if RDRF is set, the character will continue to be received until an overrun condition occurs once the entire character is received. 0 Receive data buffer empty. 1 Receive data buffer full.
20 IDLE	Idle Line Flag IDLE is set when the LPUART receive line becomes idle for a full character time after a period of activity. When ILT is cleared, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bits time count toward the full character time of logic high, 10 to 13 bit times, needed for the receiver to detect an idle line. When ILT is set, the receiver doesn't start counting

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
	idle bit times until after the stop bits. The stop bits and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, write logic 1 to the IDLE flag. After IDLE has been cleared, it cannot become set again until after a new character has been stored in the receive buffer or a LIN break character has set the LBKDIF flag . IDLE is set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
19 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If LBKDE is enabled and a LIN Break is detected, the OR field asserts if LBKDIF is not cleared before the next data character is received. While the OR flag is set, no additional data is stored in the data buffer even if sufficient room exists. To clear OR, write logic 1 to the OR flag. 0 No overrun. 1 Receive overrun (new LPUART data lost).
18 NF	Noise Flag The advanced sampling technique used in the receiver takes three samples in each of the received bits. If any of these samples disagrees with the rest of the samples within any bit time in the frame then noise is detected for that character. NF is set whenever the next character to be read from LPUART_DATA was received with noise detected within the character. To clear NF, write logic one to the NF. 0 No noise detected. 1 Noise detected in the received character in LPUART_DATA.
17 FE	Framing Error Flag FE is set whenever the next character to be read from LPUART_DATA was received with logic 0 detected where a stop bit was expected. To clear FE, write logic one to the FE. 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
16 PF	Parity Error Flag PF is set whenever the next character to be read from LPUART_DATA was received when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, write a logic one to the PF. 0 No parity error. 1 Parity error.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

38.2.3 LPUART Control Register (LPUARTx_CTRL)

This read/write register controls various optional features of the LPUART system. This register should only be altered when the transmitter and receiver are both disabled.

Address: 4005_4000h base + 8h offset = 4005_4008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_CTRL field descriptions

Field	Description
31 R8T9	Receive Bit 8 / Transmit Bit 9 R8 is the ninth data bit received when the LPUART is configured for 9-bit or 10-bit data formats. When reading 9-bit or 10-bit data, read R8 before reading LPUART_DATA. T9 is the tenth data bit received when the LPUART is configured for 10-bit data formats. When writing 10-bit data, write T9 before writing LPUART_DATA. If T9 does not need to change from its previous value, such as when it is used to generate address mark or parity, they it need not be written each time LPUART_DATA is written.
30 R9T8	Receive Bit 9 / Transmit Bit 8 R9 is the tenth data bit received when the LPUART is configured for 10-bit data formats. When reading 10-bit data, read R9 before reading LPUART_DATA T8 is the ninth data bit received when the LPUART is configured for 9-bit or 10-bit data formats. When writing 9-bit or 10-bit data, write T8 before writing LPUART_DATA. If T8 does not need to change from its previous value, such as when it is used to generate address mark or parity, they it need not be written each time LPUART_DATA is written.
29 TXDIR	LPUART_TX Pin Direction in Single-Wire Mode When the LPUART is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the LPUART_TX pin. When clearing TXDIR, the transmitter will finish receiving the current character (if any) before the receiver starts receiving data from the LPUART_TX pin. 0 LPUART_TX pin is an input in single-wire mode. 1 LPUART_TX pin is an output in single-wire mode.

Table continues on the next page...

LPUARTx_CTRL field descriptions (continued)

Field	Description
28 TXINV	Transmit Data Inversion Setting this bit reverses the polarity of the transmitted data output. **NOTE:** Setting TXINV inverts the LPUART_TX output for all cases: data bits, start and stop bits, break, and idle. 0 Transmit data not inverted. 1 Transmit data inverted.
27 ORIE	Overrun Interrupt Enable This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled; use polling. 1 Hardware interrupt requested when OR is set.
26 NEIE	Noise Error Interrupt Enable This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled; use polling. 1 Hardware interrupt requested when NF is set.
25 FEIE	Framing Error Interrupt Enable This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled; use polling. 1 Hardware interrupt requested when FE is set.
24 PEIE	Parity Error Interrupt Enable This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled; use polling). 1 Hardware interrupt requested when PF is set.
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests. 0 Hardware interrupts from TDRE disabled; use polling. 1 Hardware interrupt requested when TDRE flag is 1.
22 TCIE	Transmission Complete Interrupt Enable for TCIE enables the transmission complete flag, TC, to generate interrupt requests. 0 Hardware interrupts from TC disabled; use polling. 1 Hardware interrupt requested when TC flag is 1.
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate interrupt requests. 0 Hardware interrupts from RDRF disabled; use polling. 1 Hardware interrupt requested when RDRF flag is 1.

Table continues on the next page...

LPUARTx_CTRL field descriptions (continued)

Field	Description
20 ILIE	Idle Line Interrupt Enable ILIE enables the idle line flag, STAT[IDLE], to generate interrupt requests. 0 Hardware interrupts from IDLE disabled; use polling. 1 Hardware interrupt requested when IDLE flag is 1.
19 TE	Transmitter Enable Enables the LPUART transmitter. TE can also be used to queue an idle preamble by clearing and then setting TE. When TE is cleared, this register bit will read as 1 until the transmitter has completed the current character and the LPUART_TX pin is tristated. 0 Transmitter disabled. 1 Transmitter enabled.
18 RE	Receiver Enable Enables the LPUART receiver. When RE is written to 0, this register bit will read as 1 until the receiver finishes receiving the current character (if any). 0 Receiver disabled. 1 Receiver enabled.
17 RWU	Receiver Wakeup Control This field can be set to place the LPUART receiver in a standby state. RWU automatically clears when an RWU event occurs, that is, an IDLE event when CTRL[WAKE] is clear or an address match when CTRL[WAKE] is set with STAT[RWUID] is clear. **NOTE:** RWU must be set only with CTRL[WAKE] = 0 (wakeup on idle) if the channel is currently not idle. This can be determined by STAT[RAF]. If the flag is set to wake up an IDLE event and the channel is already idle, it is possible that the LPUART will discard data. This is because the data must be received or a LIN break detected after an IDLE is detected before IDLE is allowed to be reasserted. 0 Normal receiver operation. 1 LPUART receiver in standby waiting for wakeup condition.
16 SBK	Send Break Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 to 13, or 13 to 16 if LPUART_STATBRK13] is set, bit times of logic 0 are queued as long as SBK is set. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 LOOPS	Loop Mode Select When LOOPS is set, the LPUART_RX pin is disconnected from the LPUART and the transmitter output is internally connected to the receiver input. The transmitter and the receiver must be enabled to use the loop function.

Table continues on the next page...

LPUARTx_CTRL field descriptions (continued)

Field	Description
	0 Normal operation - LPUART_RX and LPUART_TX use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input (see RSRC bit).
6 DOZEEN	Doze Enable 0 LPUART is enabled in Doze mode. 1 LPUART is disabled in Doze mode.
5 RSRC	Receiver Source Select This field has no meaning or effect unless the LOOPS field is set. When LOOPS is set, the RSRC field determines the source for the receiver shift register input. 0 Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the LPUART does not use the LPUART_RX pin. 1 Single-wire LPUART mode where the LPUART_TX pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select 0 Receiver and transmitter use 8-bit data characters. 1 Receiver and transmitter use 9-bit data characters.
3 WAKE	Receiver Wakeup Method Select Determines which condition wakes the LPUART when RWU=1: • Address mark in the most significant bit position of a received data character, or • An idle condition on the receive pin input signal. 0 Configures RWU for idle-line wakeup. 1 Configures RWU with address-mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** In case the LPUART is programmed with ILT = 1, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables hardware parity generation and checking. When parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	Parity Type Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even.

Table continues on the next page...

LPUARTx_CTRL field descriptions (continued)

Field	Description
	0 Even parity. 1 Odd parity.

38.2.4 LPUART Data Register (LPUARTx_DATA)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for some of the LPUART status flags.

Address: 4005_4000h base + Ch offset = 4005_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOISY	PARITYE	FRETSC	0			R9T9	R8T8	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_DATA field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 NOISY	The current received dataword contained in DATA[R9:R0] was received with noise. 0 The dataword was received without noise. 1 The data was received with noise.
14 PARITYE	The current received dataword contained in DATA[R9:R0] was received with a parity error. 0 The dataword was received without a parity error. 1 The dataword was received with a parity error.
13 FRETSC	Frame Error The current received dataword contained in DATA[R9:R0] was received with a frame error.
12–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 R9T9	Read receive data buffer 9 or write transmit data buffer 9.
8 R8T8	Read receive data buffer 8 or write transmit data buffer 8.
7 R7T7	Read receive data buffer 7 or write transmit data buffer 7.
6 R6T6	Read receive data buffer 6 or write transmit data buffer 6.
5 R5T5	Read receive data buffer 5 or write transmit data buffer 5.
4 R4T4	Read receive data buffer 4 or write transmit data buffer 4.
3 R3T3	Read receive data buffer 3 or write transmit data buffer 3.
2 R2T2	Read receive data buffer 2 or write transmit data buffer 2.
1 R1T1	Read receive data buffer 1 or write transmit data buffer 1.
0 R0T0	Read receive data buffer 0 or write transmit data buffer 0.

38.2.5 LPUART Match Address Register (LPUARTx_MATCH)

Address: 4005_4000h base + 10h offset = 4005_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MA2								0								MA1							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_MATCH field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 MA2	Match Address 2 The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated BAUD[MAEN] bit is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded. Software should only write a MA register when the associated BAUD[MAEN] bit is clear.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MA1	Match Address 1 The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated BAUD[MAEN] bit is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded. Software should only write a MA register when the associated BAUD[MAEN] bit is clear.

38.2.6 LPUART Modem IrDA Register (LPUARTx_MODIR)

The MODEM register controls options for setting the modem configuration.

Address: 4005_4000h base + 14h offset = 4005_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													IREN	TNP	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_MODIR field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 IREN	Infrared enable Enables/disables the infrared modulation/demodulation.

Table continues on the next page...

LPUARTx_MODIR field descriptions (continued)

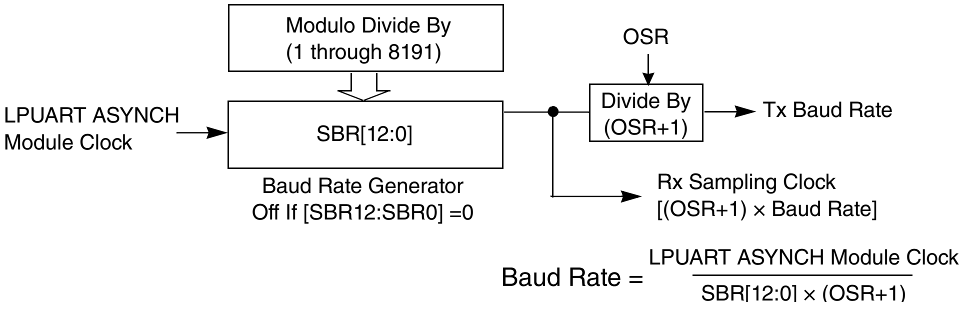
Field	Description
	0 IR disabled. 1 IR enabled.
17–16 TNP	Transmitter narrow pulse Enables whether the LPUART transmits a 1/OSR, 2/OSR, 3/OSR or 4/OSR narrow pulse. 00 1/OSR. 01 2/OSR. 10 3/OSR. 11 4/OSR.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RXRTSE	Receiver request-to-send enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. **NOTE:** Do not set both RXRTSE and TXRTSE. 0 The receiver has no effect on RTS. 1 RTS assertion is configured by the RTSWATER field
2 TXRTSPOL	Transmitter request-to-send polarity Controls the polarity of the transmitter RTS. TXRTSPOL does not affect the polarity of the receiver RTS. RTS will remain negated in the active low state unless TXRTSE is set. 0 Transmitter RTS is active low. 1 Transmitter RTS is active high.
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission. 0 The transmitter has no effect on RTS. 1 When a character is placed into an empty transmitter data buffer , RTS asserts one bit time before the start bit is transmitted. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit.
0 TXCTSE	Transmitter clear-to-send enable TXCTSE controls the operation of the transmitter. TXCTSE can be set independently from the state of TXRTSE and RXRTSE. 0 CTS has no effect on the transmitter. 1 Enables clear-to-send operation. The transmitter checks the state of CTS each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the signal TXD remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS as a character is being sent do not affect its transmission.

38.3 Functional description

The LPUART supports full-duplex, asynchronous, NRZ serial communication and comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. The following describes each of the blocks of the LPUART.

38.3.1 Baud rate generation

A 13-bit modulus counter in the baud rate generator derive the baud rate for both the receiver and the transmitter. The value from 1 to 8191 written to SBR[12:0] determines the baud clock divisor for the asynchronous LPUART baud clock. The SBR bits are in the LPUART baud rate registers, BDH and BDL. The baud rate clock drives the receiver, while the transmitter is driven by the baud rate clock divided by the over sampling ratio. Depending on the over sampling ratio, the receiver has an acquisition rate of 4 to 32 samples per bit time.



$$\text{Baud Rate} = \frac{\text{LPUART ASYNCH Module Clock}}{\text{SBR[12:0]} \times (\text{OSR}+1)}$$

Figure 38-3. LPUART baud rate generation

Baud rate generation is subject to two sources of error:

- Integer division of the asynchronous LPUART baud clock may not give the exact target frequency.
- Synchronization with the asynchronous LPUART baud clock can cause phase shift.

38.3.2 Transmitter functional description

This section describes the overall block diagram for the LPUART transmitter, as well as specialized functions for sending break and idle characters.

The transmitter output (LPUART_TX) idle state defaults to logic high, CTRL[TXINV] is cleared following reset. The transmitter output is inverted by setting CTRL[TXINV]. The transmitter is enabled by setting the CTRL[TE] bit. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the LPUART data register.

The central element of the LPUART transmitter is the transmit shift register that is 10-bit to 13 bits long depending on the setting in the CTRL[M], BAUD[M10] and BAUD[SBNS] control bits. For the remainder of this section, assume CTRL[M], BAUD[M10] and BAUD[SBNS] are cleared, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new character, the value waiting in the transmit data register is transferred to the shift register, synchronized with the baud rate clock, and the transmit data register empty (STAT[TDRE]) status flag is set to indicate another character may be written to the transmit data buffer at LPUART_DATA.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the LPUART_TX pin, the transmitter sets the transmit complete flag and enters an idle mode, with LPUART_TX high, waiting for more characters to transmit.

Writing 0 to CTRL[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed (that could include a data character, idle character or break character), although the transmitter will not start transmitting another character.

38.3.2.1 Send break and queued idle

The LPUART_CTRL[SBK] bit sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 10-bit to 12-bit times including the start and stop bits. A longer break of 13-bit times can be enabled by setting LPUART_STAT[BRK13]. Normally, a program would wait for LPUART_STAT[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to the LPUART_CTRL[SBK] bit. This action queues a break character to be sent as soon as the shifter is available. If LPUART_CTRL[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. If the receiving device is another Freescale Semiconductor LPUART, the break characters are received as 0s in all data bits and a framing error (LPUART_STAT[FE] = 1) occurs.

A break character can also be transmitted by writing to the LPUART_DATA register with bit 13 set and the data bits clear. This supports transmitting the break character as part of the normal data stream and also allows the DMA to transmit a break character.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for LPUART_STAT[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the LPUART_CTRL[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while LPUART_CTRL[TE] is cleared, the LPUART transmitter never actually releases control of the LPUART_TX pin.

An idle character can also be transmitted by writing to the LPUART_DATA register with bit 13 set and the data bits also set. This supports transmitting the idle character as part of the normal data stream and also allows the DMA to transmit a break character.

The length of the break character is affected by the LPUART_STAT[BRK13], LPUART_CTRL[M], LPUART_BAUD[M10] and LPUART_BAUD[SNBS] bits as shown below.

Table 38-1. Break character length

BRK13	M	M10	SBNS	Break character length
0	0	0	0	10 bit times
0	0	0	1	11 bit times
0	1	0	0	11 bit times
0	1	0	1	12 bit times
0	X	1	0	12 bit times
0	X	1	1	13 bit times
1	0	0	0	13 bit times
1	0	0	1	13 bit times
1	1	0	0	14 bit times
1	1	0	1	14 bit times
1	X	1	0	15 bit times
1	X	1	1	15 bit times

38.3.2.2 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS. If the clear-to-send operation is enabled, the character is transmitted when CTS is asserted. If CTS is deasserted in the middle of a transmission with characters remaining in the receiver data buffer, the character in the shift register is sent and LPUART_TX remains in the mark state until CTS is reasserted.

If the clear-to-send operation is disabled, the transmitter ignores the state of CTS.

The transmitter's CTS signal can also be enabled even if the same LPUART receiver's RTS signal is disabled.

38.3.2.3 Transceiver driver enable

The transmitter can use LPUART_RTS as an enable signal for the driver of an external transceiver. See Transceiver driver enable using LPUART_RTS for details. If the request-to-send operation is enabled, when a character is placed into an empty transmitter data buffer, LPUART_RTS asserts one bit time before the start bit is transmitted. LPUART_RTS remains asserted for the whole time that the transmitter data buffer has any characters. LPUART_RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. Transmitting a break character also asserts LPUART_RTS, with the same assertion and deassertion timing as having a character in the transmitter data buffer.

The transmitter's LPUART_RTS signal asserts only when the transmitter is enabled. However, the transmitter's LPUART_RTS signal is unaffected by its LPUART_CTS signal. LPUART_RTS will remain asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

38.3.3 Receiver functional description

In this section, the receiver block diagram is a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, different variations of the receiver wakeup function are explained.

The receiver input is inverted by setting LPUART_STAT[RXINV]. The receiver is enabled by setting the LPUART_CTRL[RE] bit. Character frames consist of a start bit of logic 0, eight to ten data bits (msb or lsb first), and one or two stop bits of logic 1. For information about 9-bit or 10-bit data mode, refer to 8-bit, 9-bit and 10-bit data modes. For the remainder of this discussion, assume the LPUART is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (LPUART_STAT[RDRF]) status flag is set. If LPUART_STAT[RDRF] was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the LPUART receiver is double-buffered, the program has one full character time after LPUART_STAT[RDRF] is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (LPUART_STAT[RDRF] = 1), it gets the data from the receive data register by reading LPUART_DATA. Refer to Interrupts and status flags for details about flag clearing.

38.3.3.1 Data sampling technique

The LPUART receiver supports a configurable oversampling rate of between 4× and 32× of the baud rate clock for sampling. The receiver starts by taking logic level samples at the oversampling rate times the baud rate to search for a falling edge on the LPUART_RX serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is located, three more samples are taken at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a received character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts the sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, the noise flag (LPUART_STAT[NF]) is set when the received character is transferred to the receive data buffer.

When the LPUART receiver is configured to sample on both edges of the baud rate clock, the number of segments in each received bit is effectively doubled (from 1 to OSR×2). The start and data bits are then sampled at OSR, OSR+1 and OSR+2. Sampling on both edges of the clock must be enabled for oversampling rates of 4× to 7× and is optional for higher oversampling rates.

The falling edge detection logic continuously looks for falling edges. If an edge is detected, the sample clock is resynchronized to bit times (unless resynchronization has been disabled). This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

38.3.3.2 Receiver wakeup operation

Receiver wakeup and receiver address matching is a hardware mechanism that allows an LPUART receiver to ignore the characters in a message intended for a different receiver.

During receiver wakeup, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up control bit (LPUART_CTRL[RWU]). When RWU bit and LPUART_S2[RWUID] bit are set, the status flags associated with the receiver, with the exception of the idle bit, IDLE, are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force LPUART_CTRL[RWU] to 0 so all receivers wake up in time to look at the first character(s) of the next message.

During receiver address matching, the address matching is performed in hardware and the LPUART receiver will ignore all characters that do not meet the address match requirements.

Table 38-2. Receiver Wakeup Options

RWU	MA1 | MA2	WAKE:RWUID	Receiver Wakeup
0	0	X	Normal operation
1	0	00	Receiver wakeup on idle line, IDLE flag not set
1	0	01	Receiver wakeup on idle line, IDLE flag set

Table continues on the next page...

Table 38-2. Receiver Wakeup Options (continued)

RWU	MA1 | MA2	WAKE:RWUID	Receiver Wakeup
1	0	10	Receiver wakeup on address mark
0	1	X0	Address mark address match, IDLE flag not set for discarded characters
0	1	X1	Address mark address match, IDLE flag set for discarded characters

38.3.3.2.1 Idle-line wakeup

When wake is cleared, the receiver is configured for idle-line wakeup. In this mode, LPUART_CTRL[RWU] is cleared automatically when the receiver detects a full character time of the idle-line level. The LPUART_CTRL[M] and LPUART_BAUD[M10] control bit selects 8-bit to 10-bit data mode and the LPUART_BAUD[SBNS] bit selects 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 10 to 13 bit times because of the start and stop bits.

When LPUART_CTRL[RWU] is one and LPUART_STAT[RWUID] is zero, the idle condition that wakes up the receiver does not set the LPUART_STAT[IDLE] flag. The receiver wakes up and waits for the first data character of the next message that sets the LPUART_STAT[RDRF] flag and generates an interrupt if enabled. When LPUART_STAT[RWUID] is one, any idle condition sets the LPUART_STAT[IDLE] flag and generates an interrupt if enabled, regardless of whether LPUART_CTRL[RWU] is zero or one.

The idle-line type (LPUART_CTRL[ILT]) control bit selects one of two ways to detect an idle line. When LPUART_CTRL[ILT] is cleared, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When LPUART_CTRL[ILT] is set, the idle bit counter does not start until after the stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

38.3.3.2.2 Address-mark wakeup

When LPUART_CTRL[WAKE] is set, the receiver is configured for address-mark wakeup. In this mode, LPUART_CTRL[RWU] is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character.

Address-mark wakeup allows messages to contain idle characters, but requires the MSB be reserved for use in address frames. The logic 1 in the MSB of an address frame clears the LPUART_CTRL[RWU] bit before the stop bits are received and sets the LPUART_STAT[RDRF] flag. In this case, the character with the MSB set is received even though the receiver was sleeping during most of this character time.

38.3.3.2.3 Address Match operation

Address match operation is enabled when the LPUART_BAUD[MAEN1] or LPUART_BAUD[MAEN2] bit is set and LPUART_BAUD[MATCFG] is equal to 00. In this function, a character received by the LPUART_RX pin with a logic 1 in the bit position immediately preceding the stop bit is considered an address and is compared with the associated MATCH[MA1] or MATCH[MA2] field. The character is only transferred to the receive buffer, and LPUART_STAT[RDRF] is set, if the comparison matches. All subsequent characters received with a logic 0 in the bit position immediately preceding the stop bit are considered to be data associated with the address and are transferred to the receive data buffer. If no marked address match occurs then no transfer is made to the receive data buffer, and all following characters with logic zero in the bit position immediately preceding the stop bit are also discarded. If both the LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] bits are negated, the receiver operates normally and all data received is transferred to the receive data buffer.

Address match operation functions in the same way for both MATCH[MA1] and MATCH[MA2] fields.

- If only one of LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] is asserted, a marked address is compared only with the associated match register and data is transferred to the receive data buffer only on a match.
- If LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either register.

38.3.3.3 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert LPUART_RTS.

- LPUART_RTS remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using LPUART_RTS for more details.

- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts LPUART_RTS if the number of characters in the receiver data register is full or a start bit is detected that will cause the receiver data register to be full.
- The receiver asserts LPUART_RTS when the number of characters in the receiver data register is not full and has not detected a start bit that will cause the receiver data register to be full. It is not affected if STAT[RDRF] is asserted.
- Even if LPUART_RTS is deasserted, the receiver continues to receive characters until the receiver data buffer is overrun.
- If the receiver request-to-send functionality is disabled, the receiver LPUART_RTS remains deasserted.

38.3.3.4 Infrared decoder

The infrared decoder converts the received character from the IrDA format to the NRZ format used by the receiver. It also has a OSR oversampling baud rate clock counter that filters noise and indicates when a 1 is received.

38.3.3.4.1 Start bit detection

When STAT[RXINV] is cleared, the first falling edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently from each other.

38.3.3.4.2 Noise filtering

Any further rising edges detected during the first half of the infrared decoder counter are ignored by the decoder. Any pulses less than one oversampling baud clock can be undetected by it regardless of whether it is seen in the first or second half of the count.

38.3.3.4.3 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as a 0, which is sent to the receiver. The decoder counter is also reset.

38.3.3.4.4 High-bit detection

At OSR oversampling baud rate clocks after the previous rising edge, if a rising edge is not seen, then the decoder sends a 1 to the receiver.

If the next bit is a 0, which arrives late, then a low-bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to a 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, then the delay of a 0 is not recorded as noise.

38.3.4 Additional LPUART functions

The following sections describe additional LPUART functions.

38.3.4.1 8-bit, 9-bit and 10-bit data modes

The LPUART transmitter and receiver can be configured to operate in 9-bit data mode by setting the LPUART_CTRL[M] or 10-bit data mode by setting LPUART_CTRL[M10]. In 9-bit mode, there is a ninth data bit in 10-bit mode there is a tenth data bit. For the transmit data buffer, these bits are stored in LPUART_CTRL[T8] and LPUART_CTRL[T9]. For the receiver, these bits are held in LPUART_CTRL[R8] and LPUART_CTRL[R9]. They are also accessible via 16-bit or 32-bit accesses to the LPUART_DATA register.

For coherent 8-bit writes to the transmit data buffer, write to LPUART_CTRL[T8] and LPUART_CTRL[T9] before writing to LPUART_DATA[7:0]. For 16-bit and 32-bit writes to the LPUART_DATA register all 10 transmit bits are written to the transmit data buffer at the same time.

If the bit values to be transmitted as the ninth and tenth bit of a new character are the same as for the previous character, it is not necessary to write to LPUART_CTRL[T8] and LPUART_CTRL[T9] again. When data is transferred from the transmit data buffer to the transmit shifter, the value in LPUART_CTRL[T8] and LPUART_CTRL[T9] is copied at the same time data is transferred from LPUART_DATA[7:0] to the shifter.

The 9-bit data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with address-mark wakeup so the ninth data bit can serve as the wakeup bit. The 10-bit data mode is typically used with parity and address-mark wakeup so the ninth data bit can serve as the wakeup bit and the tenth bit as the parity bit. In custom protocols, the ninth and/or tenth bits can also serve as software-controlled markers.

38.3.4.2 Idle length

An idle character is a character where the start bit, all data bits and stop bits are in the mark postion. The CTRL[ILT] register can be configured to start detecting an idle character from the previous start bit (any data bits and stop bits count towards the idle character detection) or from the previous stop bit.

38.3.4.3 Loop mode

When LPUART_CTRL[LOOPS] is set, the LPUART_CTRL[RSRC] bit in the same register chooses between loop mode (LPUART_CTRL[RSRC] = 0) or single-wire mode (LPUART_CTRL[RSRC] = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the LPUART_RX pin is not used by the LPUART.

38.3.4.4 Single-wire operation

When LPUART_CTRL[LOOPS] is set, the RSRC bit in the same register chooses between loop mode (LPUART_CTRL[RSRC] = 0) or single-wire mode (LPUART_CTRL[RSRC] = 1). Single-wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the LPUART_TX pin (the LPUART_RX pin is not used).

In single-wire mode, the LPUART_CTRL[TXDIR] bit controls the direction of serial data on the LPUART_TX pin. When LPUART_CTRL[TXDIR] is cleared, the LPUART_TX pin is an input to the receiver and the transmitter is temporarily disconnected from the LPUART_TX pin so an external device can send serial data to the receiver. When LPUART_CTRL[TXDIR] is set, the LPUART_TX pin is an output driven by the transmitter, the internal loop back connection is disabled, and as a result the receiver cannot receive characters that are sent out by the transmitter.

38.3.5 Infrared interface

The LPUART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the LPUART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. This design covers data rates only between 2.4 kbits/s and 115.2 kbits/s.

The LPUART has an infrared transmit encoder and receive decoder. The LPUART transmits serial bits of data that are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder, external from the LPUART. The narrow pulses are then stretched by the infrared receive decoder to get back to a serial bit stream to be received by the UART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active high pulses.

The infrared submodule receives its clock sources from the LPUART. One of these two clocks are selected in the infrared submodule to generate either 1/OSR, 2/OSR, 3/OSR, or 4/OSR narrow pulses during transmission.

38.3.5.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the LPUART_TX signal. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent at the start of the bit with a duration of 1/OSR, 2/OSR, 3/OSR, or 4/OSR of a bit time. A narrow low pulse is transmitted for a zero bit when LPUART_CTRL[TXINV] is cleared, while a narrow high pulse is transmitted for a zero bit when LPUART_CTRL[TXINV] is set.

38.3.5.2 Infrared receive decoder

The infrared receive block converts data from the LPUART_RX signal to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow low pulse is expected for a zero bit when LPUART_STAT[RXINV] is cleared, while a narrow high pulse is expected for a zero bit when LPUART_STAT[RXINV] is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

38.3.6 Interrupts and status flags

The LPUART transmitter has two status flags that can optionally generate hardware interrupt requests. Transmit data register empty LPUART_STAT[TDRE]) indicates when there is room in the transmit data buffer to write another transmit character to LPUART_DATA. If the transmit interrupt enable LPUART_CTRL[TIE]) bit is set, a hardware interrupt is requested when LPUART_STAT[TDRE] is set. Transmit complete

(LPUART_STAT[TC]) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with LPUART_TX at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (LPUART_CTRL[TCIE]) bit is set, a hardware interrupt is requested when LPUART_STAT[TC] is set. Instead of hardware interrupts, software polling may be used to monitor the LPUART_STAT[TDRE] and LPUART_STAT[TC] status flags if the corresponding LPUART_CTRL[TIE] or LPUART_CTRL[TCIE] local interrupt masks are cleared.

When a program detects that the receive data register is full (LPUART_STAT[RDRF] = 1), it gets the data from the receive data register by reading LPUART_DATA. The LPUART_STAT[RDRF] flag is cleared by reading LPUART_DATA.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the LPUART_RX line remains idle for an extended period of time. IDLE is cleared by writing 1 to the LPUART_STAT[IDLE] flag. After LPUART_STAT[IDLE] has been cleared, it cannot become set again until the receiver has received at least one new character and has set LPUART_STAT[RDRF].

If the associated error was detected in the received character that caused LPUART_STAT[RDRF] to be set, the error flags - noise flag (LPUART_STAT[NF]), framing error (LPUART_STAT[FE]), and parity error flag (LPUART_STAT[PF]) - are set at the same time as LPUART_STAT[RDRF]. These flags are not set in overrun cases.

If LPUART_STAT[RDRF] was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (LPUART_STAT[OR]) flag is set instead of the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the LPUART_RX serial data input pin causes the LPUART_STAT[RXEDGIF] flag to set. The LPUART_STAT[RXEDGIF] flag is cleared by writing a 1 to it. This function depends on the receiver being enabled (LPUART_CTRL[RE] = 1).

Chapter 39
Carrier Modulator Transmitter (CMT)

39.1 Introduction

The carrier modulator transmitter (CMT) module provides the means to generate the protocol timing and carrier signals for a wide variety of encoding schemes. The CMT incorporates hardware to off-load the critical and/or lengthy timing requirements associated with signal generation from the CPU, releasing much of its bandwidth to handle other tasks such as:

- Code data generation
- Data decompression, or,
- Keyboard scanning

The CMT does not include dedicated hardware configurations for specific protocols, but is intended to be sufficiently programmable in its function to handle the timing requirements of most protocols with minimal CPU intervention.

When the modulator is disabled, certain CMT registers can be used to change the state of the infrared output (IRO) signal directly. This feature allows for the generation of future protocol timing signals not readily producible by the current architecture.

39.2 Features

The features of this module include:

- Four modes of operation:
    - Time; with independent control of high and low times
    - Baseband
    - Frequency-shift key (FSK)
    - Direct software control of the IRO signal

- Extended space operation in Time, Baseband, and FSK modes
- Selectable input clock divider
- Interrupt on end-of-cycle
    - Ability to disable the IRO signal and use as timer interrupt

39.3 Block diagram

The following figure presents the block diagram of the CMT module.

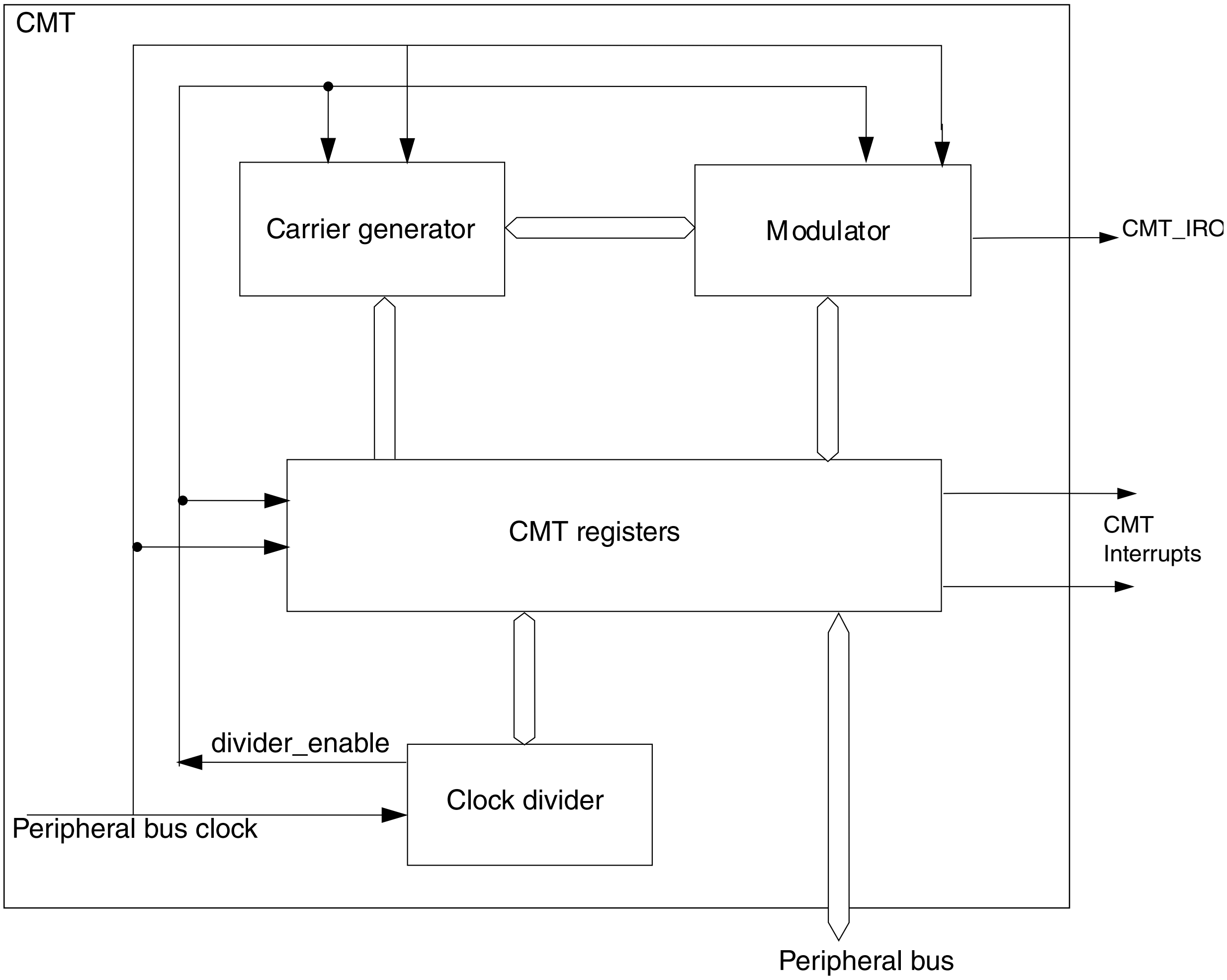


Figure 39-1. CMT module block diagram

39.4 Modes of operation

The following table describes the operation of the CMT module operates in various modes.

Table 39-1. Modes of operation

Modes	Description
Time	In Time mode, the user independently defines the high and low times of the carrier signal to determine both period and duty cycle
Baseband	When MSC[BASE] is set, the carrier output (f_{cg}) to the modulator is held high continuously to allow for the generation of baseband protocols.
Frequency-shift key	This mode allows the carrier generator to alternate between two sets of high and low times. When operating in FSK mode, the generator will toggle between the two sets when instructed by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention.

The following table summarizes the modes of operation of the CMT module.

Table 39-2. CMT modes of operation

Mode	MSC[MCGEN][1]	MSC[BASE]	MSC[FSK][2]	MSC[EXSPC]	Comment
Time	1	0	0	0	f_{cg} controlled by primary high and low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Baseband	1	1	X	0	f_{cg} is always high. The IRO signal is high when the modulator gate is open.
FSK	1	0	1	0	f_{cg} control alternates between primary high/low registers and secondary high/low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Extended Space	1	X	X	1	Setting MSC[EXSPC] causes subsequent modulator cycles to be spaces (modulator out not asserted) for the duration of the modulator period (mark and space times).
IRO Latch	0	X	X	X	OC[IROL] controls the state of the IRO signal.

[1] To prevent spurious operation, initialize all data and control registers before beginning a transmission when MSC[MCGEN]=1.

[2] This field is not double-buffered and must not be changed during a transmission while MSC[MCGEN]=1.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

39.4.1 Wait mode operation

During Wait mode, the CMT if enabled, will continue to operate normally. However, there is no change in operating modes of CMT during Wait mode, because the CPU is not operating.

39.4.2 Stop mode operation

This section describes the CMT Stop mode operations.

39.4.2.1 Normal Stop mode operation

During Normal Stop mode, clocks to the CMT module are halted. No registers are affected.

The CMT module will resume upon exit from Normal Stop mode because the clocks are halted. Software must ensure that the Normal Stop mode is not entered while the modulator is still in operation so as to prevent the IRO signal from being asserted while in Normal Stop mode. This may require a timeout period from the time that MSC[MCGEN] is cleared to allow the last modulator cycle to complete.

39.4.2.2 Low-Power Stop mode operation

During Low-Power Stop mode, the CMT module is completely powered off internally and the IRO signal state is latched and held at the time when the CMT enters this mode. To prevent the IRO signal from being asserted during Low-Power Stop mode, the software must assure that the signal is not active when entering Low-Power Stop mode. Upon wakeup from Low-Power Stop mode, the CMT module will be in the reset state.

39.5 CMT external signal descriptions

The following table shows the description of the external signal.

Table 39-3. CMT signal description

Signal	Description	I/O
CMT_IRO	Infrared Output	O

39.5.1 CMT_IRO — Infrared Output

This output signal is driven by the modulator output when MSC[MCGEN] and OC[IROPEN] are set. The IRO signal starts a valid transmission with a delay, after MSC[MCGEN] bit be asserted to high, that can be calculated based on two register bits. Table 39-4 shows how to calculate this delay.

The following table describes conditions for the IRO signal to be active.

If	Then
MSC[MCGEN] is cleared and OC[IROPEN] is set	The signal is driven by OC[IROL] . This enables user software to directly control the state of the IRO signal by writing to OC[IROL] .
OC[IROPEN] is cleared	The signal is disabled and is not driven by the CMT module. Therefore, CMT can be configured as a modulo timer for generating periodic interrupts without causing signal activity.

Table 39-4. CMT_IRO signal delay calculation

Condition	Delay (bus clock cycles)
MSC[CMTDIV] = 0	PPS[PPSDIV] + 2
MSC[CMTDIV] > 0	(PPS[PPSDIV] *2) + 3

39.6 Memory map/register definition

The following registers control and monitor the CMT operation.

The address of a register is the sum of a base address and an address offset. The base address is defined at the chip level. The address offset is defined at the module level.

CMT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2000	CMT Carrier Generator High Data Register 1 (CMT_CGH1)	8	R/W	Undefined	39.6.1/802
4006_2001	CMT Carrier Generator Low Data Register 1 (CMT_CGL1)	8	R/W	Undefined	39.6.2/803
4006_2002	CMT Carrier Generator High Data Register 2 (CMT_CGH2)	8	R/W	Undefined	39.6.3/803
4006_2003	CMT Carrier Generator Low Data Register 2 (CMT_CGL2)	8	R/W	Undefined	39.6.4/804
4006_2004	CMT Output Control Register (CMT_OC)	8	R/W	00h	39.6.5/804
4006_2005	CMT Modulator Status and Control Register (CMT_MSC)	8	R/W	00h	39.6.6/805
4006_2006	CMT Modulator Data Register Mark High (CMT_CMD1)	8	R/W	Undefined	39.6.7/807
4006_2007	CMT Modulator Data Register Mark Low (CMT_CMD2)	8	R/W	Undefined	39.6.8/808
4006_2008	CMT Modulator Data Register Space High (CMT_CMD3)	8	R/W	Undefined	39.6.9/808
4006_2009	CMT Modulator Data Register Space Low (CMT_CMD4)	8	R/W	Undefined	39.6.10/ 809
4006_200A	CMT Primary Prescaler Register (CMT_PPS)	8	R/W	00h	39.6.11/ 809
4006_200B	CMT Direct Memory Access Register (CMT_DMA)	8	R/W	00h	39.6.12/ 810

39.6.1 CMT Carrier Generator High Data Register 1 (CMT_CGH1)

This data register contains the primary high value for generating the carrier output.

Address: 4006_2000h base + 0h offset = 4006_2000h

Bit	7	6	5	4	3	2	1	0
Read Write	PH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGH1 field descriptions

Field	Description
PH	Primary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier high time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

39.6.2 CMT Carrier Generator Low Data Register 1 (CMT_CGL1)

This data register contains the primary low value for generating the carrier output.

Address: 4006_2000h base + 1h offset = 4006_2001h

Bit	7	6	5	4	3	2	1	0
Read Write	PL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGL1 field descriptions

Field	Description
PL	Primary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

39.6.3 CMT Carrier Generator High Data Register 2 (CMT_CGH2)

This data register contains the secondary high value for generating the carrier output.

Address: 4006_2000h base + 2h offset = 4006_2002h

Bit	7	6	5	4	3	2	1	0
Read Write	SH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGH2 field descriptions

Field	Description
SH	Secondary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under control of the modulator.

CMT_CGH2 field descriptions (continued)

Field	Description
	The secondary carrier high time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

39.6.4 CMT Carrier Generator Low Data Register 2 (CMT_CGL2)

This data register contains the secondary low value for generating the carrier output.

Address: 4006_2000h base + 3h offset = 4006_2003h

Bit	7	6	5	4	3	2	1	0
Read Write	SL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CGL2 field descriptions

Field	Description
SL	Secondary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under the control of the modulator. The secondary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

39.6.5 CMT Output Control Register (CMT_OC)

This register is used to control the IRO signal of the CMT module.

Address: 4006_2000h base + 4h offset = 4006_2004h

Bit	7	6	5	4	3	2	1	0
Read	IROL	CMTPOL	IROPEN	0				
Write								
Reset	0	0	0	0	0	0	0	0

CMT_OC field descriptions

Field	Description
7 IROL	IRO Latch Control Reads the state of the IRO latch. Writing to IROL changes the state of the IRO signal when MSC[MCGEN] is cleared and IROPEN is set.
6 CMTPOL	CMT Output Polarity Controls the polarity of the IRO signal. 0 The IRO signal is active-low. 1 The IRO signal is active-high.
5 IROPEN	IRO Pin Enable Enables and disables the IRO signal. When the IRO signal is enabled, it is an output that drives out either the CMT transmitter output or the state of IROL depending on whether MSC[MCGEN] is set or not. Also, the state of output is either inverted or non-inverted, depending on the state of CMTPOL. When the IRO signal is disabled, it is in a high-impedance state and is unable to draw any current. This signal is disabled during reset. 0 The IRO signal is disabled. 1 The IRO signal is enabled as output.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

39.6.6 CMT Modulator Status and Control Register (CMT_MSC)

This register contains the modulator and carrier generator enable (MCGEN), end of cycle interrupt enable (EOCIE), FSK mode select (FSK), baseband enable (BASE), extended space (EXSPC), prescaler (CMTDIV) bits, and the end of cycle (EOCF) status bit.

Address: 4006_2000h base + 5h offset = 4006_2005h

Bit	7	6	5	4	3	2	1	0
Read	EOCF	CMTDIV		EXSPC	BASE	FSK	EOCIE	MCGEN
Write	▨							
Reset	0	0	0	0	0	0	0	0

CMT_MSC field descriptions

Field	Description
7 EOCF	End Of Cycle Status Flag Sets when:

Table continues on the next page...

CMT_MSC field descriptions (continued)

Field	Description
	• The modulator is not currently active and MCGEN is set to begin the initial CMT transmission. • At the end of each modulation cycle while MCGEN is set. This is recognized when a match occurs between the contents of the space period register and the down counter. At this time, the counter is initialized with, possibly new contents of the mark period buffer, CMD1 and CMD2, and the space period register is loaded with, possibly new contents of the space period buffer, CMD3 and CMD4. This flag is cleared by reading MSC followed by an access of CMD2 or CMD4, or by the DMA transfer. 0 End of modulation cycle has not occured since the flag last cleared. 1 End of modulator cycle has occurred.
6–5 CMTDIV	CMT Clock Divide Prescaler Causes the CMT to be clocked at the IF signal frequency, or the IF frequency divided by 2 ,4, or 8 . This field must not be changed during a transmission because it is not double-buffered. 00 IF ÷ 1 01 IF ÷ 2 10 IF ÷ 4 11 IF ÷ 8
4 EXSPC	Extended Space Enable Enables the extended space operation. 0 Extended space is disabled. 1 Extended space is enabled.
3 BASE	Baseband Enable When set, BASE disables the carrier generator and forces the carrier output high for generation of baseband protocols. When BASE is cleared, the carrier generator is enabled and the carrier output toggles at the frequency determined by values stored in the carrier data registers. This field is cleared by reset. This field is not double-buffered and must not be written to during a transmission. 0 Baseband mode is disabled. 1 Baseband mode is enabled.
2 FSK	FSK Mode Select Enables FSK operation. 0 The CMT operates in Time or Baseband mode. 1 The CMT operates in FSK mode.
1 EOCIE	End of Cycle Interrupt Enable Requests to enable a CPU interrupt when EOCF is set if EOCIE is high.

Table continues on the next page...

CMT_MSC field descriptions (continued)

Field	Description
	0 CPU interrupt is disabled. 1 CPU interrupt is enabled.
0 MCGEN	Modulator and Carrier Generator Enable Setting MCGEN will initialize the carrier generator and modulator and will enable all clocks. When enabled, the carrier generator and modulator will function continuously. When MCGEN is cleared, the current modulator cycle will be allowed to expire before all carrier and modulator clocks are disabled to save power and the modulator output is forced low. **NOTE:** To prevent spurious operation, the user should initialize all data and control registers before enabling the system. 0 Modulator and carrier generator disabled 1 Modulator and carrier generator enabled

39.6.7 CMT Modulator Data Register Mark High (CMT_CMD1)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: 4006_2000h base + 6h offset = 4006_2006h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CMD1 field descriptions

Field	Description
MB[15:8]	MB[15:8] Controls the upper mark periods of the modulator for all modes.

39.6.8 CMT Modulator Data Register Mark Low (CMT_CMD2)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: 4006_2000h base + 7h offset = 4006_2007h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD2 field descriptions

Field	Description
MB[7:0]	MB[7:0] Controls the lower mark periods of the modulator for all modes.

39.6.9 CMT Modulator Data Register Space High (CMT_CMD3)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: 4006_2000h base + 8h offset = 4006_2008h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD3 field descriptions

Field	Description
SB[15:8]	SB[15:8] Controls the upper space periods of the modulator for all modes.

39.6.10 CMT Modulator Data Register Space Low (CMT_CMD4)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: 4006_2000h base + 9h offset = 4006_2009h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD4 field descriptions

Field	Description
SB[7:0]	SB[7:0] Controls the lower space periods of the modulator for all modes.

39.6.11 CMT Primary Prescaler Register (CMT_PPS)

This register is used to set the Primary Prescaler Divider field (PPSDIV).

Address: 4006_2000h base + Ah offset = 4006_200Ah

Bit	7	6	5	4	3	2	1	0
Read	0				PPSDIV			
Write	▨							
Reset	0	0	0	0	0	0	0	0

CMT_PPS field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PPSDIV	Primary Prescaler Divider Divides the CMT clock to generate the Intermediate Frequency clock enable to the secondary prescaler. 0000 Bus clock ÷ 1 0001 Bus clock ÷ 2 0010 Bus clock ÷ 3

Table continues on the next page...

CMT_PPS field descriptions (continued)

Field	Description
	0011 Bus clock ÷ 4 0100 Bus clock ÷ 5 0101 Bus clock ÷ 6 0110 Bus clock ÷ 7 0111 Bus clock ÷ 8 1000 Bus clock ÷ 9 1001 Bus clock ÷ 10 1010 Bus clock ÷ 11 1011 Bus clock ÷ 12 1100 Bus clock ÷ 13 1101 Bus clock ÷ 14 1110 Bus clock ÷ 15 1111 Bus clock ÷ 16

39.6.12 CMT Direct Memory Access Register (CMT_DMA)

This register is used to enable/disable direct memory access (DMA).

Address: 4006_2000h base + Bh offset = 4006_200Bh

Bit	7	6	5	4	3	2	1	0
Read	0							DMA
Write								
Reset	0	0	0	0	0	0	0	0

CMT_DMA field descriptions

Field	Description
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables the DMA protocol. 0 DMA transfer request and done are disabled. 1 DMA transfer request and done are enabled.

39.7 Functional description

The CMT module primarily consists of clock divider, carrier generator, and modulator.

39.7.1 Clock divider

The CMT was originally designed to be based on an 8 MHz bus clock that could be divided by 1, 2, 4, or 8 according to the specification. To be compatible with higher bus frequency, the primary prescaler (PPS) was developed to receive a higher frequency and generate a clock enable signal called intermediate frequency (IF). This IF must be approximately equal to 8 MHz and will work as a clock enable to the secondary prescaler. The following figure shows the clock divider block diagram.

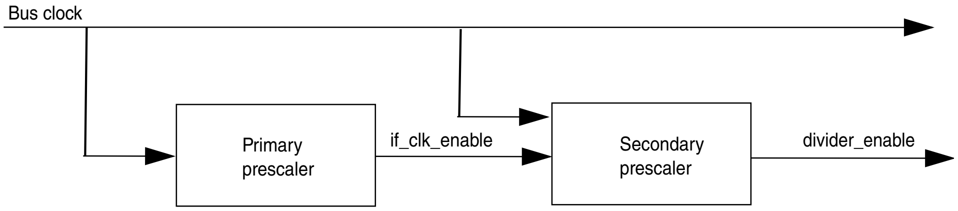


Figure 39-2. Clock divider block diagram

For compatibility with previous versions of CMT, when bus clock = 8 MHz, the PPS must be configured to zero. The PPS counter is selected according to the bus clock to generate an intermediate frequency approximately equal to 8 MHz.

39.7.2 Carrier generator

The carrier generator resolution is 125 ns when operating with an 8 MHz intermediate frequency signal and the secondary prescaler is set to divide by 1, or, when MSC[CMTDIV] = 00. The carrier generator can generate signals with periods between 250 ns (4 MHz) and 127.5 µs (7.84 kHz) in steps of 125 ns. The following table shows the relationship between the clock divide bits and the carrier generator resolution, minimum carrier generator period, and minimum modulator period.

Table 39-5. Clock divider

Bus clock (MHz)	MSC[CMTDIV]	Carrier generator resolution (µs)	Min. carrier generator period (µs)	Min. modulator period (µs)
8	00	0.125	0.25	1.0
8	01	0.25	0.5	2.0
8	10	0.5	1.0	4.0
8	11	1.0	2.0	8.0

The possible duty cycle options depend upon the number of counts required to complete the carrier period. For example, 1.6 MHz signal has a period of 625 ns and will therefore require 5 x 125 ns counts to generate. These counts may be split between high and low times, so the duty cycles available will be:

- 20% with one high and four low times
- 40% with two high and three low times
- 60% with three high and two low times, and
- 80% with four high and one low time

.

For low-frequency signals with large periods, high-resolution duty cycles as a percentage of the total period, are possible.

The carrier signal is generated by counting a register-selected number of input clocks (125 ns for an 8 MHz bus) for both the carrier high time and the carrier low time. The period is determined by the total number of clocks counted. The duty cycle is determined by the ratio of high-time clocks to total clocks counted. The high and low time values are user-programmable and are held in two registers.

An alternate set of high/low count values is held in another set of registers to allow the generation of dual-frequency FSK protocols without CPU intervention.

Note

Only nonzero data values are allowed. The carrier generator will not work if any of the count values are equal to zero.

MSC[MCGEN] must be set and MSC[BASE] must be cleared to enable carrier generator clocks. When MSC[BASE] is set, the carrier output to the modulator is held high continuously. The following figure represents the block diagram of the clock generator.

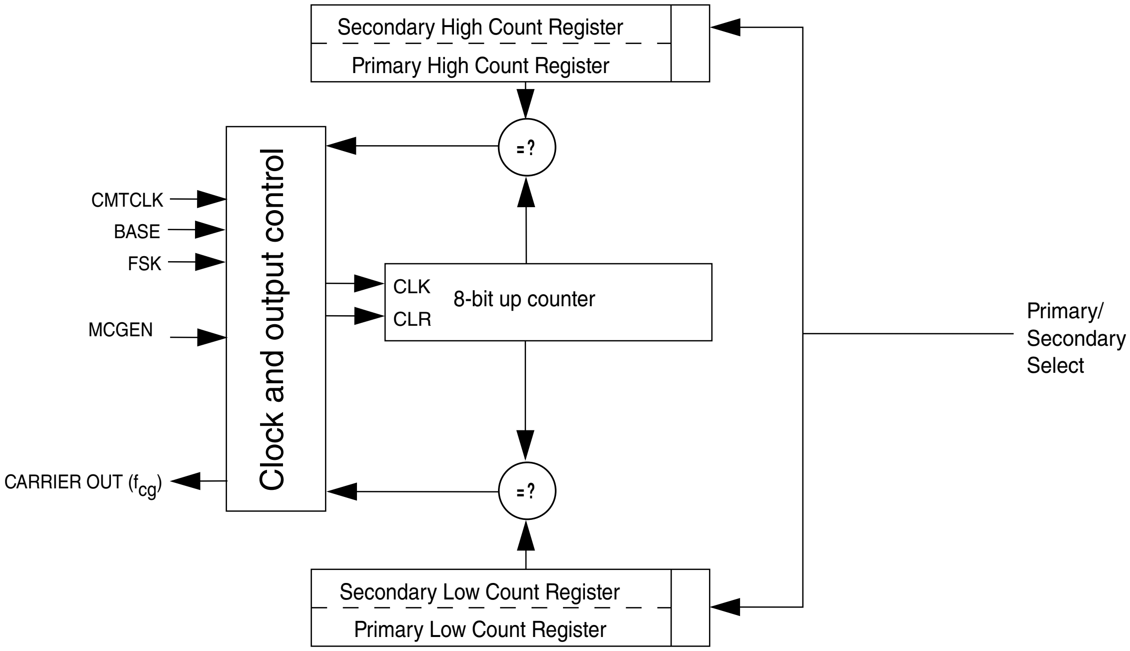


Figure 39-3. Carrier generator block diagram

The high/low time counter is an 8-bit up counter. After each increment, the contents of the counter are compared with the appropriate high or low count value register. When the compare value is reached, the counter is reset to a value of 0x01, and the compare is redirected to the other count value register.

Assuming that the high time count compare register is currently active, a valid compare will cause the carrier output to be driven low. The counter will continue to increment starting at the reset value of 0x01. When the value stored in the selected low count value register is reached, the counter will again be reset and the carrier output will be driven high.

The cycle repeats, automatically generating a periodic signal which is directed to the modulator. The lower frequency with maximum period, f_{max}, and highest frequency with minimum period, f_{min}, which can be generated, are defined as:

$f_{max} = f_{CMTCLK} \div (2 * 1)$ Hz

$f_{min} = f_{CMTCLK} \div (2 * (2^8 - 1))$ Hz

In the general case, the carrier generator output frequency is:

$f_{cg} = f_{CMTCLK} \div (\text{High count} + \text{Low count})$ Hz

Where: $0 < \text{High count} < 256$ and

$0 < \text{Low count} < 256$

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\text{DutyCycle} = \frac{\text{Highcount}}{\text{Highcount+Lowcount}}$$

39.7.3 Modulator

The modulator block controls the state of the infrared out signal (IRO). The modulator output is gated on to the IRO signal when the modulator/carrier generator is enabled. . When the modulator/carrier generator is disabled, the IRO signal is controlled by the state of the IRO latch. OC[CMTPOL] enables the IRO signal to be active-high or active-low.

The following table describes the functions of the modulators in different modes:

Table 39-6. Mode functions

Mode	Function
Time	The modulator can gate the carrier onto the modulator output.
Baseband	The modulator can control the logic level of the modulator output.
FSK	The modulator can count carrier periods and instruct the carrier generator to alternate between two carrier frequencies whenever a modulation period consisting of mark and space counts, expires.

The modulator provides a simple method to control protocol timing. The modulator has a minimum resolution of 1.0 µs with an 8 MHz. It can count bus clocks to provide real-time control, or carrier clocks for self-clocked protocols.

The modulator includes a 17-bit down counter with underflow detection. The counter is loaded from the 16-bit modulation mark period buffer registers, CMD1 and CMD2. The most significant bit is loaded with a logic 0 and serves as a sign bit.

When	Then
The counter holds a positive value	The modulator gate is open and the carrier signal is driven to the transmitter block.
The counter underflows	The modulator gate is closed and a 16-bit comparator is enabled which compares the logical complement of the value of the down counter with the contents of the modulation space period register which has been loaded from the registers, CMD3 and CMD4.

When a match is obtained, the cycle repeats by opening the modulator gate, reloading the counter with the contents of CMD1 and CMD2, and reloading the modulation space period register with the contents of CMD3 and CMD4.

The modulation space period is activated when the carrier signal is low to prohibit cutting off the high pulse of a carrier signal. If the carrier signal is high, the modulator extends the mark period until the carrier signal becomes low. To deassert the space period and assert the mark period, the carrier signal must have gone low to ensure that a space period is not erroneously shortened.

If the contents of the modulation space period register are all zeroes, the match will be immediate and no space period will be generated, for instance, for FSK protocols that require successive bursts of different frequencies).

MSC[MCGEN] must be set to enable the modulator timer.

The following figure presents the block diagram of the modulator.

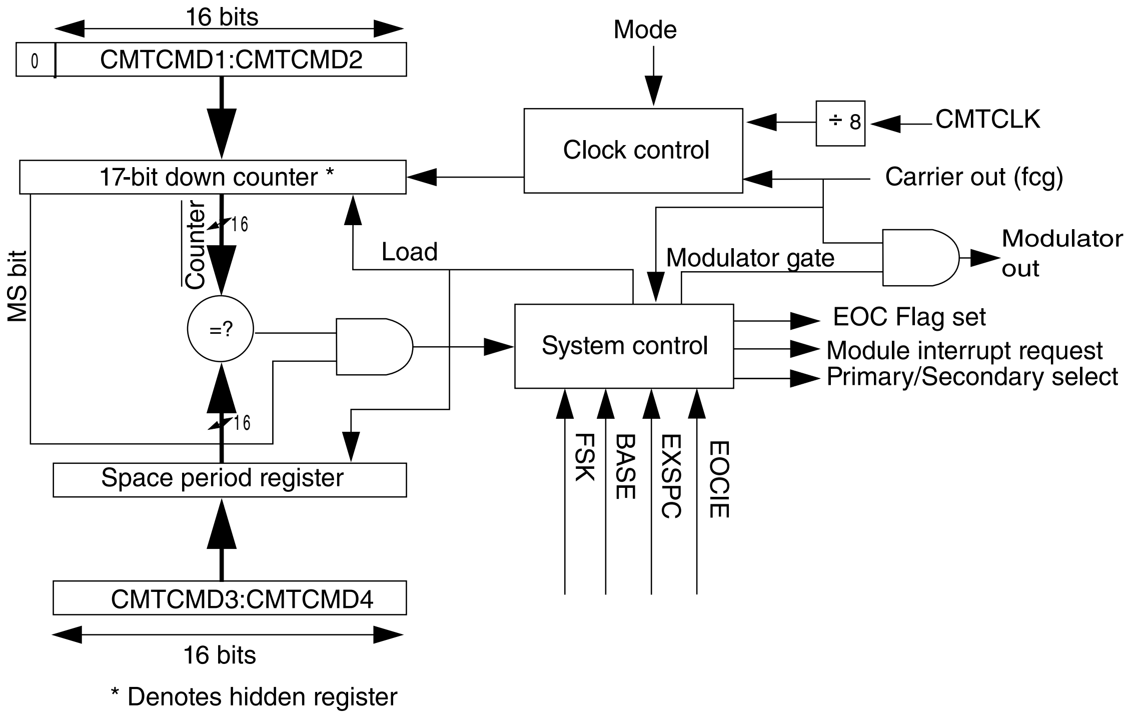


Figure 39-4. Modulator block diagram

39.7.3.1 Time mode

When the modulator operates in Time mode, or, when MSC[MCGEN] is set, and MSC[BASE] and MSC[FSK] are cleared:

- The modulation mark period consists of an integer number of (CMTCLK ÷ 8) clock periods.
- The modulation space period consists of 0 or an integer number of (CMTCLK ÷ 8) clock periods.

With an 8 MHz IF and MSC[CMTDIV] = 00, the modulator resolution is 1 μs and has a maximum mark and space period of about 65.535 ms each . See Figure 39-5 for an example of the Time and Baseband mode outputs.

The mark and space time equations for Time and Baseband mode are:

$t_{mark} = (CMD1:CMD2 + 1) \div (f_{CMTCLK} \div 8)$

$t_{space} = CMD3:CMD4 \div (f_{CMTCLK} \div 8)$

where CMD1:CMD2 and CMD3:CMD4 are the decimal values of the concatenated registers.

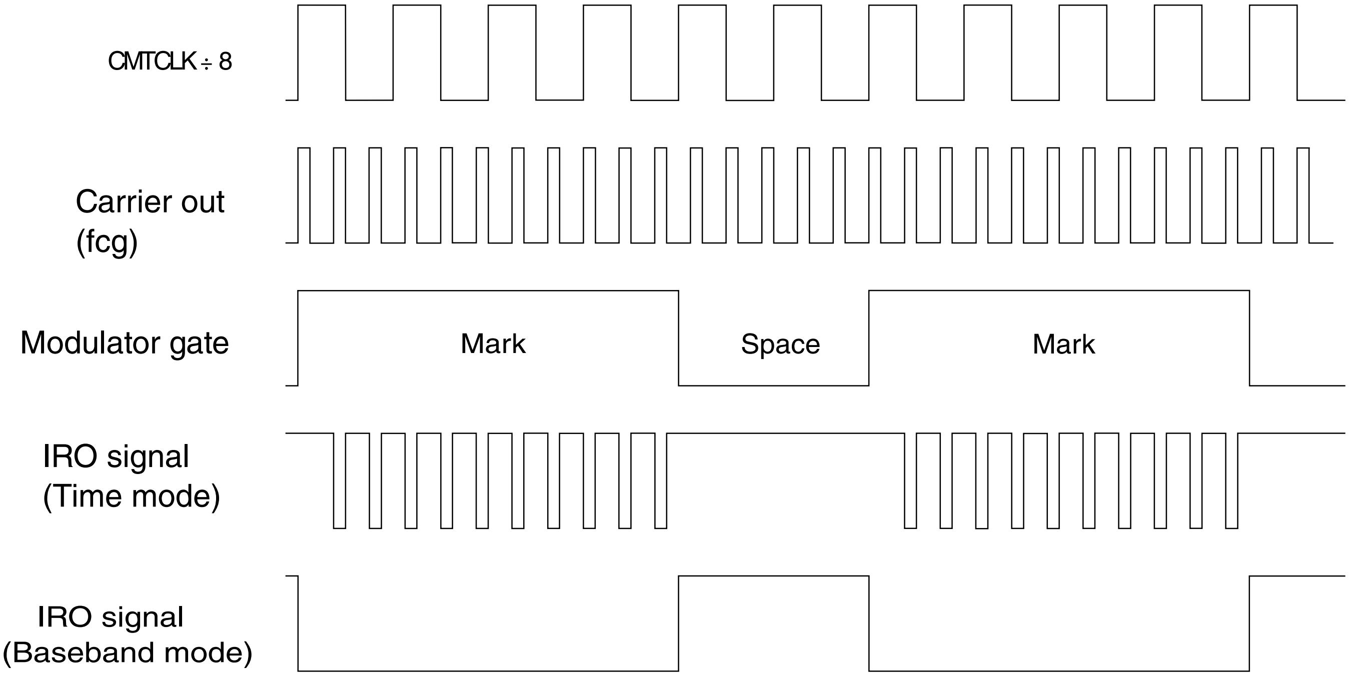


Figure 39-5. Example: CMT output in Time and Baseband modes with OC[CMTPOL]=0

39.7.3.2 Baseband mode

Baseband mode, that is, when MSC[MCGEN] and MSC[BASE] are set, is a derivative of Time mode, where the mark and space period is based on (CMTCLK ÷ 8) counts. The mark and space calculations are the same as in Time mode.

In this mode, the modulator output will be at a logic 1 for the duration of the mark period and at a logic 0 for the duration of a space period. See Figure 39-5 for an example of the output for both Baseband and Time modes. In the example, the carrier out frequency (f_{cg}) is generated with a high count of 0x01 and a low count of 0x02 that results in a divide of 3 of CMTCLK with a 33% duty cycle. The modulator down counter was loaded with the value 0x0003 and the space period register with 0x0002.

Note

The waveforms in Figure 39-5 and Figure 39-6 are for the purpose of conceptual illustration and are not meant to represent precise timing relationships between the signals shown.

39.7.3.3 FSK mode

When the modulator operates in FSK mode, that is, when MSC[MCGEN] and MSC[FSK] are set, and MSC[BASE] is cleared:

- The modulation mark and space periods consist of an integer number of carrier clocks (space period can be zero).
- When the mark period expires, the space period is transparently started as in Time mode.
- The carrier generator toggles between primary and secondary data register values whenever the modulator space period expires.

The space period provides an interpulse gap (no carrier). If CMD3:CMD4 = 0x0000, then the modulator and carrier generator will switch between carrier frequencies without a gap or any carrier glitches (zero space).

Using timing data for carrier burst and interpulse gap length calculated by the CPU, FSK mode can automatically generate a phase-coherent, dual-frequency FSK signal with programmable burst and interburst gaps.

The mark and space time equations for FSK mode are:

$t_{mark} = (\mathrm{CMD1:CMD2} + 1) \div f_{cg}$

$t_{space} = (\mathrm{CMD3:CMD4}) \div f_{cg}$

Where f_{cg} is the frequency output from the carrier generator. The example in Figure 39-6 shows what the IRO signal looks like in FSK mode with the following values:

- CMD1:CMD2 = 0x0003
- CMD3:CMD4 = 0x0002
- Primary carrier high count = 0x01
- Primary carrier low count = 0x02

- Secondary carrier high count = 0x03
- Secondary carrier low count = 0x01

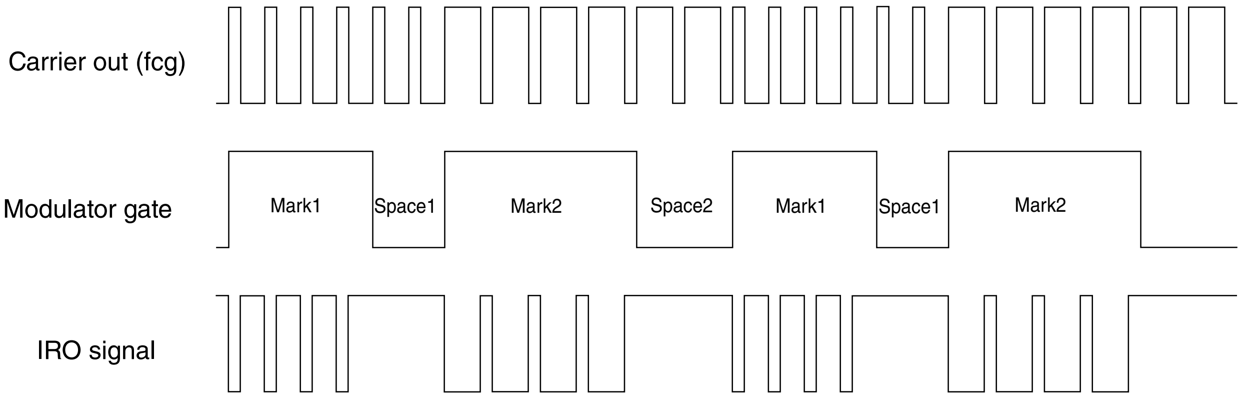


Figure 39-6. Example: CMT output in FSK mode

39.7.4 Extended space operation

In either Time, Baseband, or FSK mode, the space period can be made longer than the maximum possible value of the space period register. Setting MSC[EXSPC] will force the modulator to treat the next modulation period beginning with the next load of the counter and space period register, as a space period equal in length to the mark and space counts combined. Subsequent modulation periods will consist entirely of these extended space periods with no mark periods. Clearing MSC[EXSPC] will return the modulator to standard operation at the beginning of the next modulation period.

39.7.4.1 EXSPC operation in Time mode

To calculate the length of an extended space in Time or Baseband mode, add the mark and space times and multiply by the number of modulation periods when MSC[EXSPC] is set.

$t_{exspace} = (t_{mark} + t_{space})$ * (number of modulation periods)

For an example of extended space operation, see Figure 39-7.

Note

The extended space enable feature can be used to emulate a zero mark event.

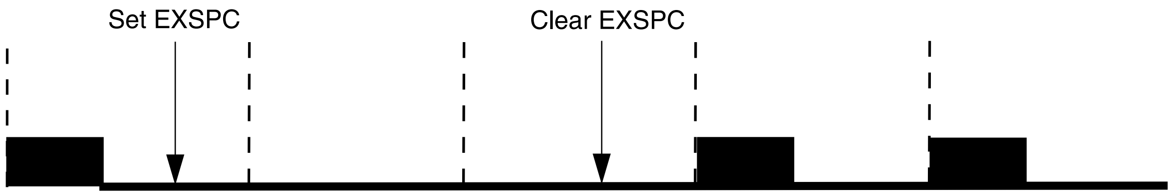


Figure 39-7. Extended space operation

39.7.4.2 EXSPC operation in FSK mode

In FSK mode, the modulator continues to count carrier out clocks, alternating between the primary and secondary registers at the end of each modulation period.

To calculate the length of an extended space in FSK mode, it is required to know whether MSC[EXSPC] was set on a primary or secondary modulation period, and the total number of both primary and secondary modulation periods completed while MSC[EXSPC] is high. A status bit for the current modulation is not accessible to the CPU. If necessary, software must maintain tracking of the current primary or secondary modulation cycle. The extended space period ends at the completion of the space period time of the modulation period during which MSC[EXSPC]is cleared.

The following table depicts the equations which can be used to calculate the extended space period depending on when MSC[EXSPC] is set.

If	Then
MSC[EXSPC] was set during a primary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_p + (t_{mark} + t_{space})_s + (t_{mark} + t_{space})_p + ...$
MSC[EXSPC] bit was set during a secondary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_s + (t_{mark} + t_{space})_p + (t_{mark} + t_{space})_s + ...$

Where the subscripts p and s refer to mark and space times for the primary and secondary modulation cycles.

39.8 CMT interrupts and DMA

The CMT generates an interrupt request or a DMA transfer request according to MSC[EOCIE], MSC[EOCF], DMA[DMA] bits.

Table 39-7. DMA transfer request x CMT interrupt request

MSC[EOCF]	DMA[DMA]	MSC[EOCIE]	DMA transfer request	CMT interrupt request
0	X	X	0	0
1	X	0	0	0
1	0	1	0	1
1	1	1	1	0

MSC[EOCF] is set:

- When the modulator is not currently active and MSC[MCGEN] is set to begin the initial CMT transmission.
- At the end of each modulation cycle when the counter is reloaded from CMD1:CMD2, while MSC[MCGEN] is set.

When MSC[MCGEN] is cleared and then set before the end of the modulation cycle, MSC[EOCF] will not be set when MSC[MCGEN] is set, but will become set at the end of the current modulation cycle.

When MSC[MCGEN] becomes disabled, the CMT module does not set MSC[EOCF] at the end of the last modulation cycle.

If MSC[EOCIE] is high when MSC[EOCF] is set, the CMT module will generate an interrupt request or a DMA transfer request.

MSC[EOCF] must be cleared to prevent from being generated by another event like interrupt or DMA request, after exiting the service routine. See the following table.

Table 39-8. How to clear MSC[EOCF]

DMA[DMA]	MSC[EOCIE]	Description
0	X	MSC[EOCF] is cleared by reading MSC followed by an access of CMD2 or CMD4.
1	X	MSC[EOCF] is cleared by the CMT DMA transfer done.

The EOC interrupt is coincident with:

- Loading the down-counter with the contents of CMD1:CMD2
- Loading the space period register with the contents of CMD3:CMD4

The EOC interrupt provides a means for the user to reload new mark/space values into the modulator data registers. Modulator data register updates will take effect at the end of the current modulation cycle.

NOTE

The down-counter and space period register are updated at the end of every modulation cycle, irrespective of interrupt handling and the state of MSC[EOCF].

Chapter 40
General-purpose input/output (GPIO)

The general-purpose input and output (GPIO) module is accessible via the peripheral bus and also communicates to the processor core via a zero wait state interface (IOPORT) for maximum pin performance. The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function.

40.1 Introduction

The general-purpose input and output (GPIO) module is accessible via the peripheral bus and also communicates to the processor core via a zero wait state interface (IOPORT) for maximum pin performance. The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled.

Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

40.1.1 Features

Features of the GPIO module include:

- Port Data Input register visible in all digital pin-multiplexing modes
- Port Data Output register with corresponding set/clear/toggle registers
- Port Data Direction register
- Zero wait state access to GPIO registers through IOPORT

NOTE

The GPIO module is clocked by system clock.

40.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 40-1. Modes of operation

Modes of operation	Description
Run	The GPIO module operates normally.
Wait	The GPIO module operates normally.
Stop	The GPIO module is disabled.
Debug	The GPIO module operates normally.

40.1.3 GPIO signal descriptions

Table 40-2. GPIO signal descriptions

GPIO signal descriptions	Description	I/O
PORTA31–PORTA0	General-purpose input/output	I/O
PORTB31–PORTB0	General-purpose input/output	I/O
PORTC31–PORTC0	General-purpose input/output	I/O

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

40.1.3.1 Detailed signal description

Table 40-3. GPIO interface-detailed signal descriptions

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">PORTA31–PORTA0
PORTB31–PORTB0
PORTC31–PORTC0</td><td rowspan="3">I/O</td><td colspan="2">General-purpose input/output</td></tr>
<tr><td>State meaning</td><td>Asserted: The pin is logic 1.
Deasserted: The pin is logic 0.</td></tr>
<tr><td>Timing</td><td>Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.</td></tr>
</table>

Table 40-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
			Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

40.2 Memory map and register definition

Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's registers appear with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. Refer to the to see the exact control bits for the non-identical port instance.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F000	Port Data Output Register (GPIOA_PDOR)	32	R/W	0000_0000h	40.2.1/826
400F_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads 0)	0000_0000h	40.2.2/827
400F_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads 0)	0000_0000h	40.2.3/827
400F_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads 0)	0000_0000h	40.2.4/828
400F_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	40.2.5/828
400F_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	40.2.6/829

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F040	Port Data Output Register (GPIOB_PDOR)	32	R/W	0000_0000h	40.2.1/826
400F_F044	Port Set Output Register (GPIOB_PSOR)	32	W (always reads 0)	0000_0000h	40.2.2/827
400F_F048	Port Clear Output Register (GPIOB_PCOR)	32	W (always reads 0)	0000_0000h	40.2.3/827
400F_F04C	Port Toggle Output Register (GPIOB_PTOR)	32	W (always reads 0)	0000_0000h	40.2.4/828
400F_F050	Port Data Input Register (GPIOB_PDIR)	32	R	0000_0000h	40.2.5/828
400F_F054	Port Data Direction Register (GPIOB_PDDR)	32	R/W	0000_0000h	40.2.6/829
400F_F080	Port Data Output Register (GPIOC_PDOR)	32	R/W	0000_0000h	40.2.1/826
400F_F084	Port Set Output Register (GPIOC_PSOR)	32	W (always reads 0)	0000_0000h	40.2.2/827
400F_F088	Port Clear Output Register (GPIOC_PCOR)	32	W (always reads 0)	0000_0000h	40.2.3/827
400F_F08C	Port Toggle Output Register (GPIOC_PTOR)	32	W (always reads 0)	0000_0000h	40.2.4/828
400F_F090	Port Data Input Register (GPIOC_PDIR)	32	R	0000_0000h	40.2.5/828
400F_F094	Port Data Direction Register (GPIOC_PDDR)	32	R/W	0000_0000h	40.2.6/829

40.2.1 Port Data Output Register (GPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDOR field descriptions

Field	Description
PDO	Port Data Output Register bits for unbonded pins return a undefined value when read. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

40.2.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTSO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PSOR field descriptions

Field	Description
PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

40.2.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTCO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PCOR field descriptions

Field	Description
PTCO	Port Clear Output

GPIOx_PCOR field descriptions (continued)

Field	Description
	Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

40.2.4 Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTTO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PTOR field descriptions

Field	Description
PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

40.2.5 Port Data Input Register (GPIOx_PDIR)

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDIR field descriptions

Field	Description
PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

40.2.6 Port Data Direction Register (GPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDDR field descriptions

Field	Description
PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

40.3 FGPIO memory map and register definition

The GPIO registers are also aliased to the IOPORT interface on the Cortex-M0+ from address 0xF800_0000.

Accesses via the IOPORT interface occur in parallel with any instruction fetches and will therefore complete in a single cycle. This aliased Fast GPIO memory map is called FGPIO.

Any read or write access to the FGPIO memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states, except error accesses which complete with one wait state.

NOTE

For simplicity, each FGPIO port's registers appear with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. Refer to the Chip Configuration chapter to see the exact control bits for the non-identical port instance.

FGPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F800_0000	Port Data Output Register (FGPIOA_PDOR)	32	R/W	0000_0000h	40.3.1/831
F800_0004	Port Set Output Register (FGPIOA_PSOR)	32	W (always reads 0)	0000_0000h	40.3.2/831
F800_0008	Port Clear Output Register (FGPIOA_PCOR)	32	W (always reads 0)	0000_0000h	40.3.3/832
F800_000C	Port Toggle Output Register (FGPIOA_PTOR)	32	W (always reads 0)	0000_0000h	40.3.4/832
F800_0010	Port Data Input Register (FGPIOA_PDIR)	32	R	0000_0000h	40.3.5/833
F800_0014	Port Data Direction Register (FGPIOA_PDDR)	32	R/W	0000_0000h	40.3.6/833
F800_0040	Port Data Output Register (FGPIOB_PDOR)	32	R/W	0000_0000h	40.3.1/831
F800_0044	Port Set Output Register (FGPIOB_PSOR)	32	W (always reads 0)	0000_0000h	40.3.2/831
F800_0048	Port Clear Output Register (FGPIOB_PCOR)	32	W (always reads 0)	0000_0000h	40.3.3/832
F800_004C	Port Toggle Output Register (FGPIOB_PTOR)	32	W (always reads 0)	0000_0000h	40.3.4/832
F800_0050	Port Data Input Register (FGPIOB_PDIR)	32	R	0000_0000h	40.3.5/833
F800_0054	Port Data Direction Register (FGPIOB_PDDR)	32	R/W	0000_0000h	40.3.6/833
F800_0080	Port Data Output Register (FGPIOC_PDOR)	32	R/W	0000_0000h	40.3.1/831
F800_0084	Port Set Output Register (FGPIOC_PSOR)	32	W (always reads 0)	0000_0000h	40.3.2/831
F800_0088	Port Clear Output Register (FGPIOC_PCOR)	32	W (always reads 0)	0000_0000h	40.3.3/832
F800_008C	Port Toggle Output Register (FGPIOC_PTOR)	32	W (always reads 0)	0000_0000h	40.3.4/832
F800_0090	Port Data Input Register (FGPIOC_PDIR)	32	R	0000_0000h	40.3.5/833
F800_0094	Port Data Direction Register (FGPIOC_PDDR)	32	R/W	0000_0000h	40.3.6/833

40.3.1 Port Data Output Register (FGPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PDOR field descriptions

Field	Description
PDO	Port Data Output Unimplemented pins for a particular device read as zero. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

40.3.2 Port Set Output Register (FGPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTSO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PSOR field descriptions

Field	Description
PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

40.3.3 Port Clear Output Register (FGPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTCO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PCOR field descriptions

Field	Description
PTCO	Port Clear Output Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

40.3.4 Port Toggle Output Register (FGPIOx_PTOR)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTTO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PTOR field descriptions

Field	Description
PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

40.3.5 Port Data Input Register (FGPIOx_PDIR)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PDIR field descriptions

Field	Description
PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

40.3.6 Port Data Direction Register (FGPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FGPIOx_PDDR field descriptions

Field	Description
PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

40.4 Functional description

40.4.1 General-purpose input

The logic state of each pin is available via the Port Data Input registers, provided the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

40.4.2 General-purpose output

The logic state of each pin can be controlled via the port data output registers and port data direction registers, provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input/output.

If	Then
A pin is configured for the GPIO function and the corresponding port data direction register bit is clear.	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding port data direction register bit is set.	The pin is configured as an output and and the logic state of the pin is equal to the corresponding port data output register.

To facilitate efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers exist to allow one or more outputs within one port to be set, cleared, or toggled from a single register write.

The corresponding Port Control and Interrupt module does not need to be enabled to update the state of the port data direction registers and port data output registers including the set/clear/toggle registers.

40.4.3 IOPORT

The GPIO registers are also aliased to the IOPORT interface on the Cortex-M0+ from address 0xF800_0000. Accesses via the IOPORT interface occur in parallel with any instruction fetches and will therefore complete in a single cycle. If the DMA attempts to access the GPIO registers on the same cycle as an IOPORT access, then the DMA access will stall until any IOPORT accesses have completed.

During Compute Operation, the GPIO registers remain accessible via the IOPORT interface only. Since the clocks to the Port Control and Interrupt modules are disabled during Compute Operation, the Pin Data Input Registers do not update with the current state of the pins.

Chapter 41
Touch Sensing Input (TSI)

The touch sensing input (TSI) module provides capacitive touch sensing detection with high sensitivity and enhanced robustness. Each TSI pin implements the capacitive measurement by a current source scan, charging and discharging the electrode, once or several times.

41.1 Introduction

The touch sensing input (TSI) module provides capacitive touch sensing detection with high sensitivity and enhanced robustness.

Each TSI pin implements the capacitive measurement by a current source scan, charging and discharging the electrode, once or several times. A reference oscillator ticks the scan time and stores the result in a 16-bit register when the scan completes. Meanwhile, an interrupt request is submitted to CPU for post-processing if TSI interrupt is enabled and DMA function is not selected.The TSI module can be periodically triggered to work in low power mode with ultra-low current adder and wake CPU at the end of scan or the conversion result is out of the range specified by TSI threshold. It provides a solid capacitive measurement module to the implementation of touch keyboard, rotaries and sliders.

41.1.1 Features

TSI features includes:

- Support up to 16 external electrodes
- Automatic detection of electrode capacitance across all operational power modes
- Internal reference oscillator for high-accuracy measurement
- Configurable software or hardware scan trigger
- Fully support Freescale touch sensing software (TSS) library, see www.freescale.com/touchsensing.
- Capability to wake MCU from low power modes

- Compensate for temperature and supply voltage variations
- High sensitivity change with 16-bit resolution register
- Configurable up to 4096 scan times.
- Support DMA data transfer
- The auxiliary noise detection mode supplies improved EMC immunity.

For electrode design recommendations, refer to AN3863: Designing Touch Sensing Electrodes

41.1.2 Modes of operation

This module supports the following operation modes.

Table 41-1. Operating modes

Mode	Description
Stop and low power stop	TSI module is fully functional in all of the stop modes as long as TSI_GENCS[STPE] is set. The channel specified by TSI_DATA[TSICH] will be scanned upon the trigger. After scan finishes, either end-of-scan or out-of-range interrupt can be selected to bring MCU out of low power modes.
Wait	TSI module is fully functional in this mode. When a scan completes, TSI submits an interrupt request to CPU if the interrupt is enabled.
Run	TSI module is fully functional in this mode. When a scan completes, TSI submits an interrupt request to CPU if the interrupt is enabled.

41.1.3 Block diagram

The following figure is a block diagram of the TSI module.

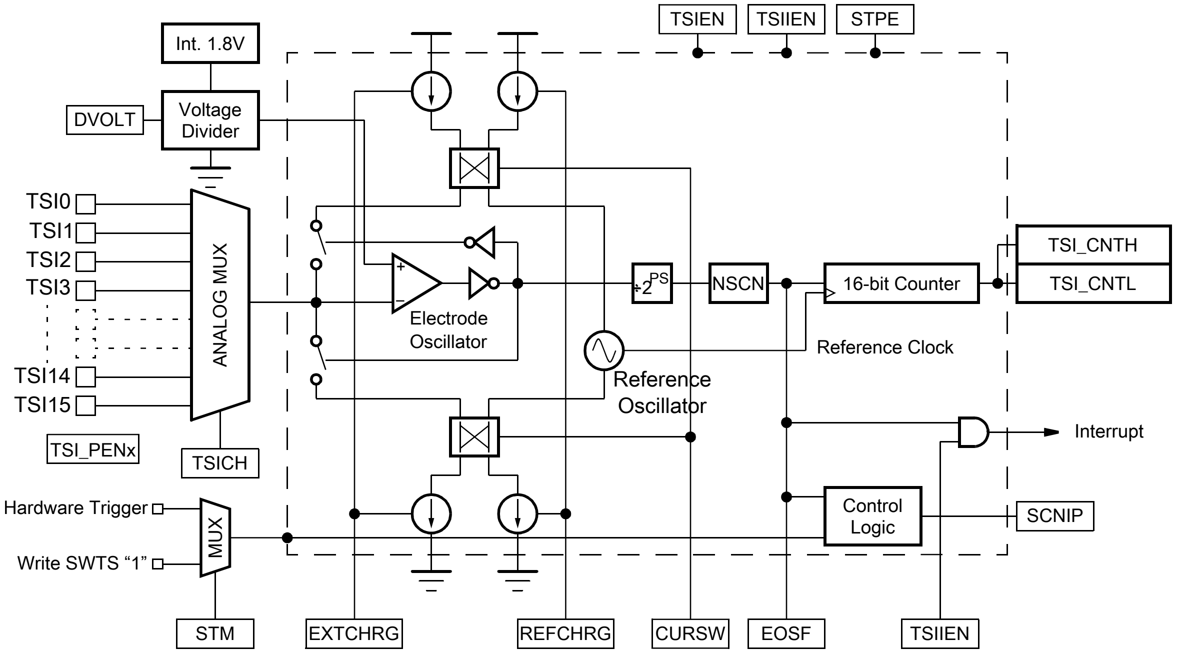


Figure 41-1. TSI module block diagram

41.2 External signal description

The TSI module contains up to 16 external pins for touch sensing. The table found here describes each of the TSI external pins.

Table 41-2. TSI signal description

Name	Port	Direction	Function	Reset state
TSI[15:0]	TSI	I/O	TSI capacitive pins. Switches driver that connects directly to the electrode pins TSI[15:0] can operate as GPIO pins.	I/O

41.2.1 TSI[15:0]

When TSI functionality is enabled , the TSI analog portion uses the corresponding channel to connect external on-board touch capacitors. The PCB connection between the pin and the touch pad must be kept as short as possible to reduce distribution capacity on board.

41.3 Register definition

This section describes the memory map and control/status registers for the TSI module.

TSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_5000	TSI General Control and Status Register (TSI0_GENCS)	32	R/W	0000_0000h	41.3.1/838
4004_5004	TSI DATA Register (TSI0_DATA)	32	R/W	0000_0000h	41.3.2/843
4004_5008	TSI Threshold Register (TSI0_TSHD)	32	R/W	0000_0000h	41.3.3/844

41.3.1 TSI General Control and Status Register (TSIx_GENCS)

This control register provides various control and configuration information for the TSI module.

NOTE

> When TSI is working, the configuration bits (GENCS[TSIEN], GENCS[TSIIEN], and GENCS[STM]) must not be changed. The EOSF flag is kept until the software acknowledge it.

Address: 4004_5000h base + 0h offset = 4004_5000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OUTRGF	0		ESOR	MODE				REFCHRG			DVOLT		EXTCHRG		
W	w1c	▨	▨													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PS			NSCN					TSIEN	TSIIEN	STPE	STM	SCNIP	EOSF	CURSW	0
W														w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_GENCS field descriptions

Field	Description
31 OUTRGF	Out of Range Flag. This flag is set if the result register of the enabled electrode is out of the range defined by the TSI_THRESHOLD register. This flag is set only when TSI is configured in non-noise detection mode. It can be read once the CPU wakes. Write "1" , when this flag is set, to clear it.
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 ESOR	End-of-scan or Out-of-Range Interrupt Selection This bit is used to select out-of-range or end-of-scan event to generate an interrupt. 0 Out-of-range interrupt is allowed. 1 End-of-scan interrupt is allowed.
27–24 MODE	TSI analog modes setup and status bits. Set up TSI analog modes, especially, setting MODE[3:2] to not 2'b00 will configure TSI to noise detection modes. MODE[1:0] take no effect on TSI operation mode and should always write to 2'b00 for setting up. When reading this field will return the analog status. Refer to chapter "Noise detection mode" for details. 0000 Set TSI in capacitive sensing(non-noise detection) mode. 0100 Set TSI analog to work in single threshold noise detection mode and the frequency limitation circuit is disabled. 1000 Set TSI analog to work in single threshold noise detection mode and the frequency limitation circuit is enabled to work in higher frequencies operations. 1100 Set TSI analog to work in automatic noise detection mode.
23–21 REFCHRG	REFCHRG These bits indicate the reference oscillator charge and discharge current value. 000 500 nA. 001 1 μA. 010 2 μA. 011 4 μA. 100 8 μA.

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	101 16 μA. 110 32 μA. 111 64 μA.
20–19 DVOLT	DVOLT These bits indicate the oscillator's voltage rails as below. 00 DV = 1.026 V; V_P = 1.328 V; V_m = 0.302 V. 01 DV = 0.592 V; V_P = 1.111 V; V_m = 0.519 V. 10 DV = 0.342 V; V_P = 0.986 V; V_m = 0.644 V. 11 DV = 0.197 V; V_P = 0.914 V; V_m = 0.716 V.
18–16 EXTCHRG	EXTCHRG These bits indicate the electrode oscillator charge and discharge current value. 000 500 nA. 001 1 μA. 010 2 μA. 011 4 μA. 100 8 μA. 101 16 μA. 110 32 μA. 111 64 μA.
15–13 PS	PS These bits indicate the prescaler of the output of electrode oscillator. 000 Electrode Oscillator Frequency divided by 1 001 Electrode Oscillator Frequency divided by 2 010 Electrode Oscillator Frequency divided by 4 011 Electrode Oscillator Frequency divided by 8 100 Electrode Oscillator Frequency divided by 16 101 Electrode Oscillator Frequency divided by 32 110 Electrode Oscillator Frequency divided by 64 111 Electrode Oscillator Frequency divided by 128
12–8 NSCN	NSCN These bits indicate the scan number for each electrode. The scan number is equal to NSCN + 1, which allows the scan time ranges from 1 to 32. By default, NSCN is configured as 0, which asserts the TSI scans once on the selected eletrode channel. 00000 Once per electrode 00001 Twice per electrode 00010 3 times per electrode 00011 4 times per electrode 00100 5 times per electrode 00101 6 times per electrode 00110 7 times per electrode 00111 8 times per electrode

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	01000 9 times per electrode 01001 10 times per electrode 01010 11 times per electrode 01011 12 times per electrode 01100 13 times per electrode 01101 14 times per electrode 01110 15 times per electrode 01111 16 times per electrode 10000 17 times per electrode 10001 18 times per electrode 10010 19 times per electrode 10011 20 times per electrode 10100 21 times per electrode 10101 22 times per electrode 10110 23 times per electrode 10111 24 times per electrode 11000 25 times per electrode 11001 26 times per electrode 11010 27 times per electrode 11011 28 times per electrode 11100 29 times per electrode 11101 30 times per electrode 11110 31 times per electrode 11111 32 times per electrode
7 TSIEN	Touch Sensing Input Module Enable This bit enables TSI module. 0 TSI module disabled. 1 TSI module enabled.
6 TSIIEN	Touch Sensing Input Interrupt Enable This bit enables TSI module interrupt request to CPU when the scan completes. The interrupt will wake MCU from low power mode if this interrupt is enabled. 0 TSI interrupt is disabled. 1 TSI interrupt is enabled.
5 STPE	TSI STOP Enable This bit enables TSI module function in low power modes (stop, VLPS, LLS and VLLS{3,2,1}). 0 TSI is disabled when MCU goes into low power mode. 1 Allows TSI to continue running in all low power modes.
4 STM	Scan Trigger Mode This bit specifies the trigger mode. User is allowed to change this bit when TSI is not working in progress. 0 Software trigger scan. 1 Hardware trigger scan.

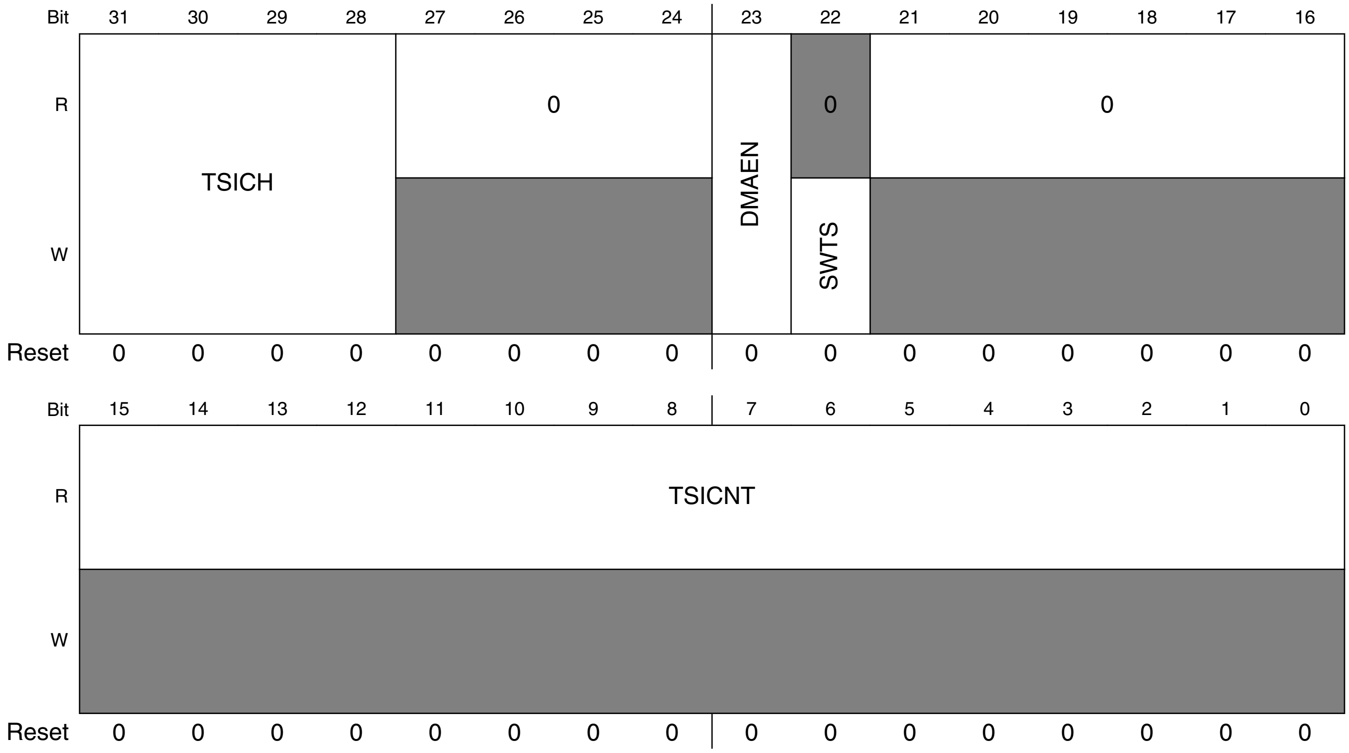
Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
3 SCNIP	Scan In Progress Status This read-only bit indicates if scan is in progress. This bit will get asserted after the analog bias circuit is stable after a trigger and it changes automatically by the TSI. 0 No scan in progress. 1 Scan in progress.
2 EOSF	End of Scan Flag This flag is set when all active electrodes are finished scanning after a scan trigger. Write "1" , when this flag is set, to clear it. 0 Scan not complete. 1 Scan complete.
1 CURSW	CURSW This bit specifies if the current sources of electrode oscillator and reference oscillator are swapped. 0 The current source pair are not swapped. 1 The current source pair are swapped.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

41.3.2 TSI DATA Register (TSIx_DATA)

Address: 4004_5000h base + 4h offset = 4004_5004h



TSIx_DATA field descriptions

Field	Description
31–28 TSICH	TSICH These bits specify current channel to be measured. In hardware trigger mode (TSI_GENCS[STM] = 1), the scan will not start until the hardware trigger occurs. In software trigger mode (TSI_GENCS[STM] = 0), the scan starts immediately when TSI_DATA[SWTS] bit is written by 1. 0000 Channel 0. 0001 Channel 1. 0010 Channel 2. 0011 Channel 3. 0100 Channel 4. 0101 Channel 5. 0110 Channel 6. 0111 Channel 7. 1000 Channel 8. 1001 Channel 9. 1010 Channel 10. 1011 Channel 11. 1100 Channel 12. 1101 Channel 13.

Table continues on the next page...

TSIx_DATA field descriptions (continued)

Field	Description
	1110 Channel 14. 1111 Channel 15.
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 DMAEN	DMA Transfer Enabled This bit is used together with the TSI interrupt enable bits(TSIIE, ESOR) to generate a DMA transfer request instead of an interrupt. 0 Interrupt is selected when the interrupt enable bit is set and the corresponding TSI events assert. 1 DMA transfer request is selected when the interrupt enable bit is set and the corresponding TSI events assert.
22 SWTS	Software Trigger Start This write-only bit is a software start trigger. When STM bit is clear, write "1" to this bit will start a scan. The electrode channel to be scanned is determinated by TSI_DATA[TSICH] bits. 0 No effect. 1 Start a scan to determine which channel is specified by TSI_DATA[TSICH].
21–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TSICNT	TSI Conversion Counter Value These read-only bits record the accumulated scan counter value ticked by the reference oscillator.

41.3.3 TSI Threshold Register (TSIx_TSHD)

Address: 4004_5000h base + 8h offset = 4004_5008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	THRESH																THRESL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_TSHD field descriptions

Field	Description
31–16 THRESH	TSI Wakeup Channel High-threshold This half-word specifies the high threshold of the wakeup channel.
THRESL	TSI Wakeup Channel Low-threshold This half-word specifies the low threshold of the wakeup channel.

41.4 Functional description

41.4.1 Capacitance measurement

The electrode pin capacitance measurement uses a dual oscillator approach. The frequency of the TSI electrode oscillator depends on the external electrode capacitance and the TSI module configuration. After going to a configurable prescaler, the TSI electrode oscillator signal goes to the input of the module counter. The time for the module counter to reach its module value is measured using the TSI reference oscillator. The measured electrode capacitance is directly proportional to the time.

41.4.1.1 TSI electrode oscillator

The TSI electrode oscillator circuit is illustrated in the following figure. A configurable constant current source is used to charge and discharge the external electrode capacitance. A buffer hysteresis defines the oscillator delta voltage. The delta voltage defines the margin of high and low voltage which are the reference input of the comparator in different time.

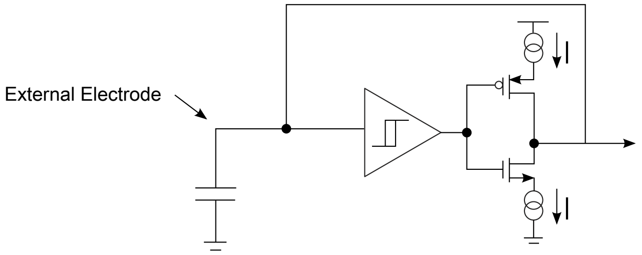


Figure 41-2. TSI electrode oscillator circuit

The current source applied to the pad capacitance is controlled by the GENCS[EXTCHRG]. The hysteresis delta voltage is defined in the module electrical specifications present in the device Data Sheet. The figure below shows the voltage amplitude waveform of the electrode capacitance charging and discharging with a programmable current.

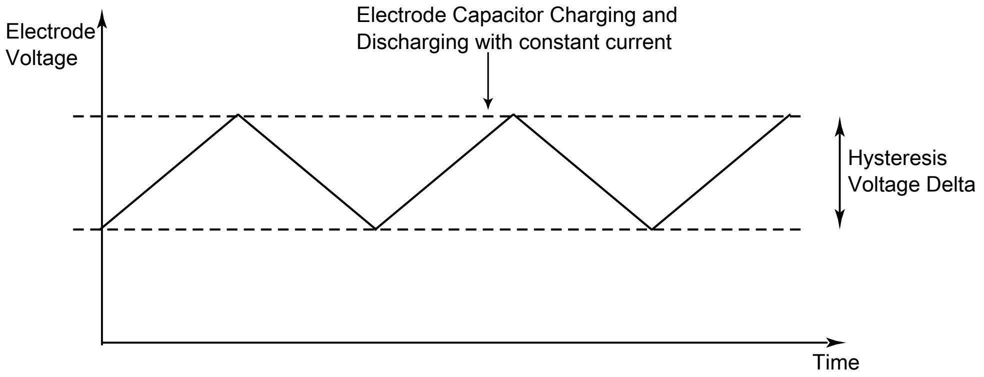

Electrode Voltage
Electrode Capacitor Charging and Discharging with constant current
Hysteresis Voltage Delta
Time

Figure 41-3. TSI electrode oscillator chart

The oscillator frequency is give by the following equation

$$F_{elec} = \frac{I}{2 * C_{elec} * \Delta V}$$

Equation 4. TSI electrode oscillator frequency

Where:

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

So by this equation, for example, an electrode with C_{elec}= 20 pF, with a current source of I = 16 μA and ΔV = 600 m V have the following oscillation frequency:

$$F_{elec} = \frac{16\ \mu A}{2 * 20pF * 600mV} = 0.67MHz$$

Equation 5. TSI electrode oscillator frequency

The current source is used to accommodate the TSI electrode oscillator frequency with different electrode capacitance sizes.

41.4.1.2 Electrode oscillator and counter module control

The TSI oscillator frequency signal goes through a prescaler defined by the GENCS[PS] and then enters in a modulus counter. GENCS[NSCN] defines the maximum count value of the modulus counter.

The pin capacitance sampling time is given by the time the module counter takes to go from 0 to its maximum value, defined by NSCN. The electrode sample time is expressed by the following equation:

$$T_{cap\_samp} = \frac{PS * NSCN}{F_{elec}}$$

Using Equation 1

$$T_{cap\_samp} = \frac{2 * PS * NSCN * C_{elec} * \Delta V}{I}$$

Equation 6. Electrode sampling time

Where:

PS: prescaler value

NSCN: module counter maximum value

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

By this equation we have that an electrode with C = 20 pF, with a current source of I = 16 µA and ΔV = 600 mV, PS = 2 and NSCN = 16 have the following sampling time:

$$T_{cap\_samp} = \frac{2*2*16*20pF*600mV}{16\mu A} = 48\mu s$$

41.4.1.3 TSI reference oscillator

The TSI reference oscillator has the same topology of the TSI electrode oscillator. The TSI reference oscillator instead of using an external capacitor for the electrode oscillator has an internal reference capacitor.

The TSI reference oscillator has an independent programmable current source controlled by GENCS[REFCHRG].

The reference oscillator frequency is given by the following equation:

$$F_{ref\_osc} = \frac{I_{ref}}{2 * C_{ref} * \Delta V}$$

Equation 7. TSI reference oscillator frequency

Where:

C_{ref}: Internal reference capacitor

I_{ref}: Reference oscillator current source

ΔV : Hysteresis delta voltage

Considering $C_{ref} = 1.0$ pF, $I_{ref} = 12$ µA and $\Delta V = 600$ mV, follows

$$F_{ref\_osc} = \frac{12\mu A}{2 * 1.0pF * 600mV} = 10.0MHz$$

41.4.2 TSI measurement result

The capacitance measurement result is defined by the number of TSI reference oscillator periods during the sample time and is stored in the TSICHnCNT register.

$\mathrm{TSICHnCNT} = T_{cap\_samp} * F_{ref\_osc}$

Using Equation 2 and Equation 1 follows:

$$\mathrm{TSICHnCNT} = \frac{I_{ref} * PS * NSCN}{C_{ref} * I_{elec}} * C_{elec}$$

Equation 8. Capacitance result value

In the example where $F_{ref\_osc} = 10.0$ MHz and $T_{cap\_samp} = 48$ µs, TSICHnCNT = 480

41.4.3 Enable TSI module

The TSI module can be fully functional in run, wait and low power modes. The TSI_GENCS[TSIEN] bit must be set to enable the TSI module in run and wait mode. When TSI_GENCS[STPE] bit is set, it allows the TSI module to work in low power mode.

41.4.4 Software and hardware trigger

The TSI module allows a software or hardware trigger to start a scan. When a software trigger is applied (TSI_GENCS[STM] bit clear), the TSI_GENCS[SWTS] bit must be written "1" to start the scan electrode channel that is identified by TSI_DATA[TSICH]. When a hardware trigger is applied (TSI_GENCS[STM] bit set), the TSI will not start scanning until the hardware trigger arrives. The hardware trigger is different depending on the MCU configuration. Generally, it could be an event that RTC overflows. See chip configuration section for details.

41.4.5 Scan times

The TSI provides multi-scan function. The number of scans is indicated by TSI_GENCS[NSCN] that allow the scan number from 1 to 32. When TSI_GENCS[NSCN] is set to 0 (only once), the single scan is engaged. The 16-bit counter accumulates all scan results until the NSCN time scan completes, and users can read TSI_DATA[TSICNT] to get this accumulation. When DMA transfer is enabled, the counter values can also be read out by DMA engine.

41.4.6 Clock setting

TSI is built with dual oscillator architecture. In normal sensing application, the reference oscillator clock is the only clock source for operations. The reference clock is used to measure the electrode oscillator by ticking a 16-bit counter. The reference oscillator frequency depends on the current source setting. Please refer to the Current source for more details.

The output of electrode oscillator has several prescalers up to 128 indicated by TSI_GENCS[PS]. This allows a flexible counter configuration for different electrode oscillator frequency.

41.4.7 Reference voltage

The TSI module offers a internal reference voltage for both electrode oscillator and reference oscillator. The internal reference voltage can work in low power modes even when the MCU regulator is partially powered down, which is ideally for low-power touch detection.

The charge and discharge difference voltage is configurable upon the setting of TSI_GENCS[DVOLT]. The following table shows the all the delta voltage configurations.

NOTE

This table doesn't apply to noise mode, see noise mode sections for its configuration.

Table 41-3. Delta voltage configuration

DVOLT	V_p (V)	V_m (V)	ΔV (V)
00	1.328	0.302	1.026

Table continues on the next page...

Table 41-3. Delta voltage configuration (continued)

DVOLT	V_p (V)	V_m (V)	ΔV (V)
01	1.111	0.519	0.592
10	0.986	0.644	0.342
11	0.914	0.716	0.198

41.4.8 Current source

The TSI module supports eight different current source power to increment from 500 nA to 64 µA. TSI_GENCS[EXTCHRG] determines the current of electrode oscillator that charges and discharges external electrodes. The TSI_GENCS[REFCHRG] determines the current of reference oscillator on which the internal reference clock depends. The lower current source takes more time for charge and discharge, which is useful to detect high-accuracy change. The higher current source takes less time, which can be used to charge a big electrode by less power consumption.

TSI_GENCS[CURSW] allows the current source to swap, so that the reference oscillator and electrode oscillator use the opposite current sources. When TSI_GENCS[CURSW] is set and the current sources are swapped, TSI_GENCS[EXTCHRG] and TSI_GENCS[REFCHRG] still control the corresponding current sources, that is, TSI_GENCS[EXTCHRG] controls the reference oscillator current and TSI_GENCS[REFCHRG] controls the electrode oscillator current.

41.4.9 End of scan

As a scan starts, [SCNIP] bit is set to indicate scan is in progress. When the scan completes, the [EOSF] bit is set. Before clearing the [EOSF] bit, the value in TSI_DATA[TSICNT] must be read. If the TSI_GENCS[TSIIEN] and TSI_GENCS[ESOR] are set andTSI_GENCS[DMAEN]is not set, an interrupt is submitted to CPU for post-processing immediately. The interrupt is also optional to wake MCU to execute ISR if it is in low power mode. When DMA function is enabled by setting TSI_GENCS[TSIIEN] and TSI_GENCS[ESOR], as soon as scan completes, a DMA transfer request is asserted to DMA controller for data movement, generally, DMA engine will fetch TSI conversion result from TSI_DATA register,store it to other memory space and then refresh the TSI scan channel index(TSI_DATA[TSICH]) for next loop. When DMA transfer is done, TSI_GENCS[EOSF] is cleared automatically.

41.4.10 Out-of-range interrupt

If enabled, TSI will scan the electrode specified by TSI_DATA[TSICH] as soon as the trigger arrives. The TSI_GENCS[OUTRGF] flag generates a TSI interrupt request if the TSI_GENCS[TSIIE] bit is set and GENCS[ESOR] bit is cleared. With this configuration, after the end-of-electrode scan, the electrode capacitance will be converted and stored to the result register TSI_DATA[TSICNT], the out-of-range interrupt is only requested if there is a considerable capacitance change defined by the TSI_TSHD. For instance, if in low power mode the electrode capacitance does not vary, the out-of-range interrupt does not interrupt the CPU. This interrupt will not happen in noise detection mode. It is worthy to note that when the counter value reaches 0xFFFF is treated as an extreme case the out-of-range will not happen. Also in noise detection mode, the out-of-range will not assert either.

41.4.11 Wake up MCU from low power modes

In low power modes, once enabled by TSI_GENCS[STPE] and TSI_GENCS[TSIIE], TSI can bring MCU out of its low power modes(STOP, VLPS, VLLS,etc) by either end of scan or out of range interrupt, that is, if TSI_GENCS[ESOR] is set, end of scan interrupt is selected and otherwise, out of range is selected.

41.4.12 DMA function support

Transmit by DMA is supported only when TSI_DATA[DMAEN] is set. A DMA transfer request is asserted when all the flags based on TSI_GENCS[ESOR] settings and TSI_GENCS[TSIIE] are set. Then the on-chip DMA controller detects this request and transfers data between memory space and TSI register space. After the data tranfer, DMA DONE is asserted to clear TSI_GENCS[EOSF] automatically. This function is normally used by DMA controller to get the conversion result from TSI_DATA[TSICNT] upon a end-of-scan event and then refresh the channel index(TSI_DATA[TSICH]) for next trigger.

41.4.13 Noise detection mode

The noise detection mode is used to detect power of noise. In this mode the thresholds are incremented internally by TSI until the point that there is no noise voltage trepassing the threshold.

The noise detection mode change the circuit configuration as shown in the following figure. With this configuration, it is possible to detect touch with high levels of EMC noise present. To enter this mode, set TSI_CS3[STAT_STUP] field to 1100b.

In noise detection mode the reference oscillator has the same configuration except the output goes to Counter2 and this counter will have its maximum count set by NSCNx2$^{(PS)}$. This means this oscillator will setup the noise detection mode sense duration as shown in Figure 41-4.

The blocks of external oscillator is changed and instead of an oscillator the circuit implements an RF amplitude detection. The threshold for this amplitude detection is set by DVOLT register bits. Be noted There is no oscillation on external pad (just if it is caused by external noise) in this mode.

Also the external voltage is biased by vmid voltage with a Rs series resistance.

The vmid voltage is defined as V(vmid) = (V(vp) + V(vm))/2.

The Rs value is defined by TSI_CS2[EXTCHRG] register bits. See Figure 41-5 for more information on noise mode TSI circuit.

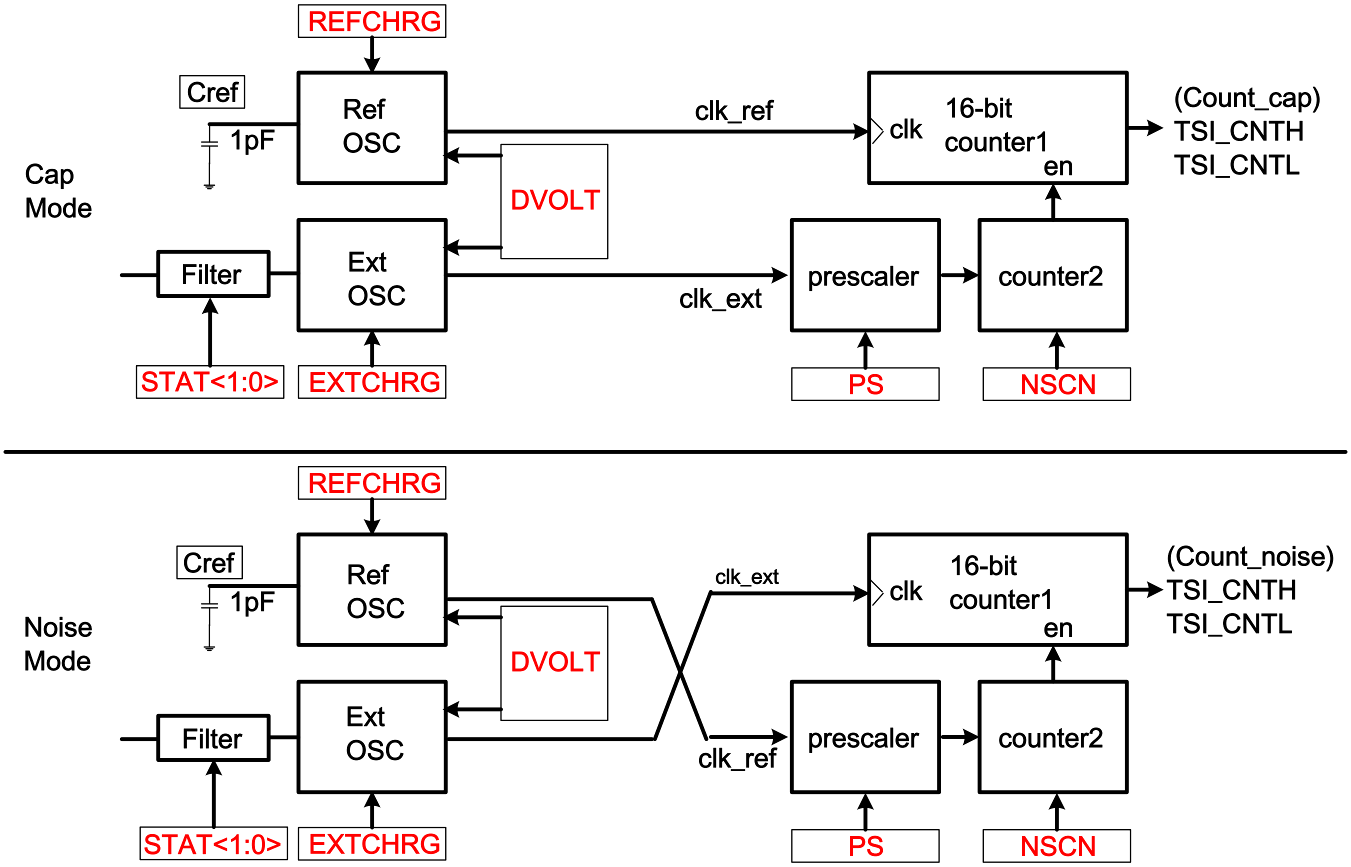


Figure 41-4. TSI noise detection mode block diagram

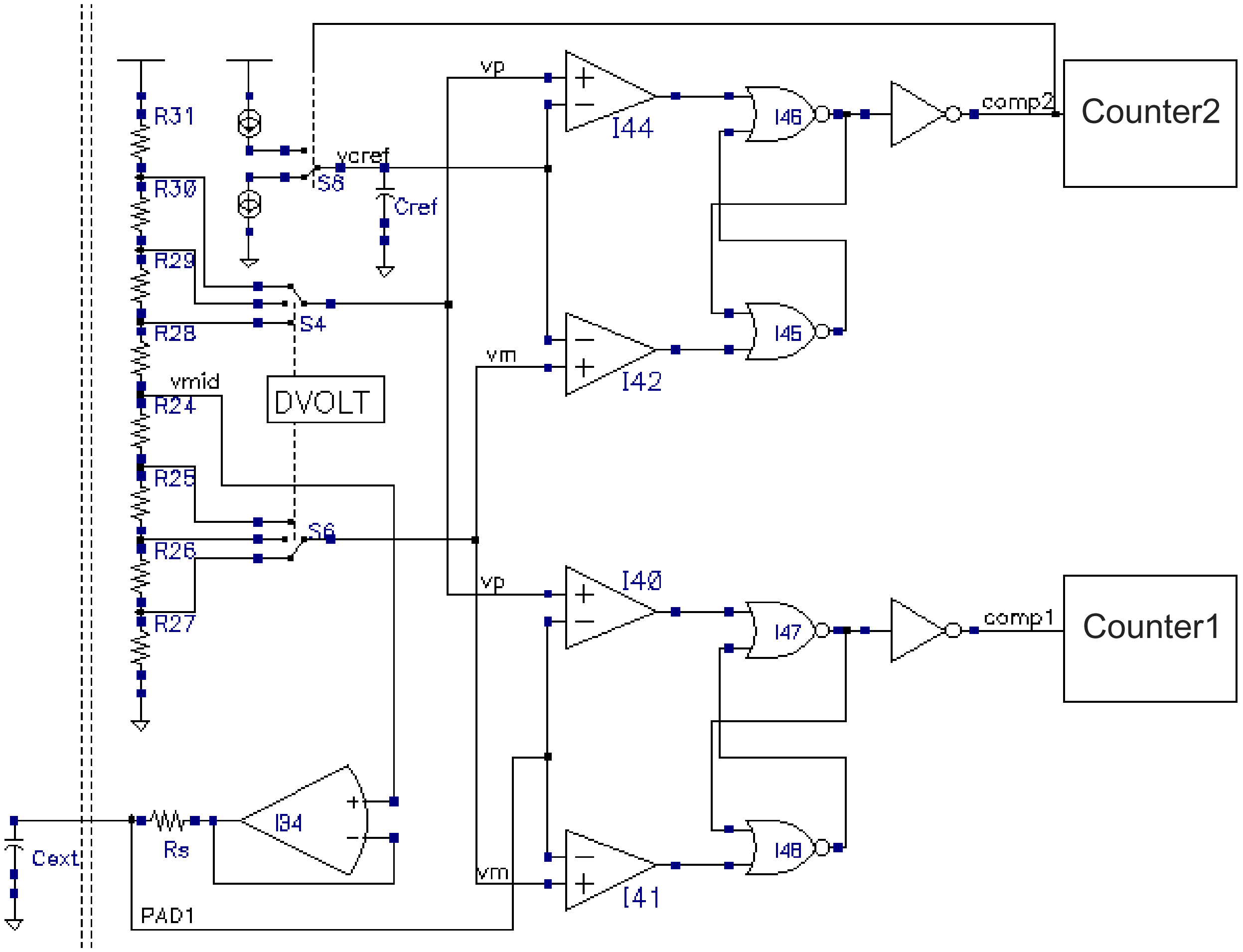
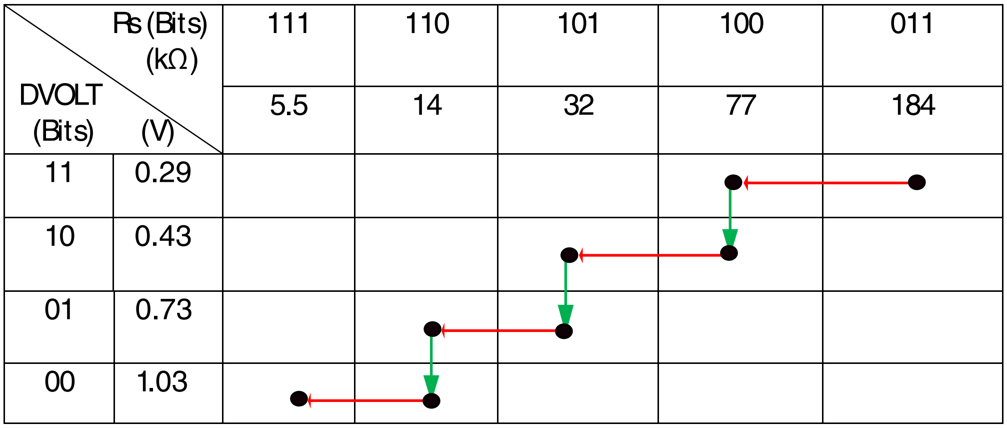


Figure 41-5. TSI circuit in noise detection mode

The table below shows all DVOLT values both in Bits and in V, all Rs values both in EXTCHRG Bits and in kΩ in order. The values indicated by valid black points can be used. The valid Rs/Dvolt values are: 184K/0.29V, 77K/0.29V, 77K/0.43V, 32K/0.43V, 32K/0.73V, 14K/0.73V, 14K/1.03V, 5K/1.03V

To determine the noise level, the TSI noise detection algorithm shall be performed by scanning this table following the arrow direction starting at maximum Rs and minimum DVOLT.



DVOLT (Bits)	Rs (Bits) (kΩ) (V)	111	110	101	100	011
		5.5	14	32	77	184
11	0.29				●	●
10	0.43			●	●	
01	0.73		●	●		
00	1.03	●	●			

The TSI noise detection algorithm shall be done by application software (assume software poll method is used to check TSI scan status) as described below, note the values in below steps are just used to illustrated the algorithm flow, the actual value should be consistent with the valid combinations as shown in the table above.

1. Enable TSI by setting TSI_CS0[TSIEN] bit and select a channel by writing a channel number to TSI_CS3[TSICH] just as does for normal function mode;
2. Enable noise detection mode by writing 11b to TSI_CS3[STAT_STUP[3:2]] (STAT_STUP bit 3 and 2 with 11);
3. Initialize the noise RF amplitude and noise detection mode sense duration (T) as below:
   a. Initialize Rs to the max value by writing 011b to TSI_CS2[EXTCHRG] and DVOLT to the minimum value by writing 11b to TSI_CS2[DVOLT];
   b. Set up TSI_CS2[REFCHRG], TSI_CS1[PS] and TSI_CS1[NSCN] bits to set the noise detection mode sense duration (T). T= (2 x (2^{PS}) x NSCN x Cref x ΔV)/ Iref .

NOTE

NOTE: This time needs to be enough to detect the number of WINDOW bits for the minimum noise frequency. The minimum value of T (Tmin) is calculated as below: Tmin = (WINDOW+1)/F_{noise_min}; Where F_{noise_min} is the minimum noise frequency (0.15 MHz) and WINDOW is 2. This results in Tmin = 20 µs. Also this algorithm needs to be consistent with the valid Rs/Dvold combinations in above table.

4. Start TSI scan with software trigger or hardware trigger just as does for normal function mode
5. Wait until TSI scan is complete (TSI_CS0[EOSF] = 1);
6. Read TSI counter value in TSI_CNTH:TSI_CNTL and then clear TSI_CS0[EOSF] flag;

7. Check whether the TSI counter value is within the given counter window (WINDOW, can be 2 or 3 or 5): If the TSI counter value < WINDOW (i.e., the noise level is detected), go to step 12; Otherwise continue with the next step (meaning the noise level is too large);
8. If (Rs = minimum value) (i.e., TSI_CS2[EXTCHRG] =000b, noise level is the largest at given DVOLT), go to step 10; otherwise continue with the next step;
9. Reduce Rs value by incrementing TSI_CS2[EXTCHRG] by 1 and then go to step 4; (This action detects the next high level of noise)
10. If (DVOLT = maximum value) (i.e., TSI_CS2[DVOLT] =00b), this means noise is too large to detect, go to END; otherwise continue with the next step;
11. Increase DVOLT to the next level by decrementing TSI_CS2[DVOLT] by 1 and set Rs to the max value, then go to step 4; (It means noise level is higher, so need find high DVOLT)
12. Reduce Rs value by incrementing TSI_CS2[EXTCHRG] by 1 if (Rs > minimum value) (i.e., TSI_CS2[EXTCHRG] < 111b), and then go to END (Now a matching DVOLT corresponding to the noise level is found)
13. Reduce DVOLT by incrementing TSI_CS2[DVOLT] by 1 if (Rs = maximum value) (i.e.,TSI_CS2[EXTCHRG] = 011b); (Now a matching DVOLT corresponding to the noise level is found)
14. END:

NOTE

The END condition of above algorithm can be one of

- TSI counter value within the WINDOW and Rs ≥ minimum value
- TSI counter value out of the WINDOW and Rs = minimum value and DVOLT = maximum value

At the end of the above steps, the correct matching DVOLT value and the electrode oscillator charge and discharge current value for the current noise level is found. That is, the correct TSI_CS2[DVOLT] value and TSI_CS2[EXTCHRG] value are found for the current noise level. And now users can proceed with normal capacitive sense procedure by keeping both TSI_CS[DVOLT] and TSI_CS2[EXTCHRG] untouched, that is, users just need switch to normal capacitive sense mode by clearing TSI_CS3[STAT_STUP[3:2]] bits and start TSI scan.

For typical applications, the noise detection/sense algorithm shall be performed first followed by normal capacitive sense for a given channel and then alternate between noise sense and capacitive sense as shown in Figure 41-6.

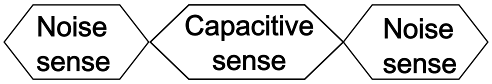


Figure 41-6. Noise detection/sense algorithm of typical application

The following flow chart shows how to detect touch with noise sense and normal capacitive sense.

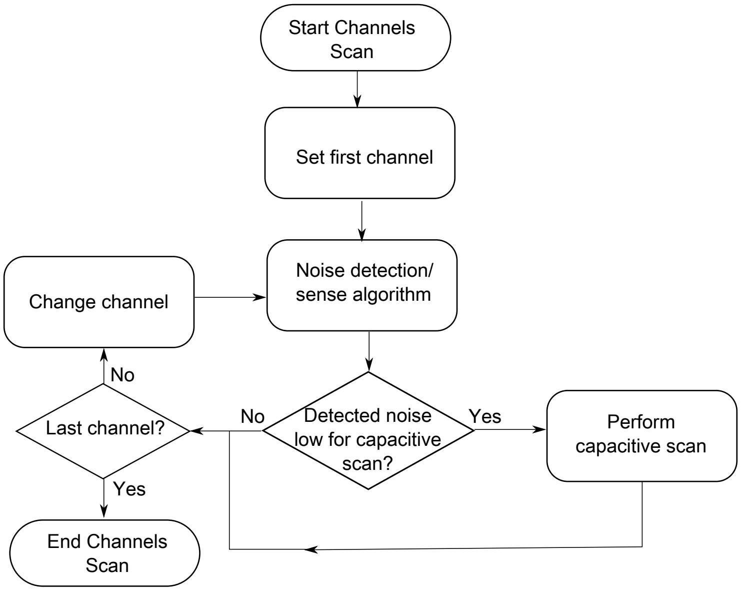


Figure 41-7. Detection of touch with noise sense and normal capacitive sense flow chart

One example of noise detection mode is shown in the following figure. in this figure the TSI is working in capacitive mode until 28 µs (T1) when it is changed to noise detection mode. In noise detection mode the selected pad is biased with 0.815V and all AC waveform in this pad is caused by a noise source external to the MCU.

It is possible to observe in the following figure that, in noise detection mode, the clkref output has the peak detection and the number of detected peaks can be counted or used by digital block. The clkext output has the internal oscillator output and can be used to set the maximum noise detection time window.

The waveform of the following figure shows two operations during noise detection mode. Again, this waveform is captured from Freescale internal design simulation data, the actual useful points for noise detection should be consisted with the table provided above.

- The V(vp) and V(vm) thresholds are changed in 34.4 µs (T2).
- The Rs series resistance value is changed between 184 kΩ (TSI_CS2[EXTCHRG]=011b), 77 kΩ (TSI_CS2[EXTCHRG]=100) and 32 kΩ(TSI_CS2[EXTCHRG] = 101). Because of this Rs change the amplitude of noise waveform change also.

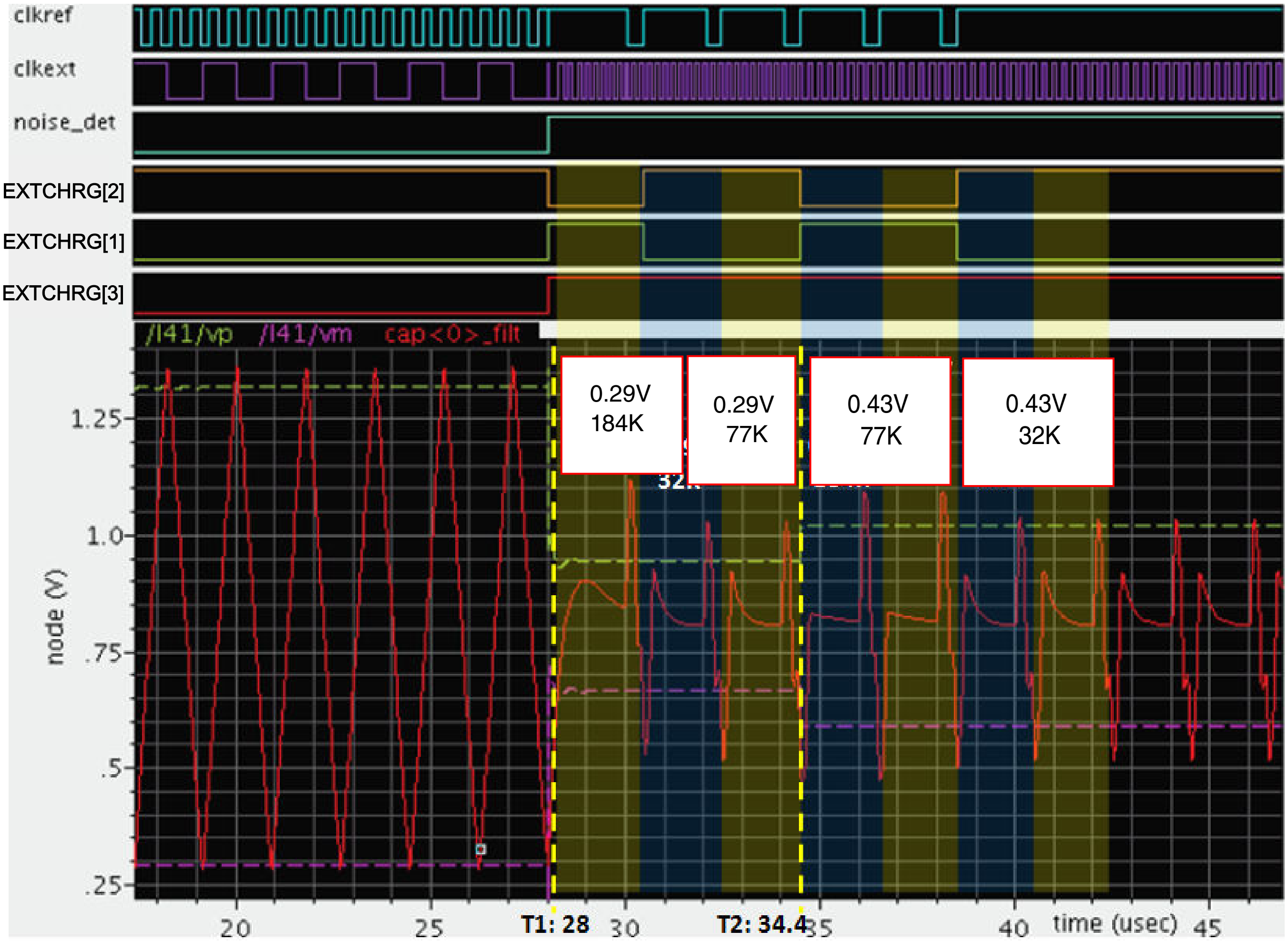


Figure 41-8. TSI noise detection mode waveform

41.4.13.1 Automatic noise mode

This mode is set by MODE[3:2] = 11 (noise mode 3). In this mode, the thresholds are incremented internally by the module until the point that there is no noise voltage trepassing the threshold.

The following diagram shows how it is done. The threshold comparator output goes to a counter and as the DVOLT control bits are increased the DVOLT thresholds are increased as well. The four bits are counted until 1111 (=15) and the counter is stop with this maximum value.

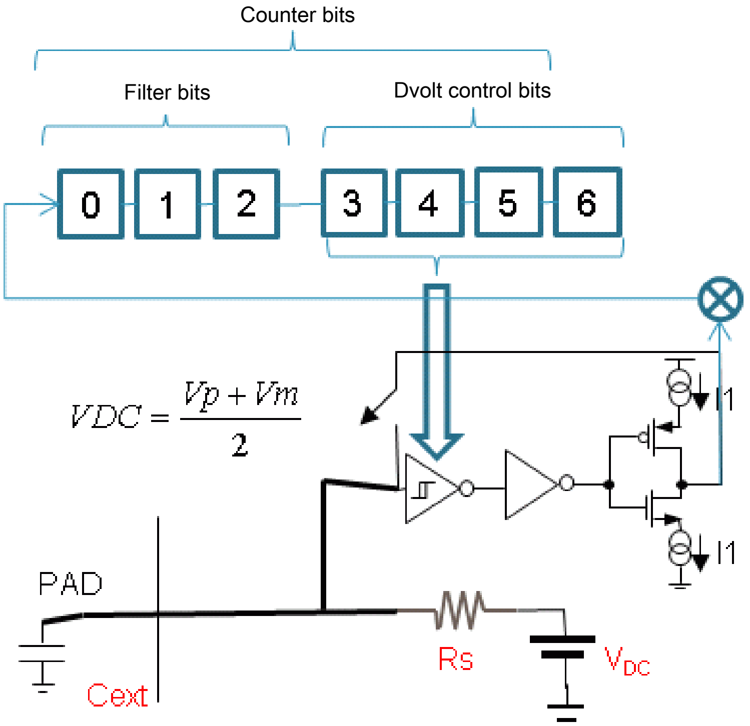


Figure 41-9. Block diagram automatic noise threshold operation

The signals that have different behavior in this noise mode (wrt capacitive mode) are shown in the following table.

Table 41-4. Signal properties in automatic noise operation mode

Name	Function	I/O type	Power Up/Reset state
MODE[3:2]	11—Noise mode operation with frequency limitation and automatic threshold counter.	I	00
EXTCHRG[2:1]	In this operation mode, these bits select the number of filter bits. 00—3 filter bits 01—2 filter bits 10—1 filter bit 11—no filter bit	I	00
EXTCHRG[0]	In this operation mode, this bit selects the series resistance. 0—uses Rs=32 kΩ 1—uses Rs=187 kΩ Independent of this bit selection, the threshold 15 is done with Rs = 5.5 kΩ	I	0

Table continues on the next page...

Table 41-4. Signal properties in automatic noise operation mode (continued)

Name	Function	I/O type	Power Up/Reset state
DVOLT[1:0]	Selects voltage rails of the internal oscillator	I	00
MODE[3:0]	DVOLT counter bits output. This field keeps 0000b if MODE[3:2] is not 11 after entering automatic noise mode.	O	0000

41.4.13.2 Single threshold noise modes

These modes are reset by MODE[3:2]=01 and 10.

In this mode, the thresholds are set by user via register bits as described in the following table.

During this mode the internal oscillator rails are set to the maximum (equivalent to DVOLT[1:0]=00).

Table 41-5. Signal properties in single noise modes (1,2)

Name	Function	I/O type	Power up / reset
MODE[3:2]	01 or 10- Single thrshold noise mode operation.	I	00
DVOLT[1:0], EXTCHRG[2:1]	In this operation mode these 4 bits are used select the noise threshold. These combinations are the maximum possible combinations, however, in real application, only the valid combinations in the above table should be used. 0000 - DVpm = 0.038 V, Vp = 0.834 V, Vm = 0.796 V 0001 - DVpm = 0.050 V, Vp = 0.830 V, Vm = 0.790 V 0010 - DVpm = 0.066 V, Vp = 0.848 V, Vm = 0.782 V 0011 - DVpm = 0.087 V, Vp = 0.858 V, Vm = 0.772 V 0100 - DVpm = 0.114 V, Vp = 0.872 V, Vm = 0.758 V 0101 - DVpm = 0.150 V, Vp = 0.890 V, Vm = 0.740 V 0110 - DVpm = 0.197 V, Vp = 0.914 V, Vm = 0.716 V 0111 - DVpm = 0.260 V, Vp = 0.945 V, Vm = 0.685 V 1000 - DVpm = 0.342 V, Vp = 0.986 V, Vm = 0.644 V 1001 - DVpm = 0.450 V, Vp = 1.040 V, Vm = 0.590 V 1010 - DVpm = 0.592 V, Vp = 1.111 V, Vm = 0.519 V 1011 - DVpm = 0.780 V, Vp = 1.205 V, Vm = 0.425 V 1100 - DVpm = 1.026 V, Vp = 1.328 V, Vm = 0.302 V 1101 - DVpm = 1.350 V, Vp = 1.490 V, Vm = 0.140 V	I	XXXX

Table continues on the next page...

Table 41-5. Signal properties in single noise modes (1,2) (continued)

Name	Function	I/O type	Power up / reset
	1110 - DVpm = 1.630 V, Vp = 1.630 V, Vm = 0 V 1111 - DVpm = 1.630 V, Vp = 1.630 V, Vm = 0 V		
EXTCHRG[0]	In this operation mode this bits selects the series resistance. 0 - uses Rs = 32 kΩ. 1- uses Rs = 187 kΩ. Independent of this bit selection the threshold 15 is done with Rs = 5.5 kΩ.	I	XX

Chapter 42
LP Trusted Cryptography (LTC)

42.1 LP Trusted Cryptography Block Diagram

LP Trusted Cryptography is an architecture that allows multiple cryptographic hardware accelerator engines to be instantiated and share common registers. This version of LTC only supports AES. The following figure presents a top-level diagram of the LP Trusted Cryptography module with an AES engine.

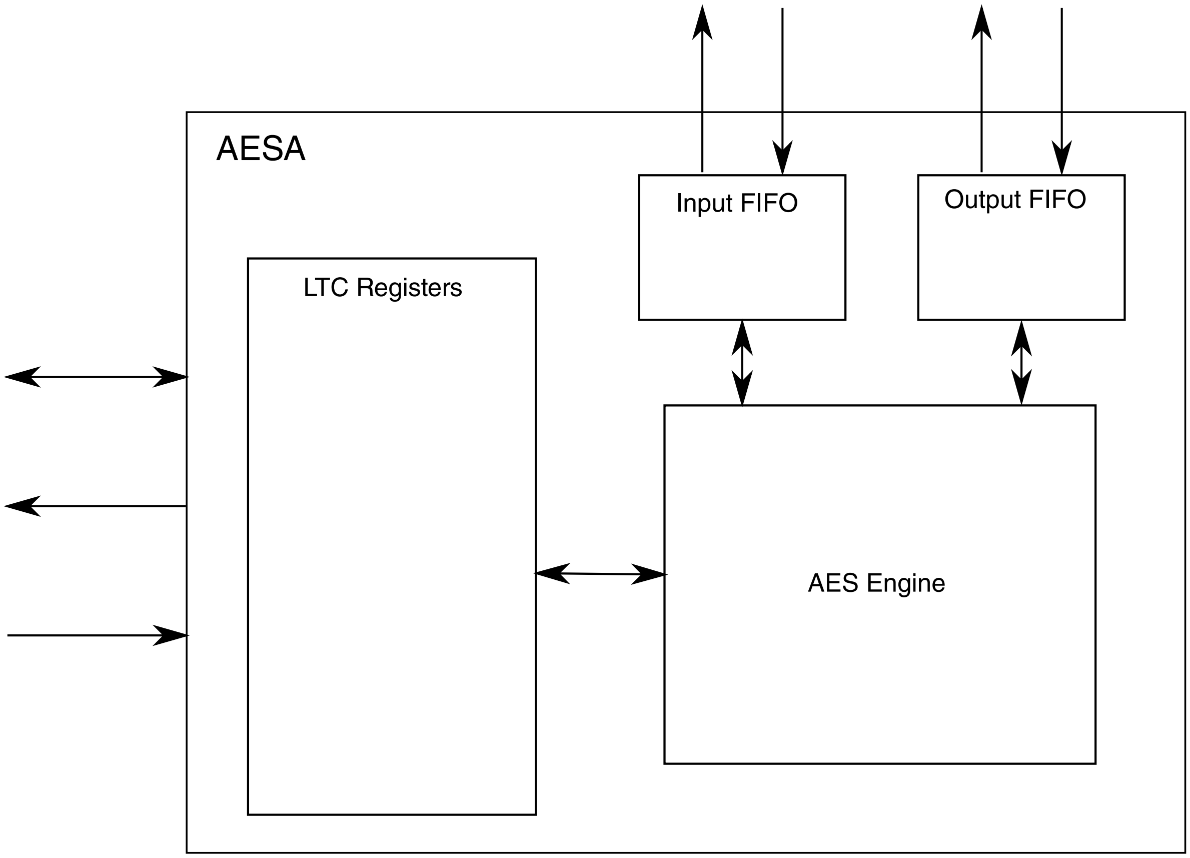


Figure 42-1. LTC Block Diagram

42.2 Feature summary

LTC includes the following features:

- Cryptographic authentication
    - Message authentication codes (MAC)
        - AES-CMAC
        - AES-XCBC-MAC
    - Auto padding
    - ICV checking
- Authenticated encryption algorithms
    - AES-CCM (counter with CBC-MAC)
    - AES-GCM (Galois counter mode)
- Symmetric key block ciphers
    - AES (128-bit keys)
    - Cipher modes
        - ECB, CBC, CTR for AES
        - ECB, CBC, CFB, OFB for DES
- Secure Scan

42.3 AES accelerator (AESA) functionality

The advanced encryption standard accelerator (AESA) module is a hardware co-processor capable of accelerating the advanced encryption standard (AES) cryptographic algorithm.

42.3.1 Differences between the AES encrypt and decrypt keys

The decrypt form of the key is different from the encrypt form of the key, because AES successively modifies the cryptographic key during the steps of the cryptographic operation. The decryption operation yields the correct result only if the modified form of the key (the decrypt key) is used at the beginning of the decryption operation. Unless told otherwise (via the DK bit in the Mode Register), AES assumes that a key loaded from memory is the encrypt key, that is, the form appropriate for encryption. If a decryption operation is specified and DK = 0, AES first goes through the steps required to derive the decrypt key from the encrypt key, and then performs the decryption operation. If a decryption operation is specified and DK = 1, the steps required to derive the decrypt key are skipped and the decryption operation is performed immediately, significantly improving performance for small data blocks.

Note that the difference between the encrypt key and the decrypt key must be taken into account when sharing keys between jobs. When an AES decryption job loads a key from memory, it is probably an encrypt key, so the DK bit in the Mode Register should be set to 0 so that AES derives the decrypt key from the encrypt key before beginning the decryption operation. But when a subsequent AES decryption job shares the key from a previous decryption job, the key that is shared is a decrypt key. In that case, the DK bit should be set to 1, which tells AES to skip the key derivation steps. If DK were set to 0 in this case, the decrypt key would be modified as if it were an encrypt key, and consequently, the wrong key value would be used in the decryption operation.

42.3.2 AESA modes of operation

The following modes are supported by AESA:

- Electronic codebook (ECB)
- Cipher block chaining (CBC)
- Counter (CTR)
- XTS tweakable block cipher
- Extended cipher block chaining message authentication code (XCBC-MAC)
- Cipher-based MAC (CMAC)
- CTR and CBC-MAC (CCM)
- Galois/Counter mode (GCM)
- Combined CBC and XCBC (CBCXCBC)
- Combined CTR and XCBC (CTRXCBC)
- Combined CBC and CMAC (CBC-CMAC)
- Combined CTR and CMAC (CTR-CMAC)

AES modes can be classified into these categories:

- Confidentiality (ECB, CBC, CTR, XTS)
- Authenticated Confidentiality (CCM, CCM*, GCM, CBCXCBC, CTRXCBC)
- Authentication (XCBC-MAC, CMAC)

CBC Mode can also be viewed as an authentication mode when used to encrypt data, because it provides CBC-MAC in the context registers.

42.3.3 AESA use of registers

Note the following regarding the AESA's use of registers:

- For all modes, if AES is selected and the mode code written to the Mode Register does not correspond to any of the implemented AES modes, the illegal-mode error is generated.
- If ICV-only(Integrity Check Value, Final MAC) jobs are created (no data to be processed, only ICV to be checked) in modes that support ICV check, the AS mode field should be reset.

42.3.4 AES ECB mode

The electronic codebook (ECB) mode is a confidentiality mode that features, for a given key, the assignment of a fixed, ciphertext block to each plaintext block, analogous to the assignment of code words in a codebook. In ECB encryption, the forward cipher function is applied directly and independently to each block of the plaintext. The resulting sequence of output blocks is the ciphertext. In ECB decryption, the inverse cipher function is applied directly and independently to each block of the ciphertext. The resulting sequence of output blocks is the plaintext.

42.3.4.1 AES ECB mode use of the Mode Register

AES ECB mode uses the Mode Register as follows:

- The Encrypt (ENC) field should be 1 for ECB encryption and 0 for ECB decryption.
- The Algorithm State (AS) field in not used in ECB mode.
- The Additional Algorithm Information (AAI) field must be set with value 20h that activates ECB mode. Setting the MSB in the AAI field (interpreted as the Decrypt Key or DK bit for AES operations) specifies that the key loaded to the Key Register is the decryption form of the key, rather than the encryption form of the key. If DK = 0, when a decryption operation is requested AES processes the content of the Key Register to yield the decryption form of the key. If DK = 1, AES skips this processing. The illegal-mode error is generated if DK = 1 and ENC=1.
- The Algorithm (ALG) field is used to activate AESA by setting it to 10h.

42.3.4.2 AES ECB mode use of the Context Register

ECB does not use Context Registers.

42.3.4.3 AES ECB Mode use of the Data Size Register

The length of the message to be processed in bytes must be written to the Data Size register. If this value is not divisible by 16, the Data Size error is generated.

42.3.4.4 AES ECB Mode use of the Key Register

ECB keys must be written to the Key Register and can have only 16 bytes.

42.3.4.5 AES ECB Mode use of the Key Size Register

The number of bytes in the ECB key must be written to the Key Size register. 16 bytes is the only key size supported for this mode.

42.3.5 AES CBC mode

The CBC mode is described in this table.

Table 42-1. AES CBC, OFB, CFB128 modes

Name	Abbreviation	Function
Cipher-block chaining mode	CBC	Confidentiality mode whose encryption process features the combining ("chaining") of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an IV (Initialization Vector) to combine with the first plaintext block **NOTE:** CBC mode uses both forward and inverse AES cipher. OFB and CFB use only forward AES cipher.

42.3.5.1 AES CBC mode use of the Mode Register

The AES CBC mode use the Mode Register as follows:

- The Encrypt (ENC) field should be 1 for encryption and 0 for decryption
- The ICV/TEST bit is not used in these modes.
- The Algorithm State (AS) field is used only in CBC mode to prevent IV update in the context for the last data block when set to "Finalize" (2h).
- The Additional Algorithm Information (AAI) field must be set with value 10h that activates CBC mode. The Decrypt Key [DK] (AAI field MSB) bit specifies that the

key loaded to the Key Register is the decrypt key. The illegal mode error is generated if DK=1 and ENC=1.
- The Algorithm (ALG) field is used to activate AESA by setting it to 10h.

42.3.5.2 AES CBC mode use of the Context Register

The AES CBC mode use the Context Register as follows:

- CBC use the Context Registers to provide IV, which is updated with every processed block of a message. When a message is split into chunks and processed in multiple sessions, the IV must be saved and later restored for the next chunk to be processed correctly. At the end of CBC processing, IV is also the MAC of the message.
- If the AS field of the Mode Register is set to "Finalize" (2h) in the CBC mode, the last IV update is not written to the context. This enables CBC encryption to effectively perform ECB encryption transformation of a single-block message located in the context in place of IV, and with an all-zero block provided as input data through the FIFO without overwriting the context.

Table 42-2. Context usage in CBC mode

Context Word	Definition
0	IV [127:96]
1	IV [95:64]
2	IV [63:32]
3	IV [31:0]

42.3.5.3 AES CBC mode use of the Data Size Register

The AES CBC mode use the Data Size Register as follows:

- The byte length of the message to be processed must be written to the Data Size Register.
- The first write to this register initiates processing. It can also be written during processing in which case the value written is accumulated to the current state of the register.
- After the Data Size Register is written for the last time, its value must be divisible by 16 in CBC mode, otherwise the data-size error is generated.

42.3.5.4 AES CBC mode use of the Key Register

The AES CBC mode use the Key Register as follows:

- A CBC key must be written to the Key Register.
- Keys must be 16 bytes.

42.3.5.5 AES CBC mode use of the Key Size Register

The AES CBC mode use the Key Size Register as follows:

- The number of bytes in a key must be written to the Key Size register.
- 16 bytes is the only key size supported for this mode.

42.3.6 AES CTR mode

The counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Note that the counter value must be unique for each data block that is encrypted with the same key. uses a 128-bit counter to ensure that the counter value will not overflow and wrap around.

NOTE

It is the user's responsibility to ensure that the same key value is not used again following a reset.

42.3.6.1 AES CTR mode use of the Mode Register

The AES CTR mode uses the Mode Register as follows:

- The Additional Algorithm Information (AAI) field should be set to 00h to activate CTR mode. If the Decrypt Key [DK] (AAI field MSB) bit is set, the illegal-mode error is generated, because CTR uses only forward AES cipher requiring encryption rather than decryption keys.
- The Algorithm State (AS) field when set to "Finalize" (2h) prevents counter update in the context for the last data block.
- The Algorithm (ALG) field is used to activate AESA by setting it to 10h.

42.3.6.2 AES CTR mode use of the Context Register

The AES CTR mode uses the Context Register as follows:

- CTR uses context words 4,5,6 and 7 to provide initial counter value (CTR0). This value is incremented with every processed block of a message. When a message is split into chunks and processed in multiple sessions, the CTRi field of context has to be saved and later restored for the next chunk to be processed correctly.
- If the AS field of the Mode Register is set to Finalize (2h) in the CTR mode, the last counter update is not written to the context. This enables CTR encryption to effectively perform ECB encryption transformation of a single-block message located in the context words 4,5,6 and 7 in place of CTR0 and with all-zero block provided as input data through the FIFO without overwriting the context.

Table 42-3. Context usage in CTR mode

Context Word	Initial-input definition	Context-switching definition
0	-	-
1	-	-
2	-	-
3	-	-
4	CTR0 [127:96]	CTRi [127:96]
5	CTR0 [95:64]	CTRi [95:64]
6	CTR0 [63:32]	CTRi [63:32]
7	CTR0 [31:0]	CTRi [31:0]

42.3.6.3 AES CTR mode use of the Data Size Register

The byte-length of the message to be processed must be written to the Data Size register. CTR decrements the value in this register with every processed block.

42.3.6.4 AES CTR mode use of the Key Register

- CTR key must be written to the Key Register.
- The Key Register only supports 16 byte keys.

42.3.6.5 AES CTR mode use of the Key Size Register

The number of bytes in a key must be written to the Key Size register by the time that MODE and DATA SIZE have been written. 16 bytes is the only key size supported for this mode.

42.3.7 AES XCBC-MAC and CMAC modes

The AES XCBC-MAC and CMAC modes are described together because of their similarities. They are extensions of the AES CBC mode that produces a key-dependent, one--way hash (or message authentication code (MAC)) in a secure fashion across messages of varying lengths. They also provide data-integrity and data-origin authentication regarding the original message source.

42.3.7.1 AES XCBC-MAC and CMAC modes use of the Mode Register

The AES XCBC-MAC and CMAC modes use the Mode Register as follows:

- The Encrypt (ENC) bit is ignored.
- The ICV bit must be set for computed MAC to be compared with the received MAC. The received MAC must be written to the Input Data FIFO after message data and the FIFO data type must be set to ICV. If this bit is not set, XCBC-MAC and CMAC do not expect received ICV to be supplied after message data.
- The Algorithm State (AS) field is defined for XCBC-MAC as shown in this table.

Table 42-4. Mode Register[AS] operation selections in AES XCBC-MAC

Operation	Description
INITIALIZE	Message is processed in multiple sessions and the current session is the first one. During initialization, derived keys K3 and K2 that are XOR-ed with the last message block are computed and stored in the context to be used in the last processing session. The derived key K1 used as an AES key is computed and written back to the Key Register over the original key
INITIALIZE/FINALIZE	Message is processed in a single XCBC session and the final MAC is computed
UPDATE	Message is processed in multiple sessions and the current session is neither the first nor the last. Derived keys K2 and K3 are provided in the context and the derived key K1 is provided in the Key Register. If decryption is requested, and data size is not written or is set to 0, and ICV bit is 1 - AS = UPDATE means that Check ICV (CICV) job is requested. The CICV-only job does not process any data, it just pops received ICV/MAC from the Input Data FIFO, and compares it to the computed MAC that is restored with the rest of the context from the previous session.
FINALIZE	Message is processed in multiple sessions and the current session is the last one. Derived keys K2 and K3 are provided in the context and the derived key K1 is provided in the Key Register. The final MAC is computed

- The Algorithm State (AS) field is defined for CMAC as shown in this table.

Table 42-5. Mode Register[AS] operation selections in CMAC

Operation	Function
INITIALIZE	Message is processed in multiple sessions and the current session is the first one. During initialization, the constant L = E(K, 0) is computed as encrypted block of zeros using key K and stored in the context to be used in the last processing session for derivation of keys K1 and K2. One of these keys will be XOR-ed with the last message block.
INITIALIZE/FINALIZE	Message is processed in a single session and the final MAC is computed
UPDATE	Message is processed in multiple sessions and the current session is neither the first nor the last. The constant L used for key derivation is provided in the context. If decryption is requested, and data size is not written or is set to 0, and ICV bit is 1 - AS = UPDATE means that Check ICV (CICV) job is requested. The CICV-only job does not process any data, it just pops received ICV/MAC from the Input Data FIFO, and compares it to the computed MAC that is restored with the rest of the context from the previous session
FINALIZE	Message is processed in multiple sessions and the current session is the last one. The constant L used for key derivation is provided in the context. The final MAC is computed

- If the AS field is not set to either "Initialize/Finalize" or "Finalize" and the ICV bit is set to 1, the illegal-mode error is generated, except for CICV-only jobs.
- The Additional Algorithm Information (AAI) field must be set to 70h for XCBC and 60h for CMAC to be activated. Setting the DK bit (AAI field MSB) will cause the Illegal Mode error.
- The Algorithm (ALG) field is used to activate AESA by setting it to 10h.

42.3.7.2 AES XCBC-MAC and CMAC Modes use of the Context Register

The AES XCBC-MAC and CMAC modes use the Context Register as follows:

- No data needs to be provided in the context when starting a new XCBC or CMAC session.
- The computed MAC and the derived keys K2 and K3 are written back to the context by XCBC.
- The computed MAC and the constant L = E(K,0), computed as encrypted block of zeros using key K, are written back to the context by CMAC.
- When a message is split into chunks and processed in multiple sessions, these values need to be saved before context switch and restored before the next chunk of a message is to be processed. At the end of message processing the first 2 words of the context contain the MAC value.

Table 42-6. Context usage in XCBC-MAC and CMAC modes

Mode	Context word	Context-switching definition	Final-result definition
XCBC-MAC	0	MAC[127:96]	MAC[127:96]

Table continues on the next page...

Table 42-6. Context usage in XCBC-MAC and CMAC modes (continued)

Mode	Context word	Context-switching definition	Final-result definition
	1	MAC[95:64]	MAC[95:64]
	2	MAC[63:32]	MAC[63:32]
	3	MAC[31:0]	MAC[31:0]
	4	K3[127:96]	-
	5	K3[95:64]	-
	6	K3[63:32]	-
	7	K3[31:0]	-
	8	K2[127:96]	-
	9	K2[95:64]	-
	10	K2[63:32]	-
	11	K2[31:0]	-
CMAC	0	MAC[127:96]	MAC[127:96]
	1	MAC[95:64]	MAC[95:64]
	2	MAC[63:32]	MAC[63:32]
	3	MAC[31:0]	MAC[31:0]
	4	L[127:96]	-
	5	L[95:64]	-
	6	L[63:32]	-
	7	L[31:0]	-

42.3.7.3 AES XCBC-MAC and CMAC modes use of the ICV Size Register

The AES XCBC-MAC and CMAC modes use the ICV Size Register as follows:

- This ICV register is used to provide received ICV/MAC byte-size when it is other than 16 bytes.
- The computed ICV/MAC written to the context in the XCBC mode is always 16 bytes.
- In CMAC mode, this register determines also the computed MAC size-the remaining bytes are cleared.
- Supported values for ICV size are 4 to 16 bytes. If this register is 0, the size of ICV is 16 bytes.

42.3.7.4 AES XCBC-MAC and CMAC modes use of the Data Size Register

The AES XCBC-MAC and CMAC modes use the Data Size Register as follows:

- The byte-length of the message to be processed must be written to the Data Size register.
- XCBC-MAC and CMAC decrement the value in this register with every processed block.

42.3.7.5 AES XCBC-MAC and CMAC modes use of the Key Register

The AES XCBC-MAC and CMAC modes use the Key Register as follows:

- The key must be written to this register.
- For XCBC-MAC, if the AS mode field is set to either "Initialize" or "Initialize/ Finalize", it is the original XCBC key (K) that must be written here. Otherwise, the derived key (K1) must be restored to this register. CMAC only uses original key K as an AES key.

42.3.7.6 AES XCBC-MAC and CMAC modes use of the Key Size Register

The AES XCBC-MAC and CMAC modes use the Key Size Register as follows:

- The total number of key bytes must be written to the Key Size register.
- 16 bytes is the only key size supported for this mode.

42.3.7.7 ICV checking in AES XCBC-MAC and CMAC modes

Automatic ICV checking is enabled by setting the ICV bit of the Mode Register to 1. When ICV is set to 1, the AS mode field must be set to either "Finalize" or "Initialize/ Finalize"; otherwise the illegal-mode error is generated, except for CICV-only (Check-ICV-only) jobs.

The received ICV must be provided on the FIFO after the message data. The FIFO data type must be set to ICV when it is put on the FIFO. The size of the received and computed ICV is provided in the ICV Size register.

If the ICV check detects a mismatch between the decrypted received ICV and the computed ICV, the ICV error is generated.

42.3.8 AESA CCM and CCM* modes

CCM and CCM* consists of two related processes: generation encryption and decryption verification, which combine two cryptographic primitives: counter mode encryption (CTR) and cipher-block chaining based authentication (CBC-MAC). Only the forward cipher function of the block cipher algorithm is used within these primitives. Note that the counter value must be unique for each data block that is encrypted with the same key. AES uses a 128-bit counter to ensure that the counter value does not overflow and wrap around.

NOTE

It is the user's responsibility to ensure that the same key value is not used again following a reset.

42.3.8.1 Generation encryption

A cipher-block chaining is applied to the payload, the associated data (AAD), and the nonce to generate a message authentication code (MAC); then counter mode encryption is applied to the MAC and the payload to transform them into an unreadable form, called the ciphertext. Thus, CCM generation encryption expands the size of the payload by the size of the MAC.

42.3.8.2 Decryption verification

Counter-mode decryption is applied to the purported ciphertext to recover the MAC and the corresponding payload; then cipher block chaining is applied to the payload, the received associated data, and the received nonce to verify the correctness of the MAC.

42.3.8.3 AES CCM and CCM* mode use of the Mode Register

The AES CCM and CCM* mode uses the Mode Register as follows:

- The Encrypt (ENC) bit must be set to 1 for encryption and 0 for decryption.
- The ICV bit must be set for CCM and CCM* to compare computed MAC with the received MAC when decryption is requested.
- The received MAC must be written to the input-data FIFO after message data and the FIFO data type must be set to ICV.
- Setting the ICV bit causes the received MAC to be decrypted and compared with the computed MAC.

- The number of MSBs to be compared is defined by the MAC size in the CCM and CCM* IV (B_0) as described in the CCM specification.
- If the AS field is set to FINALIZE, but ICV = 0, AESA does not expect received ICV to be put on the input-data FIFO. In that case, MAC is computed and truncated to the specified size for decryption.
- For encryption, the computed MAC is encrypted and truncated to size. The illegal-mode error is generated if ICV = 1 and ENC = 1.
- If ICV = 1 and the decrypted received MAC do not match computed MAC, the ICV error is generated.
- The Algorithm State (AS) field is defined for CCM and CCM* as follows:

Table 42-7. Mode Register[AS] operation selections in AES CCM and CCM*

Operation	Description
INITIALIZE	Message is processed in multiple sessions and the current session is the first one. During initialization, the initial counter CTR0 is encrypted in the CTR mode and the B0 is processed with the CBC-MAC mode. The resulting values are stored in the context. Also, the size of MAC is decoded from B0 and written to the context. This AS setting must be used whenever the first part (or whole) AAD is being processed
INITIALIZE/FINALIZE	Message is processed in a single CCM or CCM* session and the final MAC is computed and encrypted. The initial counter CTR0 and B0 must be provided in the context
UPDATE	Message is processed in multiple sessions and the current session is neither the first nor the last. All context data is restored from the previous session and the key is written to the Key Register. If decryption is requested, and data size is not written or is set to 0, and ICV bit is 1 - AS=UPDATE means that a CICV-only job is requested. The CICV-only job does not process any data, it just pops received ICV/MAC from the Input Data FIFO, decrypts it and compares it to the computed MAC that is restored with the rest of the context from the previous session
FINALIZE	Message is processed in multiple sessions and the current session is the last one. All context data is restored from the previous session and the key is written to the Key Register. The final MAC is computed and encrypted

- Whenever AS is set to Initialize or Initialize/Finalize, context registers must be zero.
- If the AS field is not set to either Initialize/Finalize or Finalize and the ICV bit is set to 1, the illegal-mode error is generated. This does not apply in case when only ICV check is requested as described for AS = UPDATE.
- The Additional Algorithm Information (AAI) field must be set to 80h for both CCM and CCM* to be activated. The C2K bit is used to select a key register. If C2K = 0, CCM and CCM* uses the key in the Key Register. Setting the DK bit causes the illegal-mode error.
- The Algorithm (ALG) field is used to activate AESA by setting it to 10h.

42.3.8.4 AES CCM and CCM* modes use of the Context Register

The AES CCM and CCM* mode uses the Context Register as follows:

- B0 and the initial counter CTR0 must be provided in the context before the first chunk of the message is to be processed. During initialization, the initial counter CTR0 is encrypted in the CTR mode and B0 (which functions like a CBC-MAC IV in CCM and CCM*) is processed with the CBC-MAC mode. The resulting values are stored in the context. Also, the size of MAC is decoded from B0 and written to context word 13.
- If there is AAD, the first block of it defines its size, and that value is decoded and written to context word 12. All of the context data must be restored before the next chunk of the message is to be processed in multi-session processing.
- For CCM and CCM* encryption, the ICV (encrypted final MAC) is written to context words 8-11. For CCM and CCM* decryption, the ICV (received MAC), which is always encrypted, is decrypted to words 8-11. The final computed MAC is written (in clear) to context words 0-3.

Table 42-8. Context usage in CCM and CCM* mode encryption

Context Word	Initial-input definition	Intermediate definition	Final-output definition
0	B0[127:96]	-	MAC[127:96]
1	B0[95:64]	-	MAC[95:64]
2	B0[63:32]	-	MAC[63:32]
3	B0[31:0]	-	MAC[31:0]
4	CTR0[127:96]	CTR[127:96]	-
5	CTR0[95:64]	CTR[95:64]	-
6	CTR0[63:32]	CTR[63:32]	-
7	CTR0[31:0]	CTR[31:0]	-
8	-	E(CTR0)[127:96][1]	E(MAC)[127:96]
9	-	E(CTR0)[95:64] [1]	E(MAC)[95:64]
10	-	E(CTR0)[63:32] [1]	E(MAC)[63:32]
11	-	E(CTR0)[31:0] [1]	E(MAC)[31:0]
12	-	AAD size; see Table 42-10	-
13	-	MAC size; see Table 42-11	-

[1] E(x) means encrypted x

Table 42-9. Context usage in CCM and CCM* modes decryption

Context Word	Initial-input definition	Context-switching Definition	Final-result definition
0	B0[127:96]	-	MAC[127:96]
1	B0[95:64]	-	MAC[95:64]
2	B0[63:32]	-	MAC[63:32]
3	B0[31:0]	-	MAC[31:0]
4	CTR0[127:96]	CTR[127:96]	-
5	CTR0[95:64]	CTR[95:64]	-
6	CTR0[63:32]	CTR[63:32]	-

Table continues on the next page...

Table 42-9. Context usage in CCM and CCM* modes decryption (continued)

Context Word	Initial-input definition	Context-switching Definition	Final-result definition
7	CTR0[31:0]	CTR[31:0]	-
8	-	E(CTR0)[127:96]	Decrypted Received MAC[127:96]
9	-	E(CTR0)[95:64]	Decrypted Received MAC[95:64]
10	-	E(CTR0)[63:32]	Decrypted Received MAC[63:32]
11	-	E(CTR0)[31:0] [1]	Decrypted Received MAC[31:0]
12	-	AAD size; see Table 42-10	-
13	-	MAC size; see Table 42-11	-

Table 42-10. Format of Context Word 12 for AES-CCM and AES-CCM* mode

Bit 31	Bits 30-16	Bits 15-0
AAD Presence Flag	0	AAD Size

Table 42-11. Format of Context Word 13 for AES-CCM and AES-CCM* mode

Bits 31-3	Bits 2-0
0	Encoded MAC Size

42.3.8.5 AES CCM and CCM* mode use of the Data Size Register

The AES CCM and CCM* mode uses the Data Size Register as follows:

- The byte-length of the message to be processed must be written to the Data Size register.
- CCM and CCM* decrements the value in this register with every processed block.
- The content of the Data Size register must be divisible by 16 if the AS mode field is set to either "Update" or "Initialize". Otherwise, the data-size error is generated. In other words, message splitting can be done only on a 16-byte boundary.

42.3.8.6 AES CCM and CCM* mode use of the Key Register

CCM and CCM* key must be written to this register; it is always an encryption key.

42.3.8.7 AES CCM and CCM* mode use of the Key Size Register

The AES CCM and CCM* mode uses the Key Size Register as follows:

- The total number of key bytes must be written to the Key Size register.
- 16 bytes is the only key size supported for this mode.

42.3.8.8 AES CCM and CCM* mode use of the ICV check

The AES CCM and CCM* mode uses ICV checking as follows:

- Automatic ICV checking is enabled by setting the ICV bit of the Mode Register to 1. When ICV is set to 1, the AS mode field must be set to either "Finalize" or "Initialize/Finalize"-otherwise the illegal-mode error is generated, unless data size is 0 indicating ICV check is only requested. Also, if ICV = 1, the ENC bit must be 0.
- The received ICV(MAC) must be provided on the input data FIFO after the message data. In CCM and CCM*, received ICV(MAC) is always encrypted. The FIFO data type must be set to ICV when it is put on the FIFO. The size of the received and computed ICV(MAC) is for CCM and CCM* encoded in the B0.
- If the ICV check detects mismatch between the decrypted received ICV(MAC) and the computed ICV(MAC), the ICV error is generated.

42.4 LTC AES Examples

Example AES ECB Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register.
   - Key Size is 16 bytes (00000010h)
3. Write Mode to Primary Mode Register. (0010020Dh)
4. Write Size of data to encrypt/decrypt to Data Size Register.
5. Write data into the Input FIFO.
6. Read data from the Output FIFO.
7. Interrupt is generated after final word is pushed to output FIFO.

Example AES CTR Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register.
   - Key Size is 16 bytes (00000010h)

3. Write initial counter value to Context Words 4-7 in the Context Register.
4. Write Mode to Primary Mode Register. (0010000Dh)
5. Write Size of data to encrypt/decrypt to Data Size Register.
6. Write data into the Input FIFO.
7. Read data from the Output FIFO.
8. Interrupt is generated after final word is pushed to output FIFO.

Example AES CCM or CCM* Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register.
    - Key Size is 16 bytes (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh) CCM and CCM* both use the same mode value.
    - CCM and CCM* both use the same mode value.
6. Write Size of Authentication Only Data to the AAD Size Register.
7. Write Authentication Only data to the Input FIFO.
    - Authentication data needs to be padded to a 16 byte boundary with zeros.
    - For example if there is 8 bytes of AAD then (00000008h) should be written to AAD Size register and 8 bytes of AAD data followed by 8 bytes of Zero should be written into the Input FIFO.
8. Write Size of data to encrypt/decrypt and authenticate to Data Size Register.
9. Write data into the Input FIFO.
10. Read data from the Output FIFO.
11. Interrupt is generated after final word is pushed to output FIFO.
12. Read MAC from Context Registers
    - MAC is read from Context Registers 0-3.
    - Encrypted MAC is read from Context Registers 8-11.

Example AES CCM or CCM* Authentication Only Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register.
    - Key Size is 16 bytes (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh).
    - CCM and CCM* both use the same mode value.
6. Write Size of Authentication Only Data to the AAD Size Register.

- The AL bit needs to be set in the AAD Size Register. This tells the AES core engine that it will recieve only Authentication Data. Note for encryption only the mechanism is handled automatically.

7. Write Authentication Only data to the Input FIFO.
    - Authentication data needs to be padded to a 16 byte boundary with zeros.
    - For example if there is 8 bytes of AAD then (00000008h) should be written to AAD Size register and 8 bytes of AAD data followed by 8 bytes of Zero should be written into the Input FIFO.
8. Write data into the Input FIFO.
9. Interrupt is generated after final word is processed from input FIFO.
10. Read MAC from Context Registers
    - MAC is read from Context Registers 0-3.
    - Encrypted MAC is read from Context Registers 8-11.

Example AES CCM or CCM* Encryption Only Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register.
    - Key Size is 16 bytes (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh).
    - CCM and CCM* both use the same mode value.
6. Write Size of data to encrypt/decrypt to Data Size Register.
7. Write data into the Input FIFO.
8. Read data from the Output FIFO.
9. Interrupt is generated after final word is pushed into the output FIFO.

42.5 Standalone AES Examples

Example AES ECB Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register. (00000010h)
3. Write Mode to Primary Mode Register. (0010020Dh)
4. Write Size of data to encrypt/decrypt to Data Size Register.
5. Write data into the Input FIFO.
6. Read data from the Output FIFO.
7. Interrupt is generated after final word is pushed to output FIFO.

Example AES CTR Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register. (00000010h)
3. Write initial counter value to Context Words 4-7 in the Context Register.
4. Write Mode to Primary Mode Register. (0010020Dh)
5. Write Size of data to encrypt/decrypt to Data Size Register.
6. Write data into the Input FIFO.
7. Read data from the Output FIFO.
8. Interrupt is generated after final word is pushed to output FIFO.

Example AES CCM or CCM* Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register. (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh) CCM and CCM* both use the same mode value.
   - CCM and CCM* both use the same mode value.
6. Write Size of Authentication Only Data to the AAD Size Register.
7. Write Authentication Only data to the Input FIFO.
   - Authentication data needs to be padded to a 16 byte boundary with zeros.
   - For example if there is 8 bytes of AAD then (00000008h) should be written to AAD Size register and 8 bytes of AAD data followed by 8 bytes of Zero should be written into the Input FIFO.
8. Write Size of data to encrypt/decrypt and authenticate to Data Size Register.
9. Write data into the Input FIFO.
10. Read data from the Output FIFO.
11. Interrupt is generated after final word is pushed to output FIFO.
12. Read MAC from Context Registers
    - MAC is read from Context Registers 0-3.
    - Encrypted MAC is read from Context Registers 8-11.

Example AES CCM or CCM* Authentication Only Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register. (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh).
   - CCM and CCM* both use the same mode value.
6. Write Size of Authentication Only Data to the AAD Size Register.

- The AL bit needs to be set in the AAD Size Register. This tells the AES core engine that it will recieve only Authentication Data. Note for encryption only the mechanism is handled automatically.

7. Write Authentication Only data to the Input FIFO.
    - Authentication data needs to be padded to a 16 byte boundary with zeros.
    - For example if there is 8 bytes of AAD then (00000008h) should be written to AAD Size register and 8 bytes of AAD data followed by 8 bytes of Zero should be written into the Input FIFO.
8. Write data into the Input FIFO.
9. Interrupt is generated after final word is processed from input FIFO.
10. Read MAC from Context Registers
    - MAC is read from Context Registers 0-3.
    - Encrypted MAC is read from Context Registers 8-11.

Example AES CCM or CCM* Encryption Only Operation:

1. Write key to Primary Key Register.
2. Write key size to Primary Key Size Register. (00000010h)
3. Write B0 value to Context Words 0-3 in the Context Register.
4. Write initial counter value to Context Words 4-7 in the Context Register.
5. Write Mode to Primary Mode Register. (0010080Dh).
    - CCM and CCM* both use the same mode value.
6. Write Size of data to encrypt/decrypt to Data Size Register.
7. Write data into the Input FIFO.
8. Read data from the Output FIFO.
9. Interrupt is generated after final word is pushed into the output FIFO.

Writing and Reading Data from the FIFOs:

1. Writing and reading by polling operations.
    - The FIFO Status Register(LTCFIFOSTA) shows the number of entries in both the input and output FIFOs. It also shows when the FIFOs are full.
    - The input and output FIFOs support 4x32bit entries each.
    - Whenever there is space in the input FIFO the user can write a word into the Input FIFO.
    - Whenever there is a word in the output FIFO then the user can read a word from the Output FIFO.
2. Writing and reading FIFOs by DMA operations.
    - The on chip DMA will handle all reads and writes of the FIFOs.
    - The IDE and ODE bits in the Control Register must be written to enable the DMA handshake.

- IDE will enable dma transfers to the input FIFO when there is space available.
- ODE will enable dma transfers from the output FIFO when there are words in the FIFO.

- The on chip DMA should then be programmed to write data to the input FIFO and read data from the output FIFO.

42.6 LTC Register Descriptions

All reads of undefined and write-only addresses always return zero. Writes to undefined and read-only addresses are ignored. LTC will never generate a transfer error on the register bus. Although many of the LTC registers hold more than 32 bits, the register addresses shown in the Memory Map below represent how these registers are accessed over the register bus as 32-bit words.

NOTE

The reset value of some registers differs between different versions of LTC. To ensure driver compatibility across different versions of LTC, when updating fields within registers, the registers should first be read, the required fields updated, and then the register should be written. This will avoid inadvertently changing the settings of other fields in the same register.

42.6.1 LTC Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	LTC Mode (LTC0_MD)	32	RW	00000000h
8h	LTC Key Size (LTC0_KS)	32	WO	00000010h
10h	LTC Data Size (LTC0_DS)	32	RW	00000000h
18h	LTC ICV Size (LTC0_ICVS)	32	RW	00000000h
30h	LTC Command (LTC0_COM)	32	WO	00000000h
34h	LTC Control (LTC0_CTL)	32	RW	00000000h
40h	LTC Clear Written (LTC0_CW)	32	WO	00000000h
48h	LTC Status (LTC0_STA)	32	W1C	00000000h
4Ch	LTC Error Status (LTC0_ESTA)	32	RO	00000000h
58h	LTC AAD Size (LTC0_AADSZ)	32	RW	00000000h
100h - 134h	LTC Context (LTC0_CTX_0 - LTC0_CTX_13)	32	RW	00000000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
200h - 20Ch	LTC Keys (LTC0_KEY_0 - LTC0_KEY_3)	32	RW	00000000h
4F4h	LTC Version ID 2 (LTC0_VID2)	32	RO	00000101h
7C0h	LTC FIFO Status (LTC0_FIFOSTA)	32	RO	00000000h
7E0h	LTC Input Data FIFO (LTC0_IFIFO)	32	WO	00000000h
7F0h	LTC Output Data FIFO (LTC0_OFIFO)	32	RO	00000000h
8F0h	LTC Version ID (LTC0_VID1)	32	RO	00340100h
8F8h	LTC CHA Version ID (LTC0_CHAVID)	32	RO	00000050h

42.6.2 LTC Mode (LTC0_MD)

42.6.2.1 Address

Register	Offset
LTC0_MD	0h

42.6.2.2 Function

The Mode Register is used to tell the cryptographic engines which operation is being requested. The interpretation of this register will be unique for each CHA.

This section defines the format of the Mode Register when used with non-public-key algorithms and non-RNG operations.

42.6.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ALG							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			AAI									AS		ICV_ TEST	ENC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.2.4 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-24
—</td><td>Reserved. Must be 0.</td></tr>
<tr><td>23-16
ALG</td><td>Algorithm. This field specifies which algorithm is being selected.
00010000b - AES</td></tr>
<tr><td>15-13
—</td><td>Reserved. Must be 0.</td></tr>
<tr><td>12-4
AAI</td><td>Additional Algorithm information. This field contains additional mode information that is associated with the algorithm that is being executed. See also the section describing the appropriate CHA.
<b>NOTE:</b> Some algorithms do not require additional algorithm information and in those cases this field should be all 0s.
<b>Table 42-12. AAI Interpretation for AES Modes</b>
<table>
<tr><th colspan="5">[For AES the MSB of AAI is the DK (Decrypt Key) bit.]</th></tr>
<tr><th>Code[1]</th><th>Interpretation</th><th></th><th>Code</th><th>Interpretation</th></tr>
<tr><td>00h</td><td>CTR (mod 2^{128})</td><td></td><td>80h</td><td>CCM, CCM*</td></tr>
<tr><td>10h</td><td>CBC</td><td></td><td>90h</td><td>Reserved</td></tr>
<tr><td>20h</td><td>ECB</td><td></td><td>A0h</td><td>Reserved</td></tr>
<tr><td>30h</td><td>Reserved</td><td></td><td>B0h</td><td>Reserved</td></tr>
<tr><td>40h</td><td>Reserved</td><td></td><td>C0h</td><td>Reserved</td></tr>
<tr><td>50h</td><td>Reserved</td><td></td><td>D0h</td><td>Reserved</td></tr>
<tr><td>60h</td><td>CMAC</td><td></td><td>E0h</td><td>Reserved</td></tr>
<tr><td>70h</td><td>XCBC-MAC</td><td></td><td></td><td></td></tr>
<tr><td colspan="5">Setting the DK bit (i.e. ORing 100h with any AES code above) causes Key Register to be loaded with the AES Dcrypt key, rather than the AES Encrypt key.</td></tr>
</table>
[1] The codes are mutually exclusive (i.e. they cannot be ORed with each other).</td></tr>
<tr><td>3-2
AS</td><td>Algorithm State. This field defines the state of the algorithm that is being executed. This may not be used by every algorithm.
00b - Update
01b - Initialize
10b - Finalize
11b - Initialize/Finalize</td></tr>
<tr><td>1
ICV_TEST</td><td>ICV Checking / Test AES fault detection.
For algorithms other than AES ECB mode: ICV Checking
This bit selects whether the current algorithm should compare the known ICV versus the calculated ICV. This bit will be ignored by algorithms that do not support ICV checking.</td></tr>
</table>

Table continues on the next page...

Field	Function
	0 - Don't compare 1 - Compare For AES ECB mode: Test AES fault detection In AES ECB mode, this bit activates fault detection testing by injecting bit level errors into AES core logic as defined in the first 128 bits of the context. 0 - Don't inject bit errors 1 - Inject bit errors
0 ENC	Encrypt/Decrypt. This bit selects encryption or decryption. 0b - Decrypt. 1b - Encrypt.

1. The codes are mutually exclusive (i.e. they cannot be ORed with each other).

42.6.3 LTC Key Size (LTC0_KS)

42.6.3.1 Address

Register	Offset
LTC0_KS	8h

42.6.3.2 Function

The Key Size Register is used to tell the crypto engine(AES) the size of the key that was loaded into the Key Register. The Key Size Register must be written after the key is written into the Key Register. Writing to the Key Size Register will prevent the user from modifying the Key Register. Only 16 byte keys are supported so this register will always read 16 bytes. This register is still required to be written to indicate to the AES engine that the key was loaded.

42.6.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W												KS				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

42.6.3.4 Fields

Field	Function
31-5 —	Reserved.
4-0 KS	Key Size. This is the size of a Key measured in bytes

42.6.4 LTC Data Size (LTC0_DS)

42.6.4.1 Address

Register	Offset
LTC0_DS	10h

42.6.4.2 Function

The Data Size Register is used to tell the AES the amount of data that will be loaded into the Input Data FIFO. This register should only be written to once during a single operation. Note that writing to the LTC AAD Size (LTC0_AADSZ), will cause this register to also update. When this register is then written directory to then the new value will be added to the previous value in the register. That is, if the DS field currently has the value 16, writing 2 to the least-significant half of the Data Size register (i.e. the DS field) will result in a value of 18 in the DS field. Note that AES decrements this register, so reading the register may return a value less than sum of the values that were written into it. This register is cleared whenever a key is decrypted or encrypted.

42.6.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DS											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.4.4 Fields

Field	Function
31-12 —	Reserved.
11-0 DS	Data Size. This is the number of whole bytes of data that will be consumed by the CHA. Note that writing the AAD Size Register will result in this register also being written to.

42.6.5 LTC ICV Size (LTC0_ICVS)

42.6.5.1 Address

Register	Offset
LTC0_ICVS	18h

42.6.5.2 Function

The ICV Size Register indicates how much of the last block of ICV is valid when performing AES integrity check modes (e.g. AES-CMAC, AES-XCBC-MAC). This register must be written prior to the corresponding word of data being consumed by AES. In practical terms, this means the register must be written prior to the corresponding data being written to the Input Data FIFO.

42.6.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											ICVS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.5.4 Fields

Field	Function
31-5 —	Reserved.
4-0 ICVS	ICV Size, in Bytes.

42.6.6 LTC Command (LTC0_COM)

42.6.6.1 Address

Register	Offset
LTC0_COM	30h

42.6.6.2 Function

The LTC Command Register is used to send control signals to the Crypto Engines.

42.6.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			Reserved	Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			Reserved		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved		
W															AES	ALL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.6.4 Fields

Field	Function
31-29 —	Reserved. To preserve software compatibility with other versions of LTC, 0 should be written to all reserved bits.
28 —	Reserved.
27-13 —	Reserved.
12-11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved

Table continues on the next page...

Field	Function
5 —	Reserved
4 —	Reserved
3 —	Reserved.
2 —	Reserved
1 AES	Reset AESA. Writing a 1 to this bit resets the AES Accelerator core engine. 0b - Do Not Reset 1b - Reset AES Accelerator
0 ALL	Reset All Internal Logic. Writing to this bit will reset all accelerator engines and as well as all the internal registers. 0b - Do Not Reset 1b - Reset all CHAs in use by this CCB.

42.6.7 LTC Control (LTC0_CTL)

42.6.7.1 Address

Register	Offset
LTC0_CTL	34h

42.6.7.2 Function

This register is used for some of the internal controls of the LTC block.

42.6.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KAL	Reserved							COS	CIS	KOS	KIS	Reserved		OFS	IFS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		OFR	OFE	Reserved		IFR	IFE	Reserved			Reserved	Reserved			IM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.7.4 Fields

Field	Function
31 KAL	Key Register Access Lock. Read access to the key register is blocked. Any reads of the key register will only return zero. Once this bit is set, it can only be cleared by hard reset. 0b - Key Register is readable. 1b - Key Register is not readable.
30-24 —	Reserved.
23 COS	Context Register Output Byte Swap. Byte swap all data that is read from the context register. Data is byte swapped only within a single word. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.
22 CIS	Context Register Input Byte Swap. Byte swap all data that is written to the context register. Data is byte swapped only within a single word. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.
21 KOS	Key Register Output Byte Swap. Byte swap all data that is read from the key register. Data is byte swapped only within a single word. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.
20 KIS	Key Register Input Byte Swap. Byte swap all data that is written to the key register. Data is byte swapped only within a single word. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.
19-18 —	Reserved.
17 OFS	Output FIFO Byte Swap. Byte swap all data that is read from the Onput FIFO. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.

Table continues on the next page...

Field	Function
16 IFS	Input FIFO Byte Swap. Byte swap all data that is written to the Input FIFO. 0b - Do Not Byte Swap Data. 1b - Byte Swap Data.
15-14 —	Reserved.
13 OFR	Output FIFO DMA Request Size. The DMA request logic will only request data if the OUTPUT FIFO has enough data to satisfy the request. 0b - DMA request size is 1 entry. 1b - DMA request size is 4 entries.
12 OFE	Output FIFO DMA Enable. 0b - DMA Request and Done signals disabled for the Output FIFO. 1b - DMA Request and Done signals enabled for the Output FIFO.
11-10 —	Reserved.
9 IFR	Input FIFO DMA Request Size. The DMA request logic will only request data if the INPUT FIFO has enough space for the request size. 0b - DMA request size is 1 entry. 1b - DMA request size is 4 entries.
8 IFE	Input FIFO DMA Enable. 0b - DMA Request and Done signals disabled for the Input FIFO. 1b - DMA Request and Done signals enabled for the Input FIFO.
7-5 —	Reserved.
4 —	Reserved.
3-1 —	Reserved.
0 IM	Interrupt Mask. Once this bit is set, it can only be cleared by hard reset. 0b - Interrupt not masked. 1b - Interrupt masked

42.6.8 LTC Clear Written (LTC0_CW)

42.6.8.1 Address

Register	Offset
LTC0_CW	40h

42.6.8.2 Function

The Clear Written Register is used to clear many of the internal registers. All fields of this register are self-clearing.

42.6.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R			Reserved		Reserved											
W	CIF	COF														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											Reserved			Reserved	
W										CKR	CCR		CICV	CDS		CM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.8.4 Fields

Field	Function
31 CIF	Clear Input FIFO. Writing a 1 to this bit causes the Input Data FIFO.
30 COF	Clear Output FIFO. Writing a 1 to this bit causes the Output FIFO to be cleared.
29-28 —	Reserved.
27-16 —	Reserved.
15-7 —	Reserved.
6 CKR	Clear the Key Register. Writing a one to this bit causes the Key and Key Size Registers to be cleared.
5 CCR	Clear the Context Register. Writing a one to this bit causes the Context Register to be cleared.
4 —	Reserved.
3 CICV	Clear the ICV Size Register. Writing a one to this bit causes the ICV Size Register to be cleared.

Table continues on the next page...

Field	Function
2 CDS	Clear the Data Size Register. Writing a one to this bit causes the Data Size Register to be cleared. This clears AAD Size as well.
1 —	Reserved.
0 CM	Clear the Mode Register. Writing a one to this bit causes the Mode Register to be cleared.

42.6.9 LTC Status (LTC0_STA)

42.6.9.1 Address

Register	Offset
LTC0_STA	48h

42.6.9.2 Function

The LTC Status Register shows the status of the internal Crypto engine and its internal registers.

42.6.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		Reserved			Reserved		Reserved		Reserved		EI		Reserved		DI
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		Reserved		Reserved		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AB	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.9.4 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31
—</td><td>Reserved.</td></tr>
<tr><td>30-28
—</td><td>Reserved.</td></tr>
<tr><td>27-25
—</td><td>Reserved.</td></tr>
<tr><td>24
—</td><td>Reserved.</td></tr>
<tr><td>23-21
—</td><td>Reserved.</td></tr>
<tr><td>20
EI</td><td>Error Interrupt. The Error Interrupt has been asserted. This error can only be cleared by resetting LTC.
0b - Not Error.
1b - Error Interrupt.</td></tr>
<tr><td>19-17
—</td><td>Reserved.</td></tr>
<tr><td>16
DI</td><td>Done Interrupt. The Done Interrupt has been asserted.
<table>
<tr><th>Valu
e</th><th>Read</th><th>Write</th></tr>
<tr><td>0</td><td>No Done Interrupt</td><td>No change</td></tr>
<tr><td>1</td><td>Done Interrupt asserted</td><td>Clear the Done Interrupt</td></tr>
</table></td></tr>
<tr><td>15-13
—</td><td>Reserved.</td></tr>
<tr><td>12-11
—</td><td>Reserved</td></tr>
<tr><td>10
—</td><td>Reserved</td></tr>
<tr><td>9
—</td><td>Reserved</td></tr>
<tr><td>8
—</td><td>Reserved</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
—</td><td>Reserved</td></tr>
<tr><td>5</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Field	Function
—	
4 —	Reserved
3 —	Reserved.
2 —	Reserved
1 AB	AESA Busy. This bit indicates that the AES Accelertor is busy. The CHA can either be busy processing data or resetting. 0b - AESA Idle 1b - AESA Busy.
0 —	Reserved.

42.6.10 LTC Error Status (LTC0_ESTA)

42.6.10.1 Address

Register	Offset
LTC0_ESTA	4Ch

42.6.10.2 Function

The LTC Error Register shows the status of the internal Crypto Engine and its internal registers.

42.6.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				CL1				Reserved				ERRID1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.10.4 Fields

Field	Function
31-12 —	Reserved.
11-8 CL1	algorithms. The algorithms field indicates which algorithm is asserting an error. Others reserved 0000b - LTC General Error 0001b - AES
7-4 —	Reserved
3-0 ERRID1	Error ID 1. These bits indicate the type of error that was found while processing the Descriptor. The Algorithm that is associated with the error can be found in the CL1 field. Others reserved. 0001b - Mode Error 0010b - Data Size Error 0011b - Key Size Error 0110b - Data Arrived out of Sequence Error 1010b - ICV Check Failed 1011b - Internal Hardware Failure 1100b - CCM AAD Size Error (either 1. AAD flag in B0 =1 and no AAD type provided, 2. AAD flag in B0 = 0 and AAD povided, or 3. AAD flag in B0 =1 and not enough AAD provided - expecting more based on AAD size.) 1111b - Invalid Crypto Engine Selected

42.6.11 LTC AAD Size (LTC0_AADSZ)

42.6.11.1 Address

Register	Offset
LTC0_AADSZ	58h

42.6.11.2 Function

The AAD Size Register is used by AESA to determine how much of the last block of AAD is valid. The write to this register should be the entire size of the AAD as it is also added directly to the Data Size Register. The size added to the Data Size Register is the AAD size rounded up to the next 16 byte boundary. For instance a size of 20 bytes written to the AAD size register will cause 32 bytes to be added to the Data Size Register. The size stored in the AADSZ field represents the number of bytes valid in the final block of AAD. However the entire size of AAD should be written to the LTC AAD Size (LTC0_AADSZ) Register address location. When authentication only is being done then the AL bit needs to be written to tell the AES engine that this is the last of the data.

42.6.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												AADSZ			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.11.4 Fields

Field	Function
31 AL	AAD Last. Only AAD data will be written into the Input FIFO.
30-4 —	Reserved.

Table continues on the next page...

Field	Function
3-0 AADSZ	AAD size in Bytes, mod 16.

42.6.12 LTC Context (LTC0_CTX_a)

42.6.12.1 Address

For a = 0 to 13:

Register	Offset
LTC0_CTX_a	100h + (a × 4h)

42.6.12.2 Function

The Context Register holds the context for the internal crypto engine. This register is 448 bits in length. The IP bus write to the Context Register is accessible only as full-word reads or writes to fourteen 32-bit registers. The MSB is located at offset 0100h with respect to the register page.

The bit assignments of this register are dependent on the algorithm, and in some cases the mode of that algorithm. See the appropriate section for the Context Register format used for that algorithm:

- AES ECB: Section AES ECB mode use of the Context Register
- AES CBC: Section AES CBC mode use of the Context Register
- AES CTR: Section AES CTR mode use of the Context Register
- AES CCM: Section AES CCM mode use of the Context Register

42.6.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CTX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.12.4 Fields

Field	Function
31-0 CTX	CTX

42.6.13 LTC Keys (LTC0_KEY_a)

42.6.13.1 Address

Register	Offset
LTC0_KEY_0	200h
LTC0_KEY_1	204h
LTC0_KEY_2	208h
LTC0_KEY_3	20Ch

42.6.13.2 Function

The Key Register normally holds the left-aligned key for the internal crypto engine. The MSB is in offset 200h. The Key Register is 128 bits in length. The IP bus write to the Context Register is accessible only as full-word reads or writes to four 32-bit registers.

Before the value in the Key Register can be used in a cryptographic operation, the size of the key must be written into the Key Size Register. Once the Key Size Register has been written, the Key Register cannot be written again until the Key Size Register has been cleared.

42.6.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KEY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.13.4 Fields

Field	Function
31-0 KEY	KEY

42.6.14 LTC Version ID 2 (LTC0_VID2)

42.6.14.1 Address

Register	Offset
LTC0_VID2	4F4h

42.6.14.2 Function

This register contains the architectural era and eco revision numbers.

42.6.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARCH_ERA								ECO_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

42.6.14.4 Fields

Field	Function
31-16 —	Reserved
15-8 ARCH_ERA	Architectural ERA.
7-0 ECO_REV	ECO revision number.

42.6.15 LTC FIFO Status (LTC0_FIFOSTA)

42.6.15.1 Address

Register	Offset
LTC0_FIFOSTA	7C0h

42.6.15.2 Function

The LTC FIFO Status shows the current levels of the Input and Output FIFO.

42.6.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OFF	Reserved								OFL						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IFF	Reserved								IFL						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.15.4 Fields

Field	Function
31 OFF	Output FIFO Full. The Output FIFO is full and should not be written to.
30-23 —	Reserved
22-16 OFL	Output FIFO Level. These bits indicate the current number of entries in the Output FIFO.
15 IFF	Input FIFO Full. The Input FIFO is full and should not be written to.
14-7 —	Reserved
6-0 IFL	Input FIFO Level. These bits indicate the current number of entries in the Input FIFO.

42.6.16 LTC Input Data FIFO (LTC0_IFIFO)

42.6.16.1 Address

Register	Offset
LTC0_IFIFO	7E0h

42.6.16.2 Function

Data to be processed by the various crypto engines is first pushed into the Input Data FIFO. The Input Data FIFO supports byte enables, allowing one to four bytes to be written to the IFIFO from the IP bus. The IFIFO is four entries deep, and each entry is four bytes. Care must be used to not overflow the Input Data FIFO. Reads from this address will always return 0x0.

42.6.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	IFIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	IFIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.16.4 Fields

Field	Function
31-0 IFIFO	IFIFO

42.6.17 LTC Output Data FIFO (LTC0_OFIFO)

42.6.17.1 Address

Register	Offset
LTC0_OFIFO	7F0h

42.6.17.2 Function

Data that is output from the AES is pushed into the Output Data FIFO. The OFIFO is four entries deep, and each entry is four bytes. During normal operation, the AES will never overflow the Output Data FIFO. Writes to this register are ignored.

42.6.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OFIFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OFIFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.6.17.4 Fields

Field	Function
31-0 OFIFO	Output FIFO

42.6.18 LTC Version ID (LTC0_VID1)

42.6.18.1 Address

Register	Offset
LTC0_VID1	8F0h

42.6.18.2 Function

This register contains the ID for LTC and major and minor revision numbers.

42.6.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJ_REV								MIN_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

42.6.18.4 Fields

Field	Function
31-16 IP_ID	ID(0x0034).
15-8 MAJ_REV	Major revision number.
7-0 MIN_REV	Minor revision number.

42.6.19 LTC CHA Version ID (LTC0_CHAVID)

42.6.19.1 Address

Register	Offset
LTC0_CHAVID	8F8h

42.6.19.2 Function

This register contains the Version ID and Revision Number for the CHAs contained within LTC.

42.6.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AESVID				AESREV			
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

42.6.19.4 Fields

Field	Function
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-4 AESVID	AES Version ID
3-0 AESREV	AES Revision Number

Chapter 43
True Random Number Generator (TRNG)

43.1 Standalone True Random Number Generator (SA-TRNG).

The Standalone True Random Number Generator (SA-TRNG) is hardware accelerator module that generates a 512-bit entropy as needed by an entropy consuming module or by other post processing functions. A typical entropy consumer is a pseudo random-number generator (PRNG) which can be implemented to achieve both true randomness and cryptographic strength random numbers using the TRNG output as its entropy seed. The PRNG is not part of this module.

The entropy generated by an TRNG is intended for direct use by functions that generate secret keys, per-message secrets, random challenges, and other similar quantities used in cryptographic algorithms. In each of these cases, it is important that a random number be difficult to guess or predict. It is important that a random number is at least as difficult to predict as it is difficult to break the cryptographic algorithm with which it is being used. This stringent requirement is particularly difficult to fulfill if the entropy source from a TRNG contains bias and/or correlation. To increase the trustworthiness/quality of the generated random data, PRNGs are often used to post process the output of a TRNG.

This document describes only the TRNG design functionality and usage.

Note that before entropy can be obtained from the TRNG, it must be initialized and instantiated in a particular mode by setting the appropriate TRNG registers.

The TRNG contains the following sub modules: IP Slave bus (SkyBlue bus) interface, the TRNG Core and the free running oscillator (OSC).

43.1.1 Standalone True Random Number Generator Block Diagram

The following figure is a top-level diagram of the True Random Number Generator.

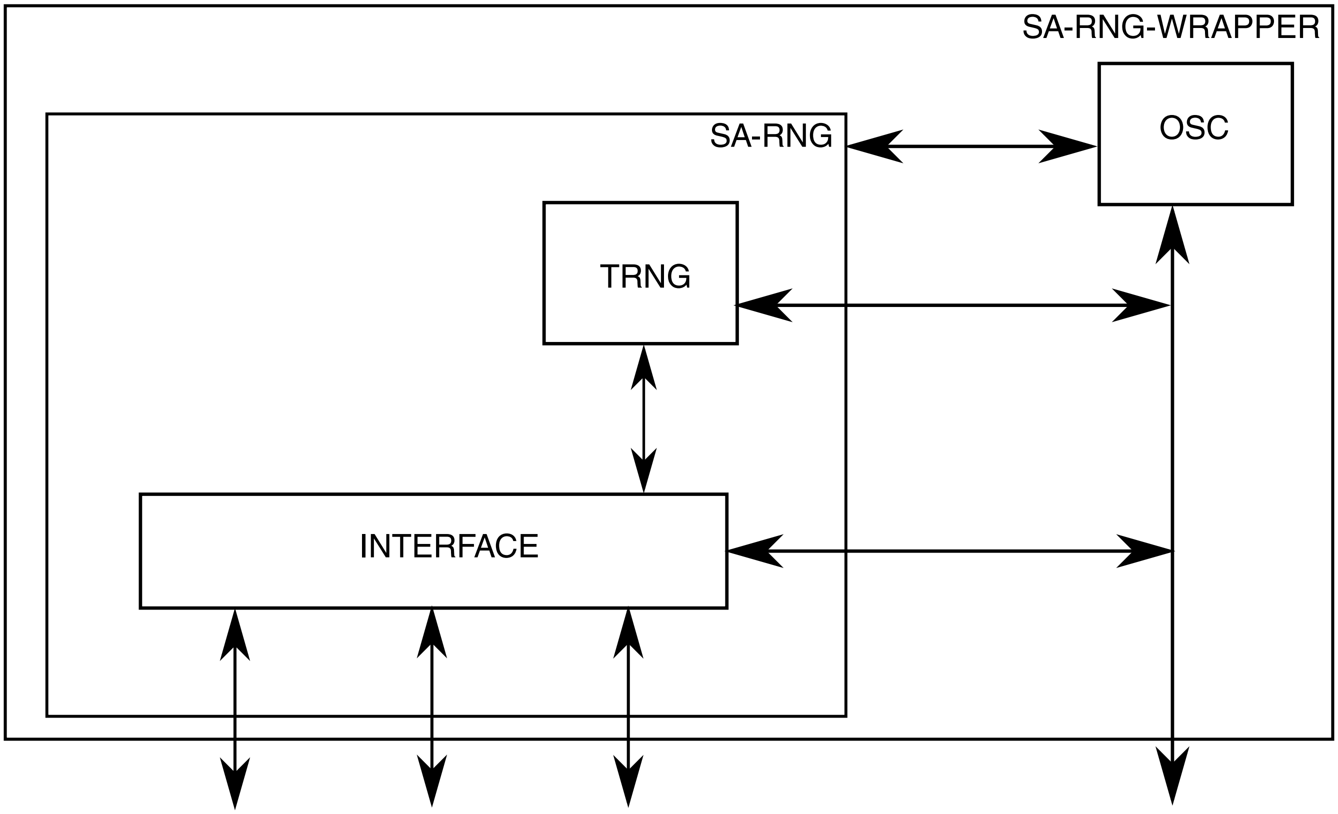


Figure 43-1. SA TRNG Block Diagram

43.1.2 TRNG Functional Description.

The TRNG consists of several functional sub-modules. Its overall functionality can be easily described from the top level in terms of generating entropy for seed generation. The functionality of each sub-module is briefly described in the following subsections.

TRNG is based on collecting bits from a random noise source. This random noise source is a ring oscillator that is sensitive to random noise (temperature variations, voltage variations, cross-talk and other random noise) within the device in which the TRNG is used. This noise causes various small changes in the period of the oscillator. Therefore, if the count of the ring oscillator clock cycles is sampled after a known period of time, this count will vary each time the sample is taken. By using the variance in this count over a large number of samples, random bits can be derived.

The TRNG comprises two entropy sources, each of which provides a single bit of output. Concatenated together, these 2 output bits are expected to provide 1 bit of entropy every 100 clock cycles. In addition to generating entropy, the TRNG also performs several statistical tests on its output.

43.1.3 SA-TRNG hardware functional description.

SA-TRNG functionality consists of several major subcomponents. This table describes these subcomponents.

Table 43-1. SA-TRNG subcomponents.

Description	Cross-reference(s)
Interfaces	
Register interface • Used for access to configuration, control, status and debugging registers	Register interface (IP Slave bus)
True Random Number Generator (TRNG)	Standalone True Random Number Generator (SA-TRNG).

43.1.3.1 Software Use Cases for the Stand Alone TRNG.

There are four things that a user (programmer/integrator) will want to do with a TRNG.

- Initialization.

  Set up the parameters to proper values, and start generation of the first block of entropy. This is done once.

- Read entropy from the TRNG, and start generation of the next block of entropy.

  This is done many times and is the normal flow of operation.

- Run a self-test on the TRNG, to assure proper continued operation.

  This involves taking TRNG off-line, setting some self-test parameters, running TRNG, and then reading the statistical test registers, to see that they are within proper operation values. This may not be needed, as TRNG has built-in self-test.

- Off-line determination and checking of TRNG parameter values.

  This is done in development in order to determine the proper initialization and self-test parameters. The TRNG is taken off-line. Test parameter values are written and entropy generation is started. If the statistical tests indicate poor operation (i.e., failing statistical tests), the entropy_delay value should be increased and entropy generation should be re-started. Every case is a variation of setting TRNG parameter values, starting or re-starting entropy generation and reading out the entropy. This process requires pausing or stopping and re-starting the TRNG.

The TRNG is designed to operate as a slave module on the standard IP Slave Bus. By understanding the TRNG register descriptions in "TRNG Register Descriptions" section below, the TRNG module can be controlled via the IP slave bus. In order to write to most TRNG registers, the MCTL register must be initialized in programming mode as described in the "TRNG Register Descriptions" section. At Power On Reset (POR), the TRNG resets to programming mode. And the it will not generate entropy until it is out of programming mode (in run mode) and access to Entropy Registers have been enabled.

Here is an example program flow of using the TRNG.

- After POR the TRNG will be reset into programming mode with the OK to stop bit set(MCTL[TSTOP_OK]=1). The TRNG must be put into Run Mode for Entropy Generation to begin (MCTL[PRGM]=0). Additionally, in order to have access to the Entropy registers and other critical TRNG registers, the TRNG access bit must be set (MCTL[TRNG_ACC]=1). Using the default self test limits that exist after bootup, the entropy valid bit can be polled until asserted (MCTL[ENT_VAL]=1). Alternatively, if using the interrupt, and the interrupts are enabled via the INT_MASK register and the ipi_rng_int_b is asserted when MCTL[ENT_VAL]=1.
- After the polling completes, the 512-bit entropy generated by the TRNG can be read. The values can be read in any order from entropy register 0 to register 15 (ENT0 to ENT 15). After reading ENT 15, the old entropy value is reset and a new entropy value is generated.

NOTE

Reading ENT 15always resets the entropy, so should always be read last.

- You can poll again for the new entropy value or you can use the Interrupt Status Register to handle reading the entropy values when the entropy valid interrupt is triggered.
- The interrupt can be masked or cleared as needed. See the Interrupt Status Register description.
- To change the self-test limits,the seed counters, how fast the entropy is generated, and how entropy is sampled, see the register description section. In particular, see the the TRNG Frequency Count Minimum Limit Register (FRQMIN), the seed control register(SCML), the statistical run length registers, and other parameter registers.
- Once in Run Mode, the entropy is re-generated automatically after ENT 15 is read. To stop the TRNG or access to TRNG registers at any point while in running mode, you can always set MCTL[TRNG_ACC]=0. Setting the TRNG back to programming mode (MCTL[PRGM]=1) also achieves the purpose of stopping entropy generation.

43.1.3.2 Register interface (IP Slave bus)

The TRNG's register interface (32-bit IP bus) is used to read and write registers within TRNG for the following purposes:

Table 43-2. Summary of register interface uses

Purpose	For more information, see
During chip initialization time	
To configure TRNG including initialization of the • Registers • TRNG Register Interface	
During hardware and software debugging	
Read status registers	• RNG TRNG Status Register • For all registers, see the TRNG Register Descriptions" appendix.

NOTE

Accesses to registers must use full-word (32-bit) reads or writes.

43.1.3.3 TRNG0 Register Descriptions

All accesses of undefined addresses always return zero and assert IPS transfer error. Writes to undefined and read-only addresses are ignored. Undefined addresses are those undocumented, protected or reserved addresses within and outside the range of the addresses defined in the memory map below. Although many of the TRNG0 registers hold more than 32 bits, the register addresses shown in the Memory Map below represent how these registers are accessed over the register bus as 32-bit words.

The format and fields in each TRNG0 register are defined below. Some of the register format figures apply to several different registers. In such cases a different register name will be associated with each of the register offset addresses that appear at the top of the register format figure. Although these registers share the same format, they are independent registers. In addition, many registers can be accessed at multiple addresses. In these cases there will be a single register name and the list of addresses at which that register is accessible will be indicated as aliases. Unless noted in the individual register descriptions, registers are reset only at Power-On Reset (POR).

43.1.3.3.1 TRNG0 Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	TRNG0 Miscellaneous Control (TRNG0_MCTL)	32	RW	00012001h
4h	TRNG0 Statistical Check Miscellaneous (TRNG0_SCMISC)	32	RW	0001001Fh[1]
8h	TRNG0 Poker Range (TRNG0_PKRRNG)	32	RW	000009A3h
Ch	TRNG0 Poker Maximum Limit (TRNG0_PKRMAX)	32	RW	00006920h
Ch	TRNG0 Poker Square Calculation Result (TRNG0_PKRSQ)	32	RO	00000000h
10h	TRNG0 Seed Control (TRNG0_SDCTL)	32	RW	0C8009C4h
14h	TRNG0 Sparse Bit Limit (TRNG0_SBLIM)	32	RW	0000003Fh
14h	TRNG0 Total Samples (TRNG0_TOTSAM)	32	RO	00000000h
18h	TRNG0 Frequency Count Minimum Limit (TRNG0_FRQMIN)	32	RW	00000640h
1Ch	TRNG0 Frequency Count Maximum Limit (TRNG0_FRQMAX)	32	RW	00006400h
1Ch	TRNG0 Frequency Count (TRNG0_FRQCNT)	32	RO	00000000h
20h	TRNG0 Statistical Check Monobit Count (TRNG0_SCMC)	32	RO	00000000h
20h	TRNG0 Statistical Check Monobit Limit (TRNG0_SCML)	32	RW	010C0568h
24h	TRNG0 Statistical Check Run Length 1 Limit (TRNG0_SCR1L)	32	RW	00B20195h
24h	TRNG0 Statistical Check Run Length 1 Count (TRNG0_SCR1C)	32	RO	00000000h
28h	TRNG0 Statistical Check Run Length 2 Limit (TRNG0_SCR2L)	32	RW	007A00DCh
28h	TRNG0 Statistical Check Run Length 2 Count (TRNG0_SCR2C)	32	RO	00000000h
2Ch	TRNG0 Statistical Check Run Length 3 Count (TRNG0_SCR3C)	32	RO	00000000h
2Ch	TRNG0 Statistical Check Run Length 3 Limit (TRNG0_SCR3L)	32	RW	0058007Dh
30h	TRNG0 Statistical Check Run Length 4 Count (TRNG0_SCR4C)	32	RO	00000000h
30h	TRNG0 Statistical Check Run Length 4 Limit (TRNG0_SCR4L)	32	RW	0040004Bh
34h	TRNG0 Statistical Check Run Length 5 Limit (TRNG0_SCR5L)	32	RW	002E002Fh
34h	TRNG0 Statistical Check Run Length 5 Count (TRNG0_SCR5C)	32	RO	00000000h
38h	TRNG0 Statistical Check Run Length 6+ Limit (TRNG0_SCR6PL)	32	RW	002E002Fh
38h	TRNG0 Statistical Check Run Length 6+ Count (TRNG0_SCR6PC)	32	RO	00000000h
3Ch	TRNG0 Status (TRNG0_STATUS)	32	RO	00000000h
40h - 7Ch	TRNG0 Entropy Read (TRNG0_ENT0 - TRNG0_ENT15)	32	RO	00000000h
80h	TRNG0 Statistical Check Poker Count 1 and 0 (TRNG0_PKRCNT10)	32	RO	00000000h
84h	TRNG0 Statistical Check Poker Count 3 and 2 (TRNG0_PKRCNT32)	32	RO	00000000h
88h	TRNG0 Statistical Check Poker Count 5 and 4 (TRNG0_PKRCNT54)	32	RO	00000000h
8Ch	TRNG0 Statistical Check Poker Count 7 and 6 (TRNG0_PKRCNT76)	32	RO	00000000h
90h	TRNG0 Statistical Check Poker Count 9 and 8 (TRNG0_PKRCNT98)	32	RO	00000000h
94h	TRNG0 Statistical Check Poker Count B and A (TRNG0_PKRCNT BA)	32	RO	00000000h
98h	TRNG0 Statistical Check Poker Count D and C (TRNG0_PKRCNT DC)	32	RO	00000000h
9Ch	TRNG0 Statistical Check Poker Count F and E (TRNG0_PKRCNT FE)	32	RO	00000000h
B0h	TRNG0 Security Configuration (TRNG0_SEC_CFG)	32	RW	00000000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B4h	TRNG0 Interrupt Control (TRNG0_INT_CTRL)	32	RW	FFFFFFFFh
B8h	TRNG0 Mask (TRNG0_INT_MASK)	32	RW	00000000h
BCh	TRNG0 Interrupt Status (TRNG0_INT_STATUS)	32	RW	00000000h
F0h	TRNG0 Version ID (MS) (TRNG0_VID1)	32	RO	00300100h
F4h	TRNG0 Version ID (LS) (TRNG0_VID2)	32	RO	00000000h

1. Reset occurs at POR, and when TRNG0_MCTL[RST_DEF] is written to 1.

43.1.3.3.2 TRNG0 Miscellaneous Control (TRNG0_MCTL)

43.1.3.3.2.1 Address

Register	Offset
TRNG0_MCTL	0h

43.1.3.3.2.2 Function

This register is intended to be used for programming, configuring and testing the RNG. It is the main register to read/write, in order to enable Entropy generation, to stop entropy generation and to block access to entropy registers. This is done via the special TRNG_ACC and PRGM bits below.

The TRNG0 Miscellaneous Control Register is a read/write register used to control the RNG's True Random Number Generator (TRNG) access, operation and test.

NOTE

Note that in many cases two RNG registers share the same address, and a particular register at the shared address is selected based upon the value in the PRGM field of the TRNG0_MCTL register.

43.1.3.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															PRGM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		TSTOP_OK	ERR	TST_OUT	ENT_VAL	FCT_VAL	FCT_FAIL	FOR_SCLK		TRNG_ACC	UNUSED	OSC_DIV		SAMP_MODE	
W				w1c						RST_DEF						
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

43.1.3.3.2.4 Fields

Field	Function
31-17 —	Reserved.
16 PRGM	Programming Mode Select. When this bit is 1, the TRNG is in Program Mode, otherwise it is in Run Mode. No Entropy value will be generated while the TRNG is in Program Mode. Note that different RNG registers are accessible at the same address depending on whether PRGM is set to 1 or 0. This is noted in the RNG register descriptions.
15-14 —	Reserved.
13 TSTOP_OK	TRNG_OK_TO_STOP. Software should check that this bit is a 1 before transitioning TRNG0 to low power mode (TRNG0 clock stopped). TRNG0 turns on the TRNG free-running ring oscillator whenever new entropy is being generated and turns off the ring oscillator when entropy generation is complete. If the TRNG0 clock is stopped while the TRNG ring oscillator is running, the oscillator will continue running even though the TRNG0 clock is stopped. TSTOP_OK is asserted when the TRNG ring oscillator is not running. and therefore it is ok to stop the TRNG0 clock.
12 ERR	Read: Error status. 1 = error detected. 0 = no error. Write: Write 1 to clear errors. Writing 0 has no effect.
11 TST_OUT	Read only: Test point inside ring oscillator.
10 ENT_VAL	Read only: Entropy Valid. Will assert only if TRNG ACC bit is set, and then after an entropy value is generated. Will be cleared at most one (1) bus clock cycle after reading the TRNG0_ENT15 register. (TRNG0_ENT0 through TRNG0_ENT14 should be read before reading TRNG0_ENT15).
9 FCT_VAL	Read only: Frequency Count Valid. Indicates that a valid frequency count may be read from TRNG0_FRQCNT.
8 FCT_FAIL	Read only: Frequency Count Fail. The frequency counter has detected a failure. This may be due to improper programming of the TRNG0_FRQMAX and/or TRNG0_FRQMIN registers, or a hardware failure in the ring oscillator. This error may be cleared by writing a 1 to the ERR bit.

Table continues on the next page...

Field	Function
7 FOR_SCLK	Force System Clock. If set, the system clock is used to operate the TRNG, instead of the ring oscillator. This is for test use only, and indeterminate results may occur. This bit is writable only if PRGM bit is 1, or PRGM bit is being written to 1 simultaneously to writing this bit. This bit is cleared by writing the RST_DEF bit to 1.
6 RST_DEF	Reset Defaults. Writing a 1 to this bit clears various TRNG registers, and bits within registers, to their default state. This bit is writable only if PRGM bit is 1, or PRGM bit is being written to 1 simultaneously to writing this bit. Reading this bit always produces a 0.
5 TRNG_ACC	TRNG Access Mode. If this bit is set to 1, the TRNG will generate an Entropy value that can be read via the TRNG0_ENT0-TRNG0_ENT15 registers. The Entropy value may be read once the ENT VAL bit is asserted. Also see TRNG0_ENTa register descriptions (For a = 0 to 15).
4 UNUSED	This bit is unused but write-able. Must be left as zero.
3-2 OSC_DIV	Oscillator Divide. Determines the amount of dividing done to the ring oscillator before it is used by the TRNG. This field is writable only if PRGM bit is 1, or PRGM bit is being written to 1 simultaneously to writing this field. This field is cleared to the default POR value by writing the RST_DEF bit to 1. 00 - use ring oscillator with no divide 01 - use ring oscillator divided-by-2 10 - use ring oscillator divided-by-4 11 - use ring oscillator divided-by-8
1-0 SAMP_MODE	Sample Mode. Determines the method of sampling the ring oscillator while generating the Entropy value: This field is writable only if PRGM bit is 1, or PRGM bit is being written to 1 simultaneously with writing this field. This field is cleared to the POR default value by writing the RST_DEF bit to 1. 00 - use Von Neumann data into both Entropy shifter and Statistical Checker 01 - use raw data into both Entropy shifter and Statistical Checker 10 - use Von Neumann data into Entropy shifter. Use raw data into Statistical Checker 11 - undefined/reserved.

43.1.3.3.3 TRNG0 Statistical Check Miscellaneous (TRNG0_SCMISC)

43.1.3.3.3.1 Address

Register	Offset
TRNG0_SCMISC	4h

43.1.3.3.3.2 Function

The TRNG0 Statistical Check Miscellaneous Register contains the Long Run Maximum Limit value and the Retry Count value. This register is accessible only when the TRNG0_MCTL[PRGM] bit is 1, otherwise this register will read zeroes, and cannot be written.

43.1.3.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												RTY_CT			
Reset	0[1]	0	0	0	0	0	0	0	0	0	0	0	0[1]	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								LRUN_MAX							
Reset	0[1]	0	0	0	0	0	0	0	0[1]	0	0	1	1	1	1	1

1. Reset occurs at POR, and when TRNG0_MCTL[RST_DEF] is written to 1.

43.1.3.3.3.4 Fields

Field	Function
31-20 —	Reserved.
19-16 RTY_CT	RETRY COUNT. If a statistical check fails during the TRNG Entropy Generation, the RTY_CT value indicates the number of times a retry should occur before generating an error. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-8 —	Reserved.
7-0 LRUN_MAX	LONG RUN MAX LIMIT. This value is the largest allowable number of consecutive samples of all 1, or all 0, that is allowed during the Entropy generation. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to the POR reset value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.4 TRNG0 Poker Range (TRNG0_PKRRNG)

43.1.3.3.4.1 Address

Register	Offset
TRNG0_PKRRNG	8h

43.1.3.3.4.2 Function

The TRNG0 Poker Range Register defines the difference between the TRNG Poker Maximum Limit and the minimum limit. These limits are used during the TRNG Statistical Check Poker Test.

43.1.3.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PKR_RNG															
Reset	0	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1

43.1.3.3.4.4 Fields

Field	Function
31-16 —	Reserved. Always 0.
15-0 PKR_RNG	Poker Range. During the TRNG Statistical Checks, a "Poker Test" is run which requires a maximum and minimum limit. The maximum is programmed in the PKRMAX[PKR_MAX] register, and the minimum is derived by subtracting the PKR_RNG value from the programmed maximum value. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1. Note that the minimum allowable Poker result is PKR_MAX - PKR_RNG + 1.

43.1.3.3.5 TRNG0 Poker Maximum Limit (TRNG0_PKRMAX)

43.1.3.3.5.1 Address

Register	Offset	Description
TRNG0_PKRMAX	Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.5.2 Function

The TRNG0 Poker Maximum Limit Register defines Maximum Limit allowable during the TRNG Statistical Check Poker Test. Note that this offset (0x0C) is used as TRNG0_PKRMAX only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as the TRNG0_PKRSQ readback register.

43.1.3.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								PKR_MAX							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PKR_MAX															
Reset	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0

43.1.3.3.5.4 Fields

Field	Function
31-24 —	
23-0 PKR_MAX	Poker Maximum Limit. During the TRNG Statistical Checks, a "Poker Test" is run which requires a maximum and minimum limit. The maximum allowable result is programmed in the TRNG0_PKRMAX[PKR_MAX] register. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1. Note that the TRNG0_PKRMAX and TRNG0_PKRRNG registers combined are used to define the minimum allowable Poker result, which is PKR_MAX - PKR_RNG + 1. Note that if TRNG0_MCTL[PRGM] bit is 0, this register address is used to read the Poker Test Square Calculation result in register TRNG0_PKRSQ, as defined in the following section.

43.1.3.3.6 TRNG0 Poker Square Calculation Result (TRNG0_PKRSQ)

43.1.3.3.6.1 Address

Register	Offset	Description
TRNG0_PKRSQ	Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.6.2 Function

The TRNG0 Poker Square Calculation Result Register is a read-only register used to read the result of the TRNG Statistical Check Poker Test's Square Calculation. This test starts with the TRNG0_PKRMAX value and decreases towards a final result, which is read here. For the Poker Test to pass, this final result must be less than the programmed TRNG0_PKRRNG value. Note that this offset (0x0C) is used as TRNG0_PKRMAX if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_PKRSQ readback register, as described here.

43.1.3.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								PKR_SQ							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_SQ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.6.4 Fields

Field	Function
31-24 —	
23-0 PKR_SQ	Poker Square Calculation Result. During the TRNG Statistical Checks, a "Poker Test" is run which starts with the value TRNG0_PKRMAX[PKR_MAX]. This value decreases according to a "sum of squares" algorithm, and must remain greater than zero, but less than the TRNG0_PKRRNG[PKR_RNG] limit. The resulting value may be read through this register, if TRNG0_MCTL[PRGM] bit is 0. Note that if TRNG0_MCTL[PRGM] bit is 1, this register address is used to access the Poker Test Maximum Limit in register TRNG0_PKRMAX, as defined in the previous section.

43.1.3.3.7 TRNG0 Seed Control (TRNG0_SDCTL)

43.1.3.3.7.1 Address

Register	Offset
TRNG0_SDCTL	10h

43.1.3.3.7.2 Function

The TRNG0 Seed Control Register contains two fields. One field defines the length (in system clocks) of each Entropy sample (ENT_DLY), and the other field indicates the number of samples that will taken during each TRNG Entropy generation (SAMP_SIZE).

43.1.3.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ENT_DLY															
Reset	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SAMP_SIZE															
Reset	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0

43.1.3.3.7.4 Fields

Field	Function
31-16 ENT_DLY	Entropy Delay. Defines the length (in system clocks) of each Entropy sample taken. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to its reset value at POR.
15-0 SAMP_SIZE	Sample Size. Defines the total number of Entropy samples that will be taken during Entropy generation. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.8 TRNG0 Sparse Bit Limit (TRNG0_SBLIM)

43.1.3.3.8.1 Address

Register	Offset	Description
TRNG0_SBLIM	14h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.8.2 Function

The TRNG0 Sparse Bit Limit Register is used when Von Neumann sampling is selected during Entropy Generation. It defines the maximum number of consecutive Von Neumann samples which may be discarded before an error is generated. Note that this address (0x14) is used as TRNG0_SBLIM only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_TOTSAM readback register.

43.1.3.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						SB_LIM									
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

43.1.3.3.8.4 Fields

Field	Function
31-10 —	Reserved. Always 0.
9-0 SB_LIM	Sparse Bit Limit. During Von Neumann sampling (if enabled by TRNG0_MCTL[SAMP_MODE], samples are discarded if two consecutive raw samples are both 0 or both 1. If this discarding occurs for a long period of time, it indicates that there is insufficient Entropy. The Sparse Bit Limit defines the maximum number of consecutive samples that may be discarded before an error is generated. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1. Note that if TRNG0_MCTL[PRGM] bit is 0, this register address is used to read the Total Samples count in register TRNG0_TOTSAM, as defined in the following section.

43.1.3.3.9 TRNG0 Total Samples (TRNG0_TOTSAM)

43.1.3.3.9.1 Address

Register	Offset	Description
TRNG0_TOTSAM	14h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.9.2 Function

The TRNG0 Total Samples Register is a read-only register used to read the total number of samples taken during Entropy generation. It is used to give an indication of how often a sample is actually used during Von Neumann sampling. Note that this offset (0x14) is used as TRNG0_SBLIM if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_TOTSAM readback register, as described here.

43.1.3.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												TOT_SAM			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TOT_SAM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.9.4 Fields

Field	Function
31-20 —	Reserved. Always 0.
19-0 TOT_SAM	Total Samples. During Entropy generation, the total number of raw samples is counted. This count is useful in determining how often a sample is used during Von Neumann sampling. The count may be read through this register, if TRNG0_MCTL[PRGM] bit is 0. Note that if TRNG0_MCTL[PRGM] bit is 1, this register address is used to access the Sparse Bit Limit in register TRNG0_SBLIM, as defined in the previous section.

43.1.3.3.10 TRNG0 Frequency Count Minimum Limit (TRNG0_FRQMIN)

43.1.3.3.10.1 Address

Register	Offset
TRNG0_FRQMIN	18h

43.1.3.3.10.2 Function

The TRNG0 Frequency Count Minimum Limit Register defines the minimum allowable count taken by the Entropy sample counter during each Entropy sample. During any sample period, if the count is less than this programmed minimum, a Frequency Count Fail is flagged in TRNG0_MCTL[FCT_FAIL] and an error is generated.

43.1.3.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										FRQ_MIN					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FRQ_MIN															
Reset	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0

43.1.3.3.10.4 Fields

Field	Function
31-22 —	Reserved. Always 0.
21-0 FRQ_MIN	Frequency Count Minimum Limit. Defines the minimum allowable count taken during each entropy sample. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field will read zeroes if TRNG0_MCTL[PRGM] = 0. This field is cleared to its reset value at POR.

43.1.3.3.11 TRNG0 Frequency Count (TRNG0_FRQCNT)

43.1.3.3.11.1 Address

Register	Offset	Description
TRNG0_FRQCNT	1Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.11.2 Function

The TRNG0 Frequency Count Register is a read-only register used to read the frequency counter within the TRNG entropy generator. It will read all zeroes unless TRNG0_MCTL[TRNG_ACC] = 1. Note that this offset (0x1C) is used as TRNG0_FRQMAX if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_FRQCNT readback register, as described here.

43.1.3.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										FRQ_CT					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FRQ_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.11.4 Fields

Field	Function
31-22 —	Reserved. Always 0.
21-0 FRQ_CT	Frequency Count. If TRNG0_MCTL[TRNG_ACC] = 1, reads a sample frequency count taken during entropy generation. Requires TRNG0_MCTL[PRGM] = 0. Note that if TRNG0_MCTL[PRGM] bit is 1, this register address is used to access the Poker Test Maximum Limit in register TRNG0_PKRMAX, as defined in the previous section.

43.1.3.3.12 TRNG0 Frequency Count Maximum Limit (TRNG0_FRQMAX)

43.1.3.3.12.1 Address

Register	Offset	Description
TRNG0_FRQMAX	1Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.12.2 Function

The TRNG0 Frequency Count Maximum Limit Register defines the maximum allowable count taken by the Entropy sample counter during each Entropy sample. During any sample period, if the count is greater than this programmed maximum, a Frequency Count Fail is flagged in TRNG0_MCTL[FCT_FAIL] and an error is generated. Note that this address (001C) is used as TRNG0_FRQMAX only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_FRQCNT readback register.

43.1.3.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										FRQ_MAX					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FRQ_MAX															
Reset	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

43.1.3.3.12.4 Fields

Field	Function
31-22 —	Reserved. Always 0.
21-0 FRQ_MAX	Frequency Counter Maximum Limit. Defines the maximum allowable count taken during each entropy sample. This field is writable only if TRNG0_MCTL[PRGM] bit is 1. This field is cleared to its reset value at POR. Note that if TRNG0_MCTL[PRGM] bit is 0, this register address is used to read the Frequency Count result in register TRNG0_FRQCNT, as defined in the following section.

43.1.3.3.13 TRNG0 Statistical Check Monobit Count (TRNG0_SCMC)

43.1.3.3.13.1 Address

Register	Offset	Description
TRNG0_SCMC	20h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.13.2 Function

The TRNG0 Statistical Check Monobit Count Register is a read-only register used to read the final monobit count after entropy generation. This counter starts with the value in TRNG0_SCML[MONO_MAX], and is decremented each time a one is sampled. Note that this offset (0x20) is used as TRNG0_SCML if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCMC readback register, as described here.

43.1.3.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MONO_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.13.4 Fields

Field	Function
31-16 —	Reserved. Always 0.
15-0 MONO_CT	Monobit Count. Reads the final Monobit count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0. Note that if TRNG0_MCTL[PRGM] bit is 1, this register address is used to access the Statistical Check Monobit Limit in register TRNG0_SCML, as defined in the previous section.

43.1.3.3.14 TRNG0 Statistical Check Monobit Limit (TRNG0_SCML)

43.1.3.3.14.1 Address

Register	Offset	Description
TRNG0_SCML	20h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.14.2 Function

The TRNG0 Statistical Check Monobit Limit Register defines the allowable maximum and minimum number of ones/zero detected during entropy generation. To pass the test, the number of ones/zeroes generated must be less than the programmed maximum value, and the number of ones/zeroes generated must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this offset (0x20) is used as TRNG0_SCML only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCMC readback register.

43.1.3.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MONO_RNG															
Reset	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MONO_MAX															
Reset	0	0	0	0	0	1	0	1	0	1	1	0	1	0	0	0

43.1.3.3.14.4 Fields

Field	Function
31-16 MONO_RNG	Monobit Range. The number of ones/zeroes detected during entropy generation must be greater than MONO_MAX - MONO_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

Table continues on the next page...

Field	Function
15-0 MONO_MAX	Monobit Maximum Limit. Defines the maximum allowable count taken during entropy generation. The number of ones/zeroes detected during entropy generation must be less than MONO_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.15 TRNG0 Statistical Check Run Length 1 Count (TRNG0_SCR1C)

43.1.3.3.15.1 Address

Register	Offset	Description
TRNG0_SCR1C	24h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.15.2 Function

The TRNG0 Statistical Check Run Length 1 Counters Register is a read-only register used to read the final Run Length 1 counts after entropy generation. These counters start with the value in TRNG0_SCRxC1L[RUN1_MAX]. The R1_1_CT decrements each time a single one is sampled (preceded by a zero and followed by a zero). The R1_0_CT decrements each time a single zero is sampled (preceded by a one and followed by a one). Note that this offset (0x24) is used as TRNG0_SCRxC1L if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC1C readback register, as described here.

43.1.3.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	R1_1_CT														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	R1_0_CT														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.15.4 Fields

Field	Function
31 —	Reserved. Always 0.
30-16 R1_1_CT	Runs of One, Length 1 Count. Reads the final Runs of Ones, length 1 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15 —	Reserved. Always 0.
14-0 R1_0_CT	Runs of Zero, Length 1 Count. Reads the final Runs of Zeroes, length 1 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.16 TRNG0 Statistical Check Run Length 1 Limit (TRNG0_SCR1L)

43.1.3.3.16.1 Address

Register	Offset	Description
TRNG0_SCR1L	24h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.16.2 Function

The TRNG0 Statistical Check Run Length 1 Limit Register defines the allowable maximum and minimum number of runs of length 1 detected during entropy generation. To pass the test, the number of runs of length 1 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 1 must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x24) is used as TRNG0_SCRxC1L only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_SCRxC1C readback register.

43.1.3.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	RUN1_RNG														
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	RUN1_MAX														
Reset	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1

43.1.3.3.16.4 Fields

Field	Function
31 —	Reserved. Always 0.
30-16 RUN1_RNG	Run Length 1 Range. The number of runs of length 1 (for both 0 and 1) detected during entropy generation must be greater than RUN1_MAX - RUN1_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15 —	Reserved. Always 0.
14-0 RUN1_MAX	Run Length 1 Maximum Limit. Defines the maximum allowable runs of length 1 (for both 0 and 1) detected during entropy generation. The number of runs of length 1 detected during entropy generation must be less than RUN1_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.17 TRNG0 Statistical Check Run Length 2 Count (TRNG0_SCR2C)

43.1.3.3.17.1 Address

Register	Offset	Description
TRNG0_SCR2C	28h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.17.2 Function

The TRNG0 Statistical Check Run Length 2 Counters Register is a read-only register used to read the final Run Length 2 counts after entropy generation. These counters start with the value in TRNG0_SCRxC2L[RUN2_MAX]. The R2_1_CT decrements each time two consecutive ones are sampled (preceded by a zero and followed by a zero). The

R2_0_CT decrements each time two consecutive zeroes are sampled (preceded by a one and followed by a one). Note that this offset (0x28) is used as TRNG0_SCRxC2L if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC2C readback register, as described here.

43.1.3.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		R2_1_CT													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		R2_0_CT													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.17.4 Fields

Field	Function
31-30 —	Reserved. Always 0.
29-16 R2_1_CT	Runs of One, Length 2 Count. Reads the final Runs of Ones, length 2 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15-14 —	Reserved. Always 0.
13-0 R2_0_CT	Runs of Zero, Length 2 Count. Reads the final Runs of Zeroes, length 2 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.18 TRNG0 Statistical Check Run Length 2 Limit (TRNG0_SCR2L)

43.1.3.3.18.1 Address

Register	Offset	Description
TRNG0_SCR2L	28h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.18.2 Function

The TRNG0 Statistical Check Run Length 2 Limit Register defines the allowable maximum and minimum number of runs of length 2 detected during entropy generation. To pass the test, the number of runs of length 2 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 2 must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x28) is used as TRNG0_SCRxC2L only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_SCRxC2C readback register.

43.1.3.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		RUN2_RNG													
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		RUN2_MAX													
Reset	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0

43.1.3.3.18.4 Fields

Field	Function
31-30 —	Reserved. Always 0.
29-16 RUN2_RNG	Run Length 2 Range. The number of runs of length 2 (for both 0 and 1) detected during entropy generation must be greater than RUN2_MAX - RUN2_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-14 —	Reserved. Always 0.
13-0 RUN2_MAX	Run Length 2 Maximum Limit. Defines the maximum allowable runs of length 2 (for both 0 and 1) detected during entropy generation. The number of runs of length 2 detected during entropy generation must be less than RUN2_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.19 TRNG0 Statistical Check Run Length 3 Count (TRNG0_SCR3C)

43.1.3.3.19.1 Address

Register	Offset	Description
TRNG0_SCR3C	2Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.19.2 Function

The TRNG0 Statistical Check Run Length 3 Counters Register is a read-only register used to read the final Run Length 3 counts after entropy generation. These counters start with the value in TRNG0_SCRxC3L[RUN3_MAX]. The R3_1_CT decrements each time three consecutive ones are sampled (preceded by a zero and followed by a zero). The R3_0_CT decrements each time three consecutive zeroes are sampled (preceded by a one and followed by a one). Note that this offset (0x2C) is used as TRNG0_SCRxC3L if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC3C readback register, as described here.

43.1.3.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			R3_1_CT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			R3_0_CT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.19.4 Fields

Field	Function
31-29 —	Reserved. Always 0.
28-16 R3_1_CT	Runs of Ones, Length 3 Count. Reads the final Runs of Ones, length 3 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15-13	Reserved. Always 0.

Table continues on the next page...

Field	Function
—	
12-0 R3_0_CT	Runs of Zeroes, Length 3 Count. Reads the final Runs of Zeroes, length 3 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.20 TRNG0 Statistical Check Run Length 3 Limit (TRNG0_SCR3L)

43.1.3.3.20.1 Address

Register	Offset	Description
TRNG0_SCR3L	2Ch	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.20.2 Function

The TRNG0 Statistical Check Run Length 3 Limit Register defines the allowable maximum and minimum number of runs of length 3 detected during entropy generation. To pass the test, the number of runs of length 3 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 3 must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x2C) is used as TRNG0_SCRxC3L only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_SCRxC3C readback register.

43.1.3.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			RUN3_RNG												
Reset	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			RUN3_MAX												
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1

43.1.3.3.20.4 Fields

Field	Function
31-29 —	Reserved. Always 0.
28-16 RUN3_RNG	Run Length 3 Range. The number of runs of length 3 (for both 0 and 1) detected during entropy generation must be greater than RUN3_MAX - RUN3_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-13 —	Reserved. Always 0.
12-0 RUN3_MAX	Run Length 3 Maximum Limit. Defines the maximum allowable runs of length 3 (for both 0 and 1) detected during entropy generation. The number of runs of length 3 detected during entropy generation must be less than RUN3_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.21 TRNG0 Statistical Check Run Length 4 Count (TRNG0_SCR4C)

43.1.3.3.21.1 Address

Register	Offset	Description
TRNG0_SCR4C	30h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.21.2 Function

The TRNG0 Statistical Check Run Length 4 Counters Register is a read-only register used to read the final Run Length 4 counts after entropy generation. These counters start with the value in TRNG0_SCRxC4L[RUN4_MAX]. The R4_1_CT decrements each time four consecutive ones are sampled (preceded by a zero and followed by a zero). The R4_0_CT decrements each time four consecutive zeroes are sampled (preceded by a one and followed by a one). Note that this offset (0x30) is used as TRNG0_SCRxC4L if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC4C readback register, as described here.

43.1.3.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				R4_1_CT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				R4_0_CT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.21.4 Fields

Field	Function
31-28 —	Reserved. Always 0.
27-16 R4_1_CT	Runs of One, Length 4 Count. Reads the final Runs of Ones, length 4 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15-12 —	Reserved. Always 0.
11-0 R4_0_CT	Runs of Zero, Length 4 Count. Reads the final Runs of Ones, length 4 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.22 TRNG0 Statistical Check Run Length 4 Limit (TRNG0_SCR4L)

43.1.3.3.22.1 Address

Register	Offset	Description
TRNG0_SCR4L	30h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.22.2 Function

The TRNG0 Statistical Check Run Length 4 Limit Register defines the allowable maximum and minimum number of runs of length 4 detected during entropy generation. To pass the test, the number of runs of length 4 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 4 must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x30) is used as TRNG0_SCRxC4L only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_SCRxC4C readback register.

43.1.3.3.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				RUN4_RNG											
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				RUN4_MAX											
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1

43.1.3.3.22.4 Fields

Field	Function
31-28 —	Reserved. Always 0.
27-16 RUN4_RNG	Run Length 4 Range. The number of runs of length 4 (for both 0 and 1) detected during entropy generation must be greater than RUN4_MAX - RUN4_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-12 —	Reserved. Always 0.
11-0 RUN4_MAX	Run Length 4 Maximum Limit. Defines the maximum allowable runs of length 4 (for both 0 and 1) detected during entropy generation. The number of runs of length 4 detected during entropy generation must be less than RUN4_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.23 TRNG0 Statistical Check Run Length 5 Count (TRNG0_SCR5C)

43.1.3.3.23.1 Address

Register	Offset	Description
TRNG0_SCR5C	34h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.23.2 Function

The TRNG0 Statistical Check Run Length 5 Counters Register is a read-only register used to read the final Run Length 5 counts after entropy generation. These counters start with the value in TRNG0_SCRxC5L[RUN5_MAX]. The R5_1_CT decrements each time five consecutive ones are sampled (preceded by a zero and followed by a zero). The R5_0_CT decrements each time five consecutive zeroes are sampled (preceded by a one and followed by a one). Note that this offset (0x34) is used as TRNG0_SCRxC5L if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC5C readback register, as described here.

43.1.3.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					R5_1_CT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					R5_0_CT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.23.4 Fields

Field	Function
31-27 —	Reserved. Always 0.
26-16 R5_1_CT	Runs of One, Length 5 Count. Reads the final Runs of Ones, length 5 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15-11 —	Reserved. Always 0.
10-0 R5_0_CT	Runs of Zero, Length 5 Count. Reads the final Runs of Ones, length 5 count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.24 TRNG0 Statistical Check Run Length 5 Limit (TRNG0_SCR5L)

43.1.3.3.24.1 Address

Register	Offset	Description
TRNG0_SCR5L	34h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.24.2 Function

The TRNG0 Statistical Check Run Length 5 Limit Register defines the allowable maximum and minimum number of runs of length 5 detected during entropy generation. To pass the test, the number of runs of length 5 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 5 must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x34) is used as TRNG0_SCRxC5L only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this address is used as TRNG0_SCRxC5C readback register.

43.1.3.3.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					RUN5_RNG										
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					RUN5_MAX										
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1

43.1.3.3.24.4 Fields

Field	Function
31-27 —	Reserved. Always 0.
26-16 RUN5_RNG	Run Length 5 Range. The number of runs of length 5 (for both 0 and 1) detected during entropy generation must be greater than RUN5_MAX - RUN5_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-11	Reserved. Always 0.

Table continues on the next page...

Field	Function
—	
10-0 RUN5_MAX	Run Length 5 Maximum Limit. Defines the maximum allowable runs of length 5 (for both 0 and 1) detected during entropy generation. The number of runs of length 5 detected during entropy generation must be less than RUN5_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.25 TRNG0 Statistical Check Run Length 6+ Count (TRNG0_SCR6PC)

43.1.3.3.25.1 Address

Register	Offset	Description
TRNG0_SCR6PC	38h	Accessible at this address when TRNG0_MCTL[PRGM] = 0]

43.1.3.3.25.2 Function

The TRNG0 Statistical Check Run Length 6+ Counters Register is a read-only register used to read the final Run Length 6+ counts after entropy generation. These counters start with the value in TRNG0_SCRxC6PL[RUN6P_MAX]. The R6P_1_CT decrements each time six or more consecutive ones are sampled (preceded by a zero and followed by a zero). The R6P_0_CT decrements each time six or more consecutive zeroes are sampled (preceded by a one and followed by a one). Note that this offset (0x38) is used as TRNG0_SCRxC6PL if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC6PC readback register, as described here.

43.1.3.3.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					R6P_1_CT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					R6P_0_CT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.25.4 Fields

Field	Function
31-27 —	Reserved. Always 0.
26-16 R6P_1_CT	Runs of One, Length 6+ Count. Reads the final Runs of Ones, length 6+ count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.
15-11 —	Reserved. Always 0.
10-0 R6P_0_CT	Runs of Zero, Length 6+ Count. Reads the final Runs of Ones, length 6+ count after entropy generation. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.26 TRNG0 Statistical Check Run Length 6+ Limit (TRNG0_SCR6PL)

43.1.3.3.26.1 Address

Register	Offset	Description
TRNG0_SCR6PL	38h	Accessible at this address when TRNG0_MCTL[PRGM] = 1]

43.1.3.3.26.2 Function

The TRNG0 Statistical Check Run Length 6+ Limit Register defines the allowable maximum and minimum number of runs of length 6 or more detected during entropy generation. To pass the test, the number of runs of length 6 or more (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 6 or more must be greater than (maximum - range). If this test fails, the Retry Counter in TRNG0_SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this offset (0x38) is used as TRNG0_SCRxC6PL only if TRNG0_MCTL[PRGM] is 1. If TRNG0_MCTL[PRGM] is 0, this offset is used as TRNG0_SCRxC6PC readback register.

43.1.3.3.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					RUN6P_RNG										
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					RUN6P_MAX										
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1

43.1.3.3.26.4 Fields

Field	Function
31-27 —	Reserved. Always 0.
26-16 RUN6P_RNG	Run Length 6+ Range. The number of runs of length 6 or more (for both 0 and 1) detected during entropy generation must be greater than RUN6P_MAX - RUN6P_RNG, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.
15-11 —	Reserved. Always 0.
10-0 RUN6P_MAX	Run Length 6+ Maximum Limit. Defines the maximum allowable runs of length 6 or more (for both 0 and 1) detected during entropy generation. The number of runs of length 6 or more detected during entropy generation must be less than RUN6P_MAX, else a retry or error will occur. This register is cleared to the default POR value by writing the TRNG0_MCTL[RST_DEF] bit to 1.

43.1.3.3.27 TRNG0 Status (TRNG0_STATUS)

43.1.3.3.27.1 Address

Register	Offset
TRNG0_STATUS	3Ch

43.1.3.3.27.2 Function

Various statistical tests are run as a normal part of the TRNG's entropy generation process. The least-significant 16 bits of the TRNG0_STATUS register reflect the result of each of these tests. The status of these bits will be valid when the TRNG has finished its entropy generation process. Software can determine when this occurs by polling the ENT_VAL bit in the TRNG0 Miscellaneous Control Register.

Note that there is a very small probability that a statistical test will fail even though the TRNG is operating properly. If this happens the TRNG will automatically retry the entire entropy generation process, including running all the statistical tests. The value in RETRY_CT is decremented each time an entropy generation retry occurs. If a statistical check fails when the retry count is nonzero, a retry is initiated. But if a statistical check fails when the retry count is zero, an error is generated by the RNG. By default RETRY_CT is initialized to 1, but software can increase the retry count by writing to the RTY_CT field in the TRNG0_SCMISC register.

All 0s will be returned if this register address is read while the RNG is in Program Mode (see PRGM field in TRNG0_MCTL register. If this register is read while the RNG is in Run Mode the value returned will be formatted as follows.

43.1.3.3.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												RETRY_CT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFMB	TFP	TFLR	TFSB	TF6PBR1	TF6PBR0	TF5BR1	TF5BR0	TF4BR1	TF4BR0	TF3BR1	TF3BR0	TF2BR1	TF2BR0	TF1BR1	TF1BR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.27.4 Fields

Field	Function
31-20 —	Reserved. Always 0.
19-16 RETRY_CT	RETRY COUNT. This represents the current number of entropy generation retries left before a statistical text failure will cause the RNG to generate an error condition.
15	Test Fail, Mono Bit. If TFMB=1, the Mono Bit Test has failed.

Table continues on the next page...

Field	Function
TFMB	
14 TFP	Test Fail, Poker. If TFP=1, the Poker Test has failed.
13 TFLR	Test Fail, Long Run. If TFLR=1, the Long Run Test has failed.
12 TFSB	Test Fail, Sparse Bit. If TFSB=1, the Sparse Bit Test has failed.
11 TF6PBR1	Test Fail, 6 Plus Bit Run, Sampling 1s. If TF6PBR1=1, the 6 Plus Bit Run, Sampling 1s Test has failed.
10 TF6PBR0	Test Fail, 6 Plus Bit Run, Sampling 0s. If TF6PBR0=1, the 6 Plus Bit Run, Sampling 0s Test has failed.
9 TF5BR1	Test Fail, 5-Bit Run, Sampling 1s. If TF5BR1=1, the 5-Bit Run, Sampling 1s Test has failed.
8 TF5BR0	Test Fail, 5-Bit Run, Sampling 0s. If TF5BR0=1, the 5-Bit Run, Sampling 0s Test has failed.
7 TF4BR1	Test Fail, 4-Bit Run, Sampling 1s. If TF4BR1=1, the 4-Bit Run, Sampling 1s Test has failed.
6 TF4BR0	Test Fail, 4-Bit Run, Sampling 0s. If TF4BR0=1, the 4-Bit Run, Sampling 0s Test has failed.
5 TF3BR1	Test Fail, 3-Bit Run, Sampling 1s. If TF3BR1=1, the 3-Bit Run, Sampling 1s Test has failed.
4 TF3BR0	Test Fail, 3-Bit Run, Sampling 0s. If TF3BR0=1, the 3-Bit Run, Sampling 0s Test has failed.
3 TF2BR1	Test Fail, 2-Bit Run, Sampling 1s. If TF2BR1=1, the 2-Bit Run, Sampling 1s Test has failed.
2 TF2BR0	Test Fail, 2-Bit Run, Sampling 0s. If TF2BR0=1, the 2-Bit Run, Sampling 0s Test has failed.
1 TF1BR1	Test Fail, 1-Bit Run, Sampling 1s. If TF1BR1=1, the 1-Bit Run, Sampling 1s Test has failed.
0 TF1BR0	Test Fail, 1-Bit Run, Sampling 0s. If TF1BR0=1, the 1-Bit Run, Sampling 0s Test has failed.

43.1.3.3.28 TRNG0 Entropy Read (TRNG0_ENTa)

43.1.3.3.28.1 Address

For a = 0 to 15:

Register	Offset	Description
TRNG0_ENTa	40h + (a × 4h)	Word a

43.1.3.3.28.2 Function

The RNG TRNG can be programmed to generate an entropy value that is readable via the SkyBlue bus. To do this, set the TRNG0_MCTL[TRNG_ACC] bit to 1. Once the entropy value has been generated, the TRNG0_MCTL[ENT_VAL] bit will be set to 1. At this point, TRNG0_ENT0 through TRNG0_ENT15 may be read to retrieve the 512-bit entropy value. Note that once TRNG0_ENT15 is read, the entropy value will be cleared and a new value will begin generation, so it is important that TRNG0_ENT15 be read last. These registers are readable only when TRNG0_MCTL[PRGM] = 0 (Run Mode), TRNG0_MCTL[TRNG_ACC] = 1 (TRNG access mode) and TRNG0_MCTL[ENT_VAL] = 1. After at most one (1) bus clock cycle of reading a valid TRNG0_ENT15 register value, reading any TRNG0_ENT0 through TRNG0_ENT15 register would return zeroes.

43.1.3.3.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ENT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.28.4 Fields

Field	Function
31-0 ENT	Entropy Value. Will be non-zero only if TRNG0_MCTL[PRGM] = 0 Run Mode) and TRNG0_MCTL[ENT_VAL] = 1 (Entropy Valid). The most significant bits of the entropy are read from the lowest offset, and the least significant bits are read from the highest offset. Note that reading the highest offset also clears the entire entropy value, and starts a new entropy generation.

43.1.3.3.29 TRNG0 Statistical Check Poker Count 1 and 0 (TRNG0_PKRCNT10)

43.1.3.3.29.1 Address

Register	Offset
TRNG0_PKRCNT10	80h

43.1.3.3.29.2 Function

The TRNG0 Statistical Check Poker Count 1 and 0 Register is a read-only register used to read the final Poker test counts of 1h and 0h patterns. The Poker 0h Count increments each time a nibble of sample data is found to be 0h. The Poker 1h Count increments each time a nibble of sample data is found to be 1h. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_1_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_0_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.29.4 Fields

Field	Function
31-16 PKR_1_CT	Poker 1h Count. Total number of nibbles of sample data which were found to be 1h. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_0_CT	Poker 0h Count. Total number of nibbles of sample data which were found to be 0h. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.30 TRNG0 Statistical Check Poker Count 3 and 2 (TRNG0_PKRCNT32)

43.1.3.3.30.1 Address

Register	Offset
TRNG0_PKRCNT32	84h

43.1.3.3.30.2 Function

The TRNG0 Statistical Check Poker Count 3 and 2 Register is a read-only register used to read the final Poker test counts of 3h and 2h patterns. The Poker 2h Count increments each time a nibble of sample data is found to be 2h. The Poker 3h Count increments each time a nibble of sample data is found to be 3h. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_3_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_2_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.30.4 Fields

Field	Function
31-16 PKR_3_CT	Poker 3h Count. Total number of nibbles of sample data which were found to be 3h. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_2_CT	Poker 2h Count. Total number of nibbles of sample data which were found to be 2h. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.31 TRNG0 Statistical Check Poker Count 5 and 4 (TRNG0_PKRCNT54)

43.1.3.3.31.1 Address

Register	Offset
TRNG0_PKRCNT54	88h

43.1.3.3.31.2 Function

The TRNG0 Statistical Check Poker Count 5 and 4 Register is a read-only register used to read the final Poker test counts of 5h and 4h patterns. The Poker 4h Count increments each time a nibble of sample data is found to be 4h. The Poker 5h Count increments each time a nibble of sample data is found to be 5h. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_5_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_4_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.31.4 Fields

Field	Function
31-16 PKR_5_CT	Poker 5h Count. Total number of nibbles of sample data which were found to be 5h. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_4_CT	Poker 4h Count. Total number of nibbles of sample data which were found to be 4h. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.32 TRNG0 Statistical Check Poker Count 7 and 6 (TRNG0_PKRCNT76)

43.1.3.3.32.1 Address

Register	Offset
TRNG0_PKRCNT76	8Ch

43.1.3.3.32.2 Function

The TRNG0 Statistical Check Poker Count 7 and 6 Register is a read-only register used to read the final Poker test counts of 7h and 6h patterns. The Poker 6h Count increments each time a nibble of sample data is found to be 6h. The Poker 7h Count increments each time a nibble of sample data is found to be 7h. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_7_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_6_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.32.4 Fields

Field	Function
31-16 PKR_7_CT	Poker 7h Count. Total number of nibbles of sample data which were found to be 7h. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_6_CT	Poker 6h Count. Total number of nibbles of sample data which were found to be 6h. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.33 TRNG0 Statistical Check Poker Count 9 and 8 (TRNG0_PKRCNT98)

43.1.3.3.33.1 Address

Register	Offset
TRNG0_PKRCNT98	90h

43.1.3.3.33.2 Function

The TRNG0 Statistical Check Poker Count 9 and 8 Register is a read-only register used to read the final Poker test counts of 9h and 8h patterns. The Poker 8h Count increments each time a nibble of sample data is found to be 8h. The Poker 9h Count increments each time a nibble of sample data is found to be 9h. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_9_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_8_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.33.4 Fields

Field	Function
31-16 PKR_9_CT	Poker 9h Count. Total number of nibbles of sample data which were found to be 9h. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_8_CT	Poker 8h Count. Total number of nibbles of sample data which were found to be 8h. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.34 TRNG0 Statistical Check Poker Count B and A (TRNG0_PKRCNTBA)

43.1.3.3.34.1 Address

Register	Offset
TRNG0_PKRCNTBA	94h

43.1.3.3.34.2 Function

The TRNG0 Statistical Check Poker Count B and A Register is a read-only register used to read the final Poker test counts of Bh and Ah patterns. The Poker Ah Count increments each time a nibble of sample data is found to be Ah. The Poker Bh Count increments each time a nibble of sample data is found to be Bh. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_B_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_A_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.34.4 Fields

Field	Function
31-16 PKR_B_CT	Poker Bh Count. Total number of nibbles of sample data which were found to be Bh. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_A_CT	Poker Ah Count. Total number of nibbles of sample data which were found to be Ah. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.35 TRNG0 Statistical Check Poker Count D and C (TRNG0_PKRCNTDC)

43.1.3.3.35.1 Address

Register	Offset
TRNG0_PKRCNTDC	98h

43.1.3.3.35.2 Function

The TRNG0 Statistical Check Poker Count D and C Register is a read-only register used to read the final Poker test counts of Dh and Ch patterns. The Poker Ch Count increments each time a nibble of sample data is found to be Ch. The Poker Dh Count increments each time a nibble of sample data is found to be Dh. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_D_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_C_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.35.4 Fields

Field	Function
31-16 PKR_D_CT	Poker Dh Count. Total number of nibbles of sample data which were found to be Dh. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_C_CT	Poker Ch Count. Total number of nibbles of sample data which were found to be Ch. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.36 TRNG0 Statistical Check Poker Count F and E (TRNG0_PKRCNTFE)

43.1.3.3.36.1 Address

Register	Offset
TRNG0_PKRCNTFE	9Ch

43.1.3.3.36.2 Function

The TRNG0 Statistical Check Poker Count F and E Register is a read-only register used to read the final Poker test counts of Fh and Eh patterns. The Poker Eh Count increments each time a nibble of sample data is found to be Eh. The Poker Fh Count increments each time a nibble of sample data is found to be Fh. Note that this register is readable only if TRNG0_MCTL[PRGM] is 0, otherwise zeroes will be read.

43.1.3.3.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_F_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_E_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.36.4 Fields

Field	Function
31-16 PKR_F_CT	Poker Fh Count. Total number of nibbles of sample data which were found to be Fh. Requires TRNG0_MCTL[PRGM] = 0.
15-0 PKR_E_CT	Poker Eh Count. Total number of nibbles of sample data which were found to be Eh. Requires TRNG0_MCTL[PRGM] = 0.

43.1.3.3.37 TRNG0 Security Configuration (TRNG0_SEC_CFG)

43.1.3.3.37.1 Address

Register	Offset
TRNG0_SEC_CFG	B0h

43.1.3.3.37.2 Function

The TRNG0 Security Configuration Register is a read/write register used to control the test mode, programmability and state modes of the TRNG0. Many bits are place holders for this version. More configurability will be added here. Clears on asynchronous reset. For TRNG0 releases before 2014/July/01, offsets 0xA0 to 0xAC used to be 0xB0 to 0xBC respectively. So, update newer tests that use these registers, if hard coded.

43.1.3.3.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													SK_VAL	NO_PRGM	SH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.37.4 Fields

Field	Function
31-3 —	Reserved.
2 SK_VAL	Reserved. DRNG-specific, not applicable to this version. 0 - See DRNG version. 1 - See DRNG version.
1 NO_PRGM	If set, the TRNG registers cannot be programmed. That is, regardless of the TRNG access mode in the TRNG0 Miscellaneous Control Register. 0 - Programability of registers controlled only by the TRNG0 Miscellaneous Control Register's access mode bit. 1 - Overides TRNG0 Miscellaneous Control Register access mode and prevents TRNG register programming.
0 SH0	Reserved. DRNG specific, not applicable to this version. 0 - See DRNG version. 1 - See DRNG version.

43.1.3.3.38 TRNG0 Interrupt Control (TRNG0_INT_CTRL)

43.1.3.3.38.1 Address

Register	Offset
TRNG0_INT_CTRL	B4h

43.1.3.3.38.2 Function

The TRNG0 Interrupt Control Register is a read/write register used to control the status for the (currently) three important interrupts that are generated by the TRNG. See TRNG0_INT_STATUS register description above. Each interrupt can be cleared by de-asserting the corresponding bit in the TRNG0_INT_CTRL register. Only a new interrupt will reassert the corresponding bit in the status register. Even if the interrupt is cleared or masked, interrupt status information can be read from the TRNG0_MCTL register.

43.1.3.3.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UNUSED															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UNUSED													FRQ_CT_FAIL	ENT_VAL	HW_ERR
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

43.1.3.3.38.4 Fields

Field	Function
31-3 UNUSED	Reserved but writeable.
2 FRQ_CT_FAIL	Same behavior as bit 0 above. 0 - Same behavior as bit 0 above.

Table continues on the next page...

Field	Function
	1 - Same behavior as bit 0 above.
1 ENT_VAL	Same behavior as bit 0 above. 0 - Same behavior as bit 0 above. 1 - Same behavior as bit 0 above.
0 HW_ERR	Bit position that can be cleared if corresponding bit of TRNG0_INT_STATUS has been asserted. 0 - Corresponding bit of TRNG0_INT_STATUS cleared. 1 - Corresponding bit of TRNG0_INT_STATUS active.

43.1.3.3.39 TRNG0 Mask (TRNG0_INT_MASK)

43.1.3.3.39.1 Address

Register	Offset
TRNG0_INT_MASK	B8h

43.1.3.3.39.2 Function

The TRNG0 Interrupt Mask Register is a read/write register used to disable/mask the status reporting of the (currently) three important interrupts that are generated by the TRNG. See TRNG0_INT_STATUS register description above. Each interrupt can be masked/disabled by de-asserting the corresponding bit in the TRNG0_INT_MASK register. Only setting this bit high will re-enable the interrupt in the status register. Even if the interrupt is cleared or masked, interrupt status information can be read from the TRNG0_MCTL register.

43.1.3.3.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													FRQ_CT_FAIL	ENT_VAL	HW_ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.39.4 Fields

Field	Function
31-3 —	Reserved.
2 FRQ_CT_FAIL	Same behavior as bit 0 above. 0 - Same behavior as bit 0 above. 1 - Same behavior as bit 0 above.
1 ENT_VAL	Same behavior as bit 0 above. 0 - Same behavior as bit 0 above. 1 - Same behavior as bit 0 above.
0 HW_ERR	Bit position that can be cleared if corresponding bit of TRNG0_INT_STATUS has been asserted. 0 - Corresponding interrupt of TRNG0_INT_STATUS is masked. 1 - Corresponding bit of TRNG0_INT_STATUS is active.

43.1.3.3.40 TRNG0 Interrupt Status (TRNG0_INT_STATUS)

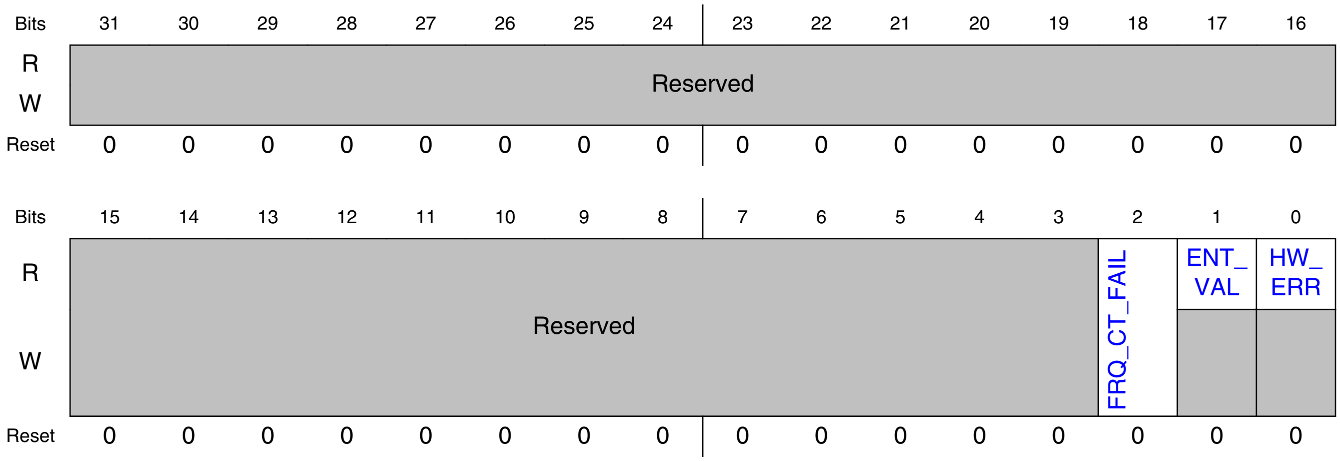
43.1.3.3.40.1 Address

Register	Offset
TRNG0_INT_STATUS	BCh

43.1.3.3.40.2 Function

The TRNG0 Interrupt Status Register is a read register used to control and provide status for the (currently) three important interrupts that are generated by the TRNG. The ipi_rng_int_b interrupt signals that TRNG0 has either generated a Frequency Count Fail, Entropy Valid or Error Interrupt. The cause of the interrupt can be decoded by checking the least significant bits of the TRNG0_INT_STATUS register. Each interrupt can be temporarily cleared by de-asserting the corresponding bit in the TRNG0_INT_CTRL register. To mask the interrupts, clear the corresponding bits in the TRNG0_INT_MASK register. The description of each of the 3 interrupts is defined in the Block Guide under the TRNG0_MCTL register description. Even if the interrupt is cleared or masked, interrupt status information can be read from the TRNG0_MCTL register.

43.1.3.3.40.3 Diagram



Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													FRQ_CT_FAIL	ENT_ VAL	HW_ ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.40.4 Fields

Field	Function
31-3 —	Reserved.
2 FRQ_CT_FAIL	Read only: Frequency Count Fail. The frequency counter has detected a failure. This may be due to improper programming of the TRNG0_FRQMAX and/or TRNG0_FRQMIN registers, or a hardware failure in the ring oscillator. 0 - No hardware nor self test frequency errors. 1 - The frequency counter has detected a failure.
1 ENT_VAL	Read only: Entropy Valid. Will assert only if TRNG ACC bit is set, and then after an entropy value is generated. Will be cleared when TRNG0_ENT15 is read. (TRNG0_ENT0 through TRNG0_ENT14 should be read before reading TRNG0_ENT15). 0 - Busy generation entropy. Any value read is invalid. 1 - TRNG can be stopped and entropy is valid if read.
0 HW_ERR	Read: Error status. 1 = error detected. 0 = no error. Any HW error in the TRNG will trigger this interrupt. 0 - no error 1 - error detected.

43.1.3.3.41 TRNG0 Version ID (MS) (TRNG0_VID1)

43.1.3.3.41.1 Address

Register	Offset
TRNG0_VID1	F0h

43.1.3.3.41.2 Function

The TRNG0 Version ID Register is a read only register used to identify the version of the TRNG in use. This register as well as TRNG0_VID2 should both be read to verify the expected version.

43.1.3.3.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRNG0_IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRNG0_MAJ_REV								TRNG0_MIN_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

43.1.3.3.41.4 Fields

Field	Function
31-16 TRNG0_IP_ID	Shows the Freescale IP ID. 0000000000110000 - ID for TRNG.
15-8 TRNG0_MAJ_REV	Shows the Freescale IP's Major revision of the TRNG. 00000001 - Major revision number for TRNG.
7-0 TRNG0_MIN_REV	Shows the Freescale IP's Minor revision of the TRNG. 00000000 - Minor revision number for TRNG.

43.1.3.3.42 TRNG0 Version ID (LS) (TRNG0_VID2)

43.1.3.3.42.1 Address

Register	Offset
TRNG0_VID2	F4h

43.1.3.3.42.2 Function

The TRNG0 Version ID Register LSB is a read only register used to identify the architecture of the TRNG in use. This register as well as TRNG0_VID1 should both be read to verify the expected version.

43.1.3.3.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRNG0_ERA								TRNG0_INTG_OPT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRNG0_ECO_REV								TRNG0_CONFIG_OPT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.1.3.3.42.4 Fields

Field	Function
31-24 TRNG0_ERA	Shows the Freescale compile options for the TRNG. 00000000 - COMPILE_OPT for TRNG.
23-16 TRNG0_INTG_OPT	Shows the Freescale integration options for the TRNG. 00000000 - INTG_OPT for TRNG.
15-8 TRNG0_ECO_REV	Shows the Freescale IP's ECO revision of the TRNG. 00000000 - TRNG_ECO_REV for TRNG.
7-0 TRNG0_CONFIG_OPT	Shows the Freescale IP's Configuaration options for the TRNG. 00000000 - TRNG_CONFIG_OPT for TRNG.

43.1.4 Another TRNG usage example.

The TRNG can be used by a post processing pseudo-random number generator function. For example, TRNG can be used to seed a hardware or software based implementation of a DRBG defined by SP800-90.

Chapter 44
2.4 GHz Radio

44.1 Introduction

The 2.4 GHz radio module provides the RF communication interface for the device. It converts and decodes between RF modulated signal domain and digital data domain.

Use of the radio will be facilitated through firmware provided by Freescale giving customers access to all the radio features through appropriate API interfaces. As such, detailed information on radio control registers is not provided here. Please see the relevant software documentation for radio control.

44.1.1 Features and supported protocols

The KW40 is a 2.4GHz multi-protocol device with a radio that is capable of supporting the following modes in 2.4GHz ISM band:

- IEEE Std. 802.15.1 v4.1 Bluetooth Low Energy(BLE) single-mode
- IEEE Std. 802.15.4-2011 compliant ZigBee RF4CE/IP
- IEEE Std. 802.15.4j-2012 in Medical Body Area Network (MBAN) frequency bands spanning from 2360 MHz to 2400 MHz

The radio comprises of a constant-envelope transmit and a quadrature zero-IF receiver with built-in Voltage Controlled Oscillator (VCO), Fractional Division Phase Lock Loop (Frac-N PLL) and Power Amplifier (PA) circuitry to provide complete RF communication in 2.4GHz ISM band.

44.1.2 Block Diagram

The following is the block diagram of the 2.4 GHz Radio

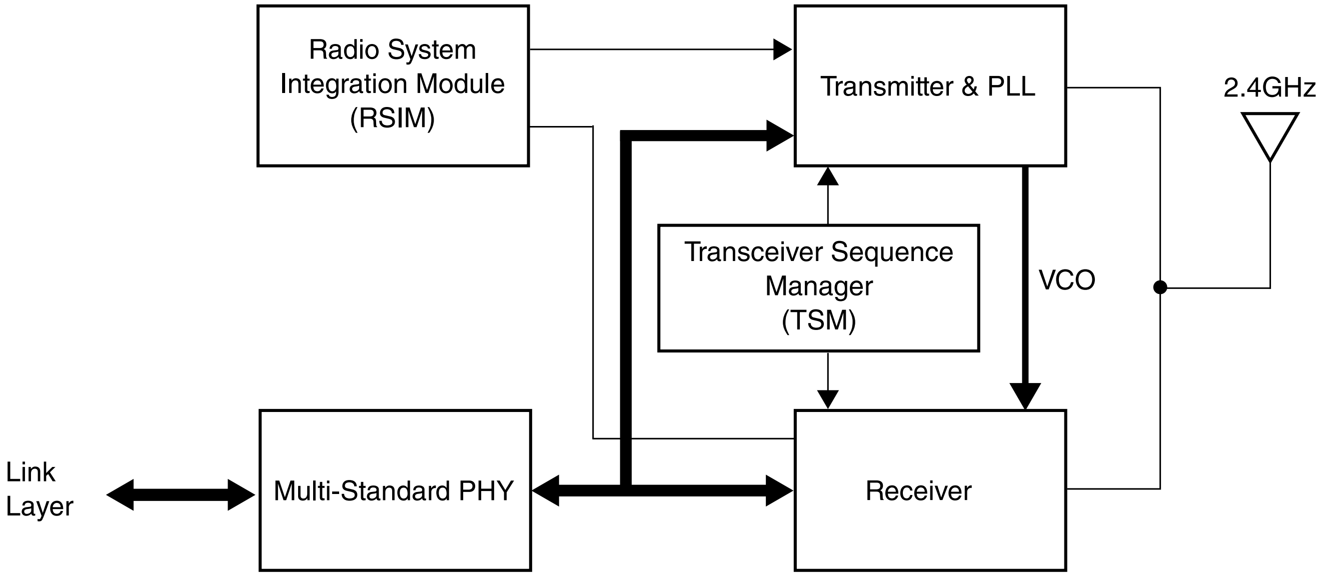


Figure 44-1. 2.4 GHz Radio Block Diagram

44.2 Radio Configuration

The Radio is a module which contains the various blocks responsible for overall radio operation and control. It includes:

A Transceiver, which consists of -

- RX digital block
- TX digital block
- PLL digital block
- Zigbee demod block
- Transceiver Sequence Manager (TSM) block

A Bluetooth Low Energy Link Layer

An 802.15.4 Link Layer

The various radio blocks are interconnected as shown below:

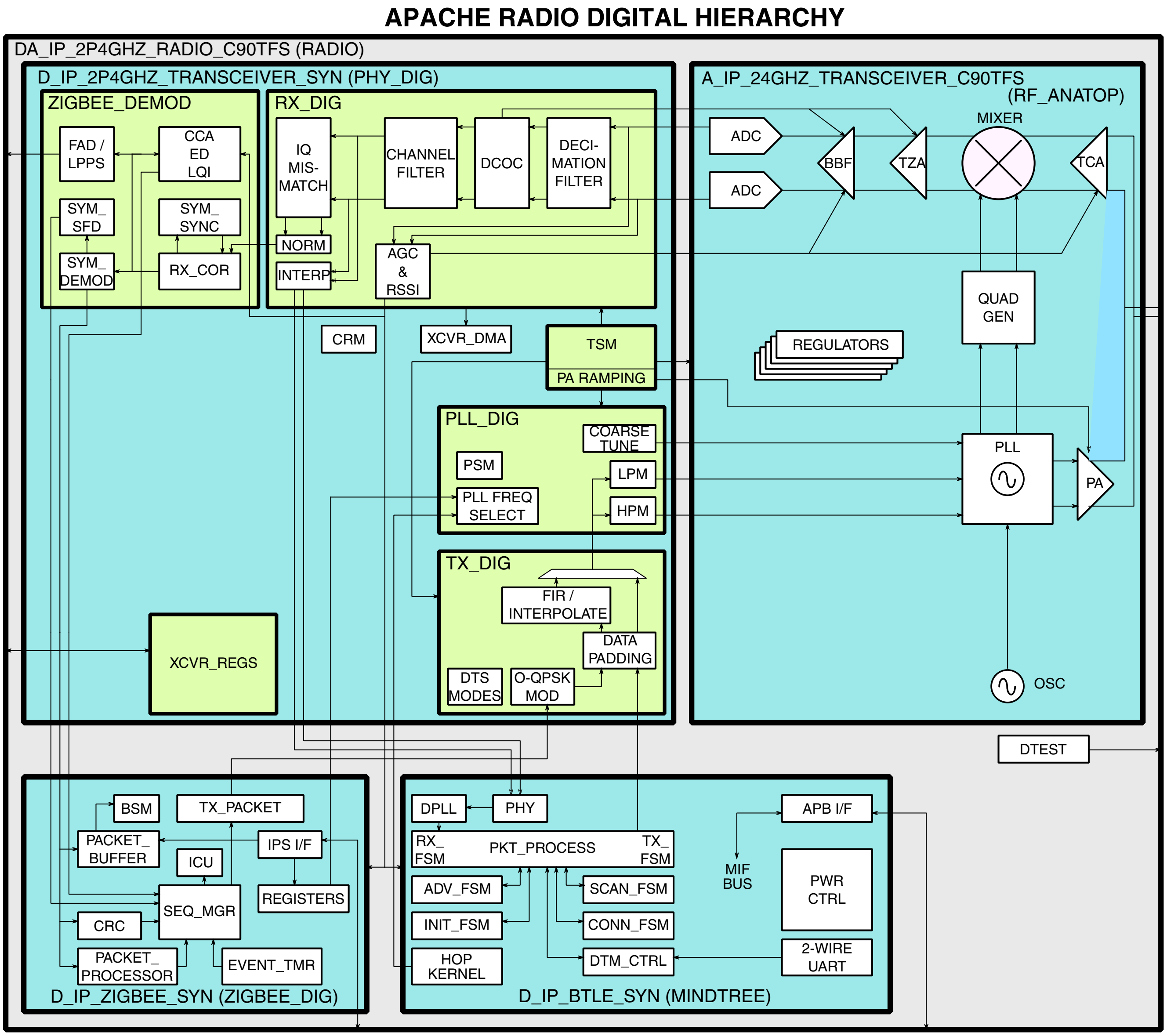


Figure 44-2. Radio Digital Block Diagram

44.3 Functional Blocks

The 2.4GHz radio module contains the following functional blocks.

44.3.1 Transceiver Sequence Manager (TSM)

The Transceiver Sequence Manager (TSM) controls the warm-up and warm-down processes for all radio sequences. TSM provides a simplified interface to control the timing event of individual transceiver blocks. For TX and RX, warm-up and warm-down sequences are provided by the TSM.

44.3.2 Radio System Integration Module (RSIM)

The Radio System Integration Module (RSIM) provides top-level integration support for the radio. It provides radio reset generation, clock enable control, and radio analog circuitry trim.

44.3.3 Transmitter and PLL

The transmitter and PLL support constant envelope modulation. It comprises a VCO, Fractional-N PLL, and Power Amplifier. The reference oscillator of the Fractional-N PLL is the 32 MHz crystal oscillator.

44.3.4 Receiver

The receiver architecture is Zero IF where the received signal passes through the RF front end and is down converted to baseband. The baseband signal is filtered and amplified before it is fed to an analog-to-digital converter (ADC). The digital signal is then digitally processed, demodulated, and passed on to packet processing.

Automatic gain control (AGC) is provided in the receiver line to avoid over-driving the ADC.

The DC Offset Correction (DCOC) circuit is included in the receiver to ensure minimum DC offset, which will improve the receiver dynamic range. The correction is fully automatic.

44.3.5 Digital Radio Processing

Digital hardware is provided to demodulate, detect preambles, synchronize, correct for frequency offsets, process the packets and perform link layer acceleration. Energy detection, CCA and LQI functions are also handled by the digital radio.

44.4 Register Definition

The register information of the radio can be found in the following chapters:

- RF Transceiver Register Definition
- BLE RF Register Definition
- 802.15.4 Register Definition

Chapter 45
Clock and Reset Module

45.1 About this module

45.1.1 Introduction

The Clocks and Resets module is the source for all the Radio clocks and Resets.

45.1.2 Features

Support for data, symbol and chip clocks for various RF Protocols:

- Bluetooth Low Energy
- IEEE 802.15.4

Clock gating for unused digital logic

45.1.3 Modes and Operations

The Clocks and Resets module supports the modes and operations described in the indicated sections:

- Operating Modes
- Low Power Modes
- Debug Mode
- Operations

45.1.4 Block diagram

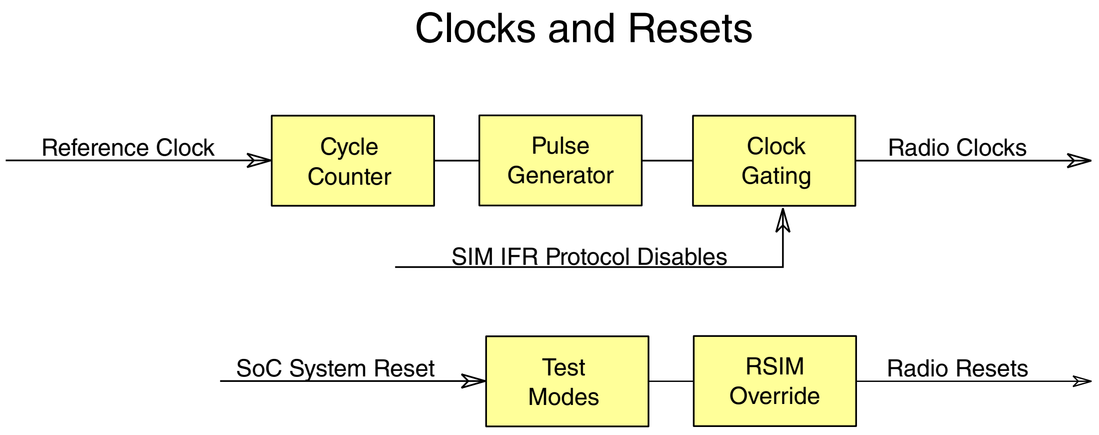


Figure 45-1. Block diagram

45.2 Signals

This table describes the signals on the boundary of the Clocks and Resets module.

Signal	I/O	Description
async_gasket_bypass	I	When asserted, ips_clk source is ipg_clk_rf_osc_bypass, otherwise ips_clk source is ref_clk
btle_1MHz_clk	I	The 1 MHz symbol clock used by Bluetooth Link Layer, this is used by the TX Digital as the data_clk source
dma_clk_en	I	When asserted, the ref_clk is un-gated and is sent to the xcvr_dma
ipg_clk_rf_osc_bypass	I	Bypass mode clock source for ips_clk
ipg_hard_async_reset_b	I	Radio digital domain asynchronous reset signal
ips_module_en_xcvr	I	IPS module enable for xcvr, un-gates the ips_clk to the PHY Digital when asserted
ips_module_en_zigbee	I	IPS module enable for zigbee, un-gates the ips_clk to the Zigbee Digital if sim_ifr_zigbee_disable is Not asserted
ipt_se_gatedclk	I	SoC gated clock enable in scan mode
osc_rf_clk_lv_gated	I	This is used for ips_clk in normal mode
protocol	I	Radio Protocol, 000 = BTLE, 001 = MBAN, 010 = BTLE/MBAN, 011 = ANT, 100 = Zigbee, 111 = Bypass
ref_clk	I	Reference Clock
ref_clk_freq	I	Reference Clock frequency, 00 = 32, 01 = 36, 10 = 40, 11 = 26 MHz

Table continues on the next page...

Signal	I/O	Description
scan_mode	I	SoC scan mode signal
sim_ifr_zigbee_disable	I	Disables all clocks to the Zigbee Digital when asserted
tsm_clk_en	I	Asynchronous signal from the TSM, enables clocks to TSM
tsm_clk_freq	I	Selects TSM frequency, 0 = 1 MHz, 1 = 500 kHz
tx_mode	I	From the TSM, when asserted this enables the TX Digital clock
ipg_clk_dma	O	Clock for DMA, test mode only
ips_clk_xcvr	O	IPS clock for PHY Digital
ips_clk_zigbee	O	IPS clock for Zigbee Digital
ipg_clk_tsm	O	TSM clock, 1 MHz constant frequency
ipg_clk_tsm_dly	O	TSM clock, delayed by one ref_clk cycle
ipg_clk_tsm_4m	O	TSM clock, 4 MHz constant frequency, for PA ramp
chip_clock	O	TX Digital Chip clock for 802.15.4 protocols
data_clock	O	TX Digital Data clock, frequency selected by protocol {1, 2 MHz}
oversample_clock	O	TX Digital Oversample clock, frequency selected by protocol and ref_clk_freq {8, 9, 10, 13 MHz}
seq_mgr_clk	O	Sequence Manager clock for Zigbee Digital, synchronized with ipg_clk_tsm
cg125k_en	O	Clock Gate Enable, 125 kHz
cg250k_en	O	Clock Gate Enable, 250 kHz
cg500k_en	O	Clock Gate Enable, 500 kHz
cg1m_en	O	Clock Gate Enable, 1 MHz
cg2m_en	O	Clock Gate Enable, 2 MHz
cg4m_en	O	Clock Gate Enable, 4 MHz, only supported for 32, 36, and 40 MHz ref_clk
cg4m_b_en	O	Clock Gate Enable, 4 MHz Bar, only supported for 32, 36, and 40 MHz ref_clk
cg8m_en	O	Clock Gate Enable, 8 MHz, only supported for 32MHz ref_clk
cg16m_en	O	Clock Gate Enable, 16 MHz, only supported for 32MHz ref_clk

45.3 Memory Map and Register Definition

The Clocks and Reset module memory map and detailed descriptions of all its registers is included in the Transceiver Registers section.

The Clocks and Reset module registers are as follows:

- XCVR_XCVR_CTRL

45.4 Functional Description

The following sections describe functional details of the Clocks and Resets module.

45.4.1 Operating Modes

The Clocks and Resets module is the source of the Radio clocks and resets during Radio Transmit and Radio Receive operations.

45.4.2 Low Power Modes

The Clocks and Resets module is the source of the Radio clock gating during SoC run modes.

In SoC low power modes all of the CRM clocks should be gated off (but this hasn't been implemented yet)

45.4.3 Debug Mode

The Clocks and Resets module does not contain any debug features.

45.4.4 Clocks

The Clocks and Resets module has the following clock domain groups:

Reference Clock domain

- Generated by either the Radio Oscillator or an external source
- Used to derive all the other clocks, except for the Bluetooth 1 MHz symbol clock

TSM Clock domains

- 1 MHz
- 4 MHz

TX Clock domains

- Data clock, 1 or 2 MHz
- Over Sample Clock, 8, 9, 10, or 13 MHz

RX Clock domains

- Same as reference clock

DMA Clock domain
- Same as reference clock

These domains are configured based on the REF_CLK_FREQ and PROTOCOL registers.

45.4.5 Reset

The Clocks and Resets module normally uses the SoC system reset signal as the Radio reset source. However, in test mode the Radio reset source can be a register bit in the RSIM module.

45.4.6 Interrupts

The Clocks and Reset module does not generate any interrupts.

Chapter 46
Radio System Integration Module (RSIM)

46.1 About this module

46.1.1 Introduction

The Radio System Integration Module (RSIM) provides support for the Radio specific System Integration requirements.

46.1.2 Features

The RSIM includes the following features:

- Status bit and enable of the BLE Ref Osc (Sysclk) Request
- Interrupt enable and flag for BLE Ref Osc (Sysclk) Request
- Power Mode specific enables for the RF Ref Osc
- Status bit and override control of the RF Ref Osc Ready signal
- Control of the SoC Platform Asynchronous Gasket Bypass
- Override control of the Stop Acknowledge signal from the Radio to the SoC
- Control bit to force BLE to exit deep sleep mode
- Control of the delay time for the BLE Active signal to the Flash
- Output enables for the two pad outputs of the BLE Active signal
- A 40-bit unique MAC address loaded from the SoC Flash IFR at Power On Reset
- SoC padring isolation controls for sensitive RF analog circuits
- Radio Reset blocking and a software Radio Reset control bit to support SoC and Radio concurrent testing

46.1.3 Modes and Operations

The RSIM is always powered on, and is available for register reads and writes in any SoC mode where such are allowed.

The register values in the RSIM are powered by the 3v power domain, and will be maintained in all SoC low power modes except for a complete power down.

Any voltage isolation that is needed by the Radio to separate the 3v power domain from the 1.2v power domain is done in the RSIM.

In SoC test modes it is possible to isolate the Radio from the SoC to allow for concurrent testing of the two systems.

46.1.4 Radio Registers Access (CAUTION)

WARNING: If the Radio Register Clocks are not configured correctly before a Memory Access the SoC Core will halt. The descriptions below and the RSIM Register descriptions are intended to help avoid such a result.

46.1.4.1 BLE Link Layer Registers Access

The Radio Registers for the BLE Link Layer are accessed through an SoC Asynchronous Gasket in the Core Platform, and can only be accessed when the RF Ref Osc is Enabled and Ready, this is because the BLE Link Layer requires an accurate time-base clock for its time-keeping functions.

The BLE link layer will always exit reset by first Requesting the RF Ref Osc (Sysclk Req). By default, the BLE_RF_OSC_REQ_EN bit blocks that behavior to allow software to first configure the Radio and then enable the RF Ref Osc request. The status of the BLE RF Ref Osc request can be monitored using the BLE_RF_OSC_REQ_STAT bit.

Software can enable the BLE RF Ref Osc request to activate the RF Ref Osc, or it can use the RF_OSC_EN register to force the RF Ref Osc on. After the RF Ref Osc is enabled, software can monitor the RF_OSC_READY bit to determine when the RF Ref Osc is Enabled and Ready

WARNING: If the RF Ref Osc is not Enabled and Ready, and a BLE Link Layer Memory Access is attempted, then the SoC Asynchronous Gasket will block the SoC Core from proceeding as it waits for the Radio to send it the RF Ref Osc.

46.1.4.2 XCVR and Zigbee Registers Access

The Radio Registers for the XCVR and Zigbee are accessed through an SoC Asynchronous Gasket in the Core Platform, and they can be accessed with either the RF Ref Osc (Not Bypassed) or the SoC IPG clock (Bypassed).

The default behaviour of this Gasket is Not Bypassed, which means the Radio registers are accessed using the RF Ref Osc clock.

When the Gasket is Not Bypassed, the Radio sends the RF Ref Osc to the asynchronous gasket which then uses that clock for SoC register accesses of the Radio registers. This requires the RF Ref Osc to be Enabled and Ready.

If the RF Ref Osc is not Enabled and Ready, then the IPS Gasket can be overridden to be in Bypass Mode, which forces the use of the SoC IPG clock as the register access clock.

If the RF Ref Osc IS Enabled and Ready, then the IPS Gasket can be overridden to be in Bypass Mode, but this is an ILLEGAL ACCESS if the Radio is operational, DO NOT write Radio registers with the Asynchronous Gasket Bypassed when the Radio is Operating. Test mode access is allowed but glitches may occur.

WARNING: If the RF Ref Osc is not Enabled and Ready, and a XCVR or Zigbee Memory Access is attempted, then the SoC Asynchronous Gasket will block the SoC Core from proceeding as it waits for the Radio to send it the RF Ref Osc.

46.1.4.3 Asynchronous Gasket Bypass

The intent of Bypass Mode is to allow software to configure the Radio XCVR and Zigbee registers before the RF Ref Osc is Enabled and Ready.

The following table shows which clock is being used to access the Radio XCVR and Zigbee registers.

Gasket Bypass Override Enable	Gasket Bypass Override	XCVR and Zigbee Register Clock
1	0	RF Ref Osc Clock
1	1	SoC IPG Clock
0	x	RF Ref Osc Clock

More details are available in the Radio Clocks diagram and in the Gasket Bypass Registers descriptions.

46.1.5 Interrupts

The RSIM module can generate two asynchronous interrupts:

- A 3v interrupt that occurs on a rising edge of BLE Ref Osc (Sysclk) Request
- A 1.2v interrupt that occurs on a rising edge of BLE Ref Osc (Sysclk) Request, or when the BLE Link Layer generates an interrupt

The 3v interrupt is intended as an SoC Low Voltage Power Mode wakeup source.

The 1.2v interrupt is intended as an asynchronous interrupt for the SoC core platform.

Both Interrupts are blocked if BLOCK_RADIO_OUTPUTS is set.

46.1.6 Block diagram

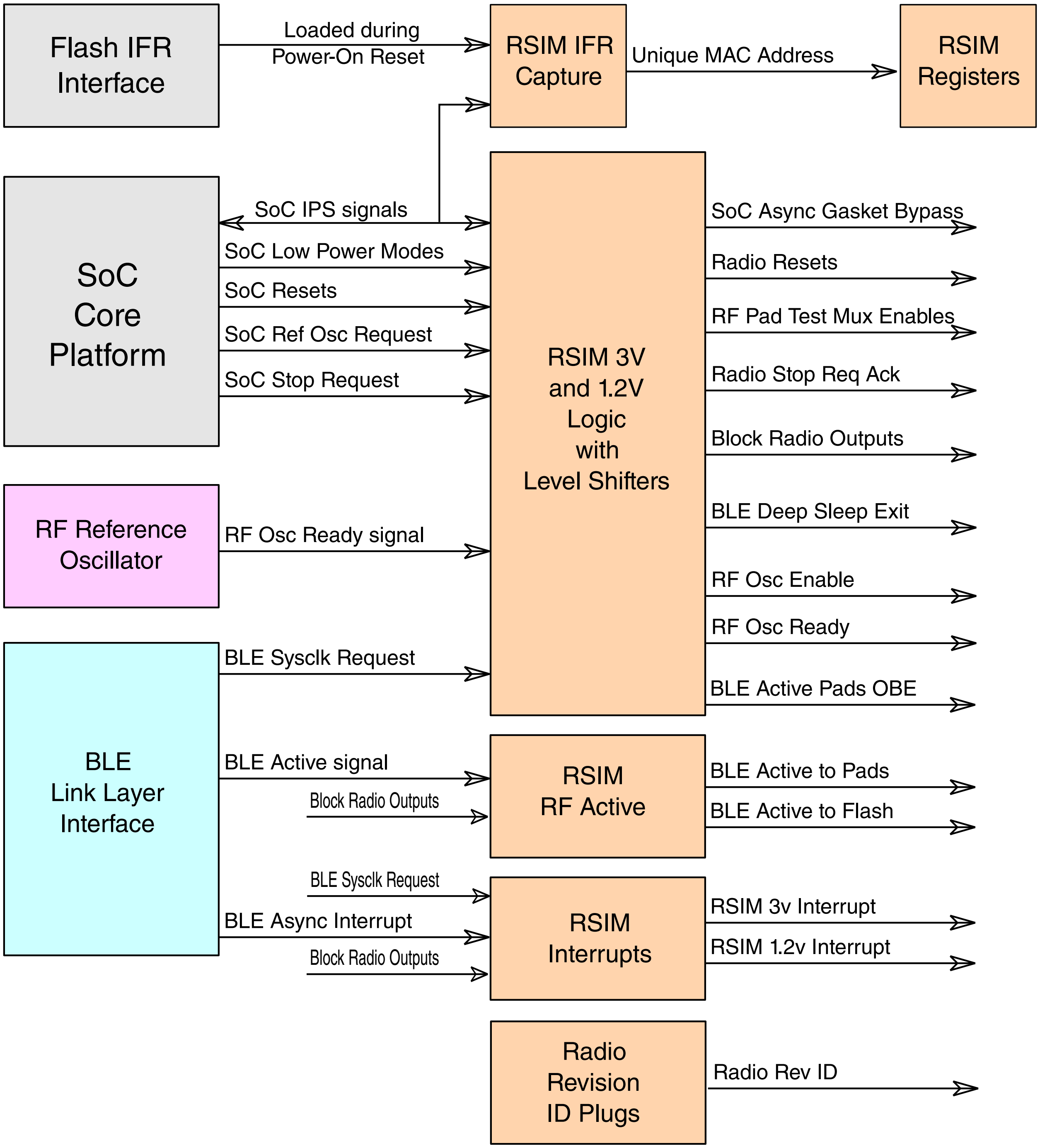


Figure 46-1. Radio System Integration Module

46.1.7 Radio Clocks diagram

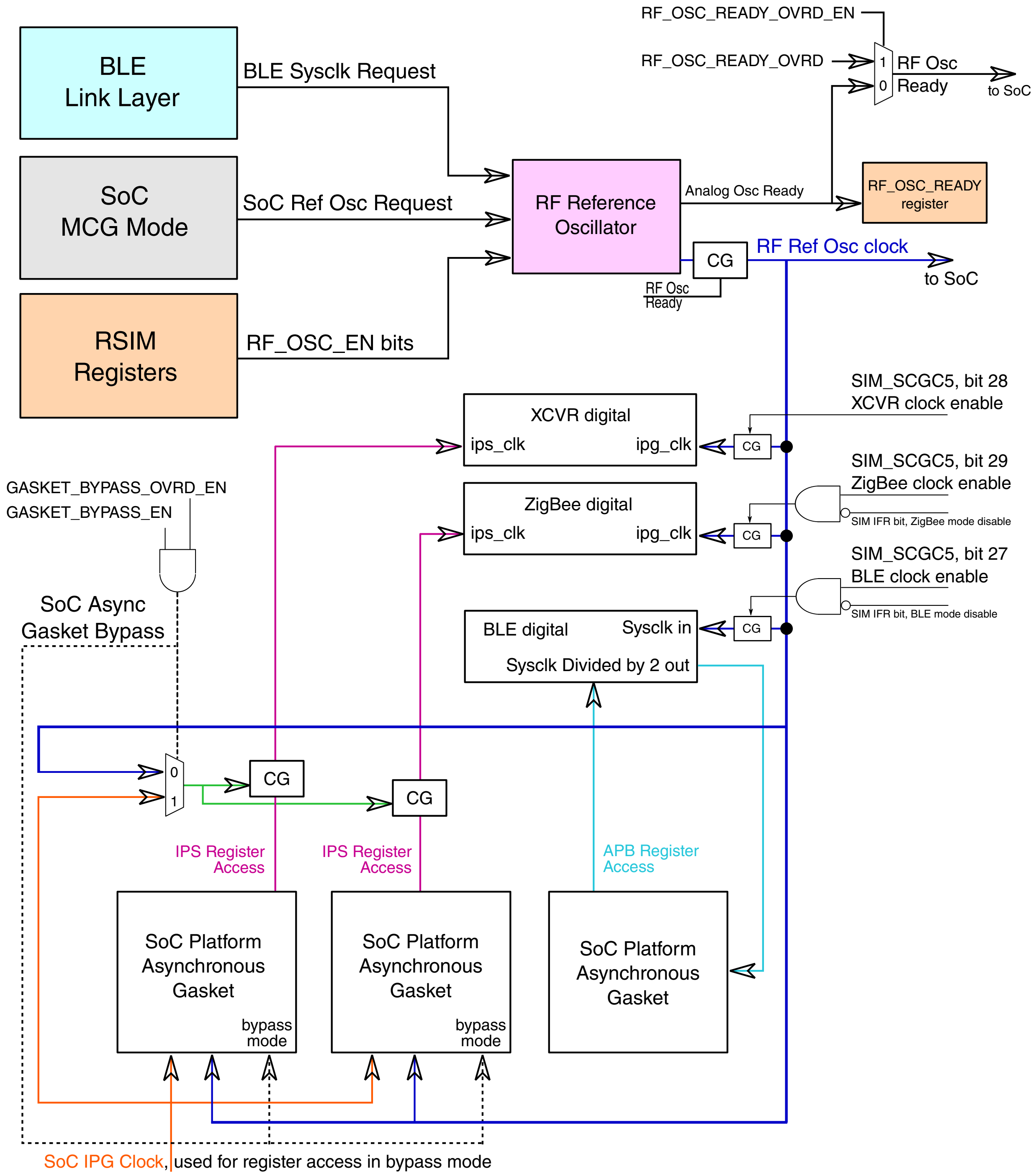


Figure 46-2. Radio Clocking Configuration

46.2 Memory Map and Register Definition

Radio System Integration Module

RSIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_9000	RSIM Control (RSIM_CONTROL)	32	R/W	See section	46.2.1/982
4005_9004	RSIM BLE Active Delay (RSIM_ACTIVE_DELAY)	32	R/W	0000_0000h	46.2.2/987
4005_9008	RSIM MAC MSB (RSIM_MAC_MSB)	32	R	See section	46.2.3/988
4005_900C	RSIM MAC LSB (RSIM_MAC_LSB)	32	R	See section	46.2.4/989
4005_9010	RSIM Analog Test (RSIM_ANA_TEST)	32	R/W	0100_0000h	46.2.5/989

46.2.1 RSIM Control (RSIM_CONTROL)

The RSIM Control register provides various control bits for the Radio System and for its interaction with the SoC Systems.

Address: 4005_9000h base + 0h offset = 4005_9000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RADIO_RESET	0	BLOCK_RADIO_OUTPUTS	BLOCK_RADIO_RESETS	0	RF_OSC_READY_OVRD	RF_OSC_READY_OVRD_EN	RF_OSC_READY	STOP_ACK_OVRD	STOP_ACK_OVRD_EN	0	BLE_DEEP_SLEEP_EXIT	0		BLE_ACTIVE_PORT_2_SEL	BLE_ACTIVE_PORT_1_SEL
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RF_OSC_BYPASS_EN	GASKET_BYPASS_OVRD	GASKET_BYPASS_OVRD_EN	RF_OSC_EN				0		BLE_RF_OSC_REQ_INT	BLE_RF_OSC_REQ_INT_EN	0		BLE_RF_OSC_REQ_STAT	BLE_RF_OSC_REQ_EN
W											w1c					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSIM_CONTROL field descriptions

Field	Description
31 RADIO_RESET	Software Reset for the Radio This bit resets on POR only. When the Radio Resets are Blocked, setting this bit will reset all the radio logic until this bit is cleared. Note that due to internal Radio Reset Exit synchronizing logic there must be a second access to an RSIM register to clear this software reset, so please write this bit to 0 twice when clearing it.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 BLOCK_RADIO_OUTPUTS	Block Radio Outputs This bit resets on POR only. This bit is intended to allow the SoC to perform concurrent testing of various SoC logic while the Radio is operating independently. Any Radio output signals that go to the SoC will be blocked so as to not affect the SoC testing when this bit is set.
28 BLOCK_RADIO_RESETS	Block Radio Resets This bit resets on POR only.

Table continues on the next page...

RSIM_CONTROL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>This bit is intended to allow the SoC to perform concurrent testing of various SoC logic while the Radio is operating independently. Any SoC resets will be blocked and the Radio will not be affected by them when this bit is set.</td></tr>
<tr><td>27
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>26
RF_OSC_
READY_OVRD</td><td>RF Ref Osc Ready Override

This bit directly controls the Radio RF Ref Osc Ready signal when the RF Ref Osc Ready Override is enabled.

All Radio and SoC signals that are derived from the RF Ref Osc Ready signal can be overridden using this bit.</td></tr>
<tr><td>25
RF_OSC_
READY_OVRD_
EN</td><td>RF Ref Osc Ready Override Enable

This bit enables the RF Ref Osc Ready Override bit.</td></tr>
<tr><td>24
RF_OSC_
READY</td><td>RF Ref Osc Ready

The RF Reference Oscillator has an internal counter that gates off the RF Ref Osc clock until the selected count is reached. This bit shows the status of the RF Ref Osc ready signal that comes from that counter, except in RF Ref Osc Bypass Mode.

In RF Ref Osc Bypass Mode this bit will always be asserted, but the signals that are derived from the RF Ref Osc Ready signal can be overridden using the RF_OSC_READY_OVRD bit</td></tr>
<tr><td>23
STOP_ACK_
OVRD</td><td>Stop Acknowledge Override

This bit controls the Stop Acknowledge signal to the SoC Core Platform in Override mode.</td></tr>
<tr><td>22
STOP_ACK_
OVRD_EN</td><td>Stop Acknowledge Override Enable

This bit enables an override of the Stop Acknowledge signal.

If not overwritten, Radio Stop Acknowledge is nominally based on the enabled version of the BLE Ref Osc (Sysclk) Request.

The following table shows the state of the nominal Radio Stop Acknowledge signal as presented to the SoC Core Platform
<table>
<tr><th>BLE Ref Osc (Sysclk) Request Enable</th><th>BLE Ref Osc (Sysclk) Request</th><th>Radio Stop Acknowledge to SoC</th></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>x</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>21
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>20
BLE_DEEP_
SLEEP_EXIT</td><td>BLE Deep Sleep Exit

This bit forces the BLE link layer to wakeup from Deep Sleep Mode.</td></tr>
<tr><td>19–18
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
</table>

Table continues on the next page...

RSIM_CONTROL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>17
BLE_ACTIVE_
PORT_2_SEL</td><td>BLE Active port 2 select
This bit enables the Output Driver (OBE) on the SoC port 2 that provides the BLE Active signal as a pad interface option.</td></tr>
<tr><td>16
BLE_ACTIVE_
PORT_1_SEL</td><td>BLE Active port 1 select
This bit enables the Output Driver (OBE) on the SoC port 1 that provides the BLE Active signal as a pad interface option.</td></tr>
<tr><td>15
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>14
RF_OSC_
BYPASS_EN</td><td>RF Ref Osc Bypass Enable
This bit engages the RF Ref Osc analog bypass circuit if the RF Ref Osc is enabled. When the RF Ref Osc is in bypass mode it passes the EXTAL clock as the RF Ref Osc clock.
Note that the RF Ref Osc Ready signal is always asserted in RF Ref Osc Bypass mode, unless overridden with the RF_OSC_READY_OVRD_EN bit.</td></tr>
<tr><td>13
GASKET_
BYPASS_OVRD</td><td>Gasket Bypass Override
This bit directly controls the SoC platform asynchronous gasket bypass signal when the Gasket Bypass Override is enabled.
The default behavior of the SoC Asynchronous Gasket is Not Bypassed, which means the Radio registers are accessed using the RF Ref Osc clock. The Radio sends the RF Ref Osc to the asynchronous gasket which then uses that clock for SoC register accesses of the Radio registers. This requires the RF Ref Osc to be Enabled and Ready.
If the RF Ref Osc is not Enabled and Ready, then the IPS Gasket can be overridden to be in Bypass Mode, which forces the use of the SoC IPG clock as the register access clock.
If the RF Ref Osc IS Enabled and Ready, then the IPS Gasket can be overridden to be in Bypass Mode, but this is an ILLEGALL ACCESS if the Radio is operational, DO NOT write Radio registers with the Asynchronous Gasket Bypassed when the Radio is Operating. Test mode access is allowed but glitches may occur.
The intent of Bypass Mode is to allow software to configure the Radio before the RF Ref Osc is Enabled and Ready.
Note that the BLE Link Layer registers can only be accessed when the RF Ref Osc is Enabled and Ready.
The following table shows which clock is being used to access the Radio XCVR and Zigbee registers.
<table>
<tr><th>Gasket Bypass Override Enable</th><th>Gasket Bypass Override</th><th>XCVR and Zigbee Register Clock</th></tr>
<tr><td>1</td><td>0</td><td>RF Ref Osc Clock</td></tr>
<tr><td>1</td><td>1</td><td>SoC IPG Clock</td></tr>
<tr><td>0</td><td>x</td><td>RF Ref Osc Clock</td></tr>
</table></td></tr>
<tr><td>12
GASKET_
BYPASS_
OVRD_EN</td><td>Gasket Bypass Override Enable
The SoC platform has an asynchronous gasket that allows register access for the Radio registers in all SoC clocking modes.
This bit allows software to directly control the SoC platform asynchronous gasket bypass signal.</td></tr>
<tr><td>11–8
RF_OSC_EN</td><td>RF Ref Osc Enable [3:0]</td></tr>
</table>

Table continues on the next page...

RSIM_CONTROL field descriptions (continued)

Field	Description
	The RF Reference Oscillator can be enabled by the BLE link layer, by an SoC MCG mode, or by these bits. If these bits are all cleared, 0000, then the RF Ref Osc will be controlled by the SoC or the BLE link layer. If any of these bits are set then the RF Ref Osc will be on in the SoC power modes as shown below. Note that the enables are additive; each bit adds another low power mode. 0000 RF Ref Osc will be controlled by the SoC or the BLE link layer 0001 RF Ref Osc on in Run/Wait 0011 RF Ref Osc on in Stop 0111 RF Ref Osc on in VLPR/VLPW 1111 RF Ref Osc on in VLPS
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 BLE_RF_OSC_ REQ_INT	BLE Ref Osc (Sysclk) Request Interrupt Flag This bit is an interrupt flag that is set when the enabled version of the BLE Ref Osc (Sysclk) Request is asserted high. This interrupt flag is cleared by writing a 1 to it.
4 BLE_RF_OSC_ REQ_INT_EN	BLE Ref Osc (Sysclk) Request Interrupt Enable This bit enables an interrupt request when the enabled version of the BLE Ref Osc (Sysclk) Request is asserted high.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 BLE_RF_OSC_ REQ_STAT	BLE Ref Osc (Sysclk) Request Status This bit indicates the current status of the BLE link layer request to turn on the RF Ref Oscillator (Sysclk Req).
0 BLE_RF_OSC_ REQ_EN	BLE Ref Osc (Sysclk) Request Enable This bit resets on POR only. If this bit is cleared (the default state), then all BLE link layer requests to turn on the RF Ref Oscillator (Sysclk Req) will be blocked and ignored. In BLE protocols the BLE link layer will always restart when exiting reset by first Requesting the RF Ref Osc (Sysclk Req), this bit blocks that behavior until software configures the Radio and enables the requests.

46.2.2 RSIM BLE Active Delay (RSIM_ACTIVE_DELAY)

The RSIM BLE Active Delay register provides control bits to adjust the delay of the BLE Active signal that is presented to the SoC Flash System.

Address: 4005_9000h base + 4h offset = 4005_9004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												BLE_ ACTIVE_ COARSE_ DELAY				0										BLE_ACTIVE_FINE_ DELAY					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSIM_ACTIVE_DELAY field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 BLE_ACTIVE_ COARSE_ DELAY	The SoC Flash is presented with a BLE Active early warning signal to allow the Flash to complete any program or erase activities prior to a Radio communication event. This warning signal is delayed from the BLE Active signal provided by the BLE link layer. The timing of the Flash delay is calculated as follows: BLE Active link layer delay - (BLE Active Flash Fine Delay x 32 kHz clock period x 4) - (BLE Active Flash Coarse Delay x 32 kHz clock period x 64)
15–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BLE_ACTIVE_ FINE_DELAY	The SoC Flash is presented with a BLE Active early warning signal to allow the Flash to complete any program or erase activities prior to a Radio communication event. This warning signal is delayed from the BLE Active signal provided by the BLE link layer. The amount of the delay from the BLE link layer is calculated as follows: BLE Active link layer delay - (BLE Active Flash Fine Delay x 32 kHz clock period x 4) - (BLE Active Flash Coarse Delay x 32 kHz clock period x 64)

46.2.3 RSIM MAC MSB (RSIM_MAC_MSB)

The RSIM MAC Address registers provide a unique ID that is stored in the Flash during factory test

Address: 4005_9000h base + 8h offset = 4005_9008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								MAC_ADDR_MSB							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:

- u = Unaffected by reset.

RSIM_MAC_MSB field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MAC_ADDR_ MSB	MAC Address MSB The RSIM MAC Address is loaded from the Flash IFR during the SoC Power on Reset sequence. The MAC Address is a unique ID that is stored in the Flash during factory test.

46.2.4 RSIM MAC LSB (RSIM_MAC_LSB)

The RSIM MAC Address registers provide a unique ID that is stored in the Flash during factory test

Address: 4005_9000h base + Ch offset = 4005_900Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAC_ADDR_LSB																															
W																																
Reset	0	0																														
POR	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

RSIM_MAC_LSB field descriptions

Field	Description
MAC_ADDR_ LSB	MAC Address LSB The RSIM MAC Address is loaded from the Flash IFR during the SoC Power on Reset sequence. The MAC Address is a unique ID that is stored in the Flash during factory test.

46.2.5 RSIM Analog Test (RSIM_ANA_TEST)

The RSIM Analog Test register provides controls for validation and factory test of the RF Analog Circuits.

Address: 4005_9000h base + 10h offset = 4005_9010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0																			ATST_GATE_EN				
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSIM_ANA_TEST field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

RSIM_ANA_TEST field descriptions (continued)

Field	Description
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ATST_GATE_EN	ATST Transmission Gate Enables The RSIM Analog Transmission Gate Enables open up the transmissions gates in the pads to allow testing of the Radio analog signals. Each bit opens up one Analog Transmission gate in the padring.

Chapter 47
PLL Digital

47.1 About this module

47.1.1 Introduction

The PLL Digital is a module that -

1. Coarse Tunes and Calibrates the VCO, for every Radio operation.
2. Sets the Radio Carrier Frequency, for every Radio operation.
3. Takes the Baseband Frequency Word supplied by the TX Digital module and applies it as High Port Modulation (HPM) to the VCO High Port capacitor array, and as Low Port Modulation (LPM) to the PLL Loop Divider, for Transmit operations only.
4. Monitors the PLL locked state, for every Radio operations.

Both the HPM and LPM use Sigma Delta Modulators (SDM) clocked by the PLL Loop Divider output clock, which is twice the reference clock frequency.

The PLL Digital State Machine performs a Coarse Tune frequency selection on the VCO Coarse Tune capacitor array, and then calculates a calibration factor for the VCO High Port capacitor array to adjust for variations in the VCO modulation frequency gain (K_{mod}).

A PLL Frequency Selector module is included to map the 128 available Radio Channels to the various RF protocols channel numbers.

47.1.2 Features

The PLL Digital module includes the following features:

Support for Radio Carrier Frequency selection by RF Protocols:

- Bluetooth Low Energy (BLE)
- Zigbee

Support for PLL Loss of Lock detection:

- Coarse Tuning frequency failure
- Fine Tuning frequency failure
- PLL Cycle Slipping failure

47.1.3 PLL Digital Block diagram

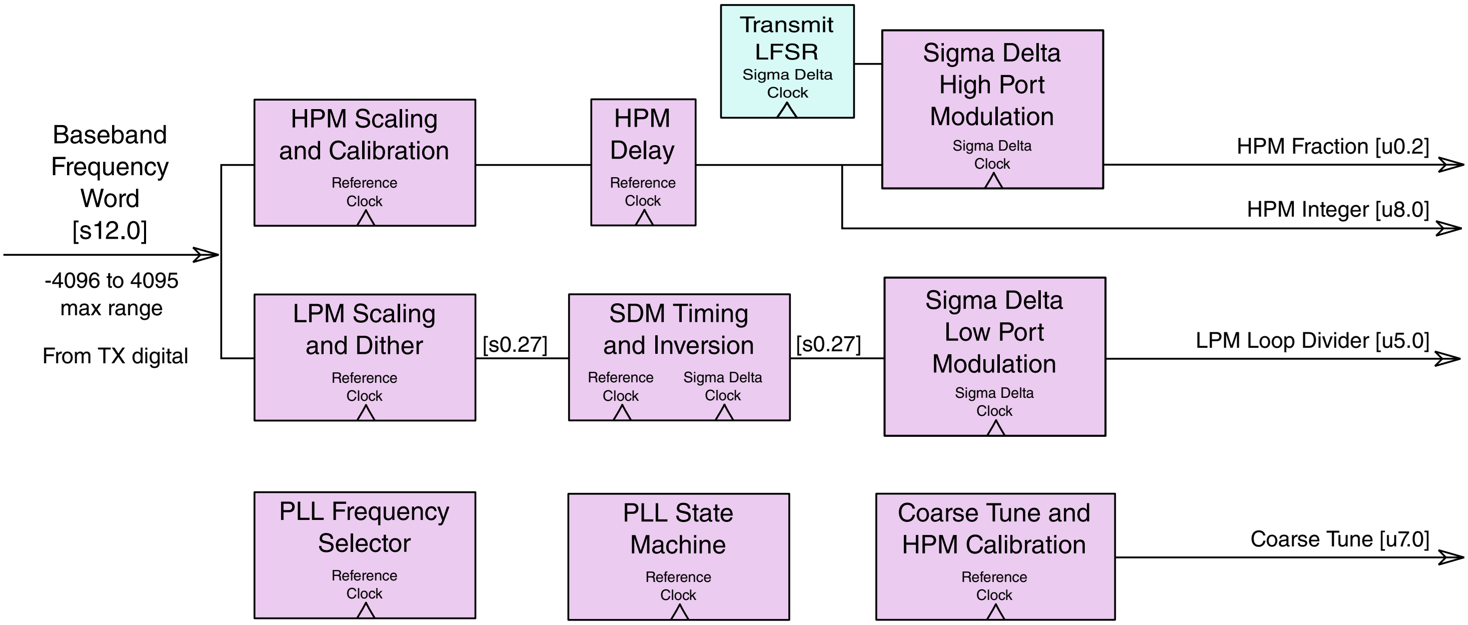


Figure 47-1. PLL Digital

47.2 Functional Description

The following sections describe functional details of the PLL Digital module.

47.2.1 VCO Calibration

The PLL analog has a 17-bit Ripple Counter that counts the number of VCO clock cycles whenever the counter is engaged.

The PLL digital has a State Machine that controls the analog PLL Ripple Counter, and the resulting count values are used to Coarse Tune the VCO before every Radio operation.

If the Radio operation is to be a transmission, then the VCO High Port capacitor array is calibrated to adjust for variations in the VCO modulation frequency gain (K_{mod}); the PLL Ripple Counter is used to do this calibration as described below.

47.2.1.1 Coarse Tune Frequency Calibration

The VCO coarse tune frequency calibration is carried out by a Successive Approximation Register (SAR) scheme that computes the best 7-bit Coarse Tune setting for the VCO Coarse Tune capacitor array in successive steps from MSB to LSB.

The PLL 17-bit ripple counter counts during 1 us timing windows and the counter value is compared to expected counts (based upon the target radio frequency) to successively determine a coarse tune value for the VCO coarse tune array. There are 1 us settling times before each count, please see the TSM documentation for more details on the timing.

The best coarse tune count difference found during this SAR activity can be read as CTUNE_BEST_DIFF in the XCVR_PLL_CTUNE_RESULTS register.

47.2.1.2 High Port Modulation Calibration

The HPM calibration is done before each transmission to determine the adjustment needed for the variations in the VCO modulation frequency gain (K_{mod}).

The HPM calibration factor is determined by setting the HPM capacitor array to the minimum and maximum ranges and counting for a period of time at each range (25us or 50us, as selected by the HPM_CAL_TIME bit field). The resulting count difference is used in a lookup table to determine the HPM calibration factor.

The XCVR_PLL_HPM_CAL_CTRL register shows the current calibration factor being used, and a means to override it.

47.2.1.2.1 HPM K_{mod} Calibration Factor

The HPM calibration factor is calculated as follows:

- DIFF = (HPM_COUNT1-HPM_COUNT2)
- Delta_Frequency = DIFF /(accumulation time x 1,000,000)

- Kmod_LSB_Hz = Delta_Frequency / HPM array size
- SD_LSB/Kmod_LSB_Hz = (Reference_Freq/2^17) / Kmod_LSB_Hz

The above calculations are carried out without any actual multiplication or division using a lookup table.

47.2.2 Frequency Tuning

The Radio supports various RF protocols with carrier frequencies in the 2.360 to 2.487 GHz frequency range. The PLL Digital tunes the VCO to the carrier frequency required by the currently selected RF Protocol.

Each supported carrier frequency is derived by Fractional-N Frequency Synthesis using the Low Port Sigma Delta Modulator (LP SDM). The Integer and Fraction required for each supported carrier frequency are implemented in two lookup tables that include all the supported channels and all the supported reference clock frequencies.

The Protocol register selections are shown in the table below.

Table 47-1. Radio Frequency Protocol Selections

PROTOCOL Register Setting	Protocol Description
000	Bluetooth Low Energy (BLE)
001	BLE in Medical Body Area Network spectrum (MBAN)
010	BLE overlap MBAN spectrum
011	Reserved
100	Zigbee
101	802.15.4j
110	128 Channel FSK
111	128 Channel GFSK

The PLL Frequency Lookup Table supports 128 RF channels in 1 MHz increments spanning the range from 2.360 GHz to 2.487 GHz. These radio channels are mapped to the various supported RF protocols channel frequencies using the PLL Channels Lookup Table.

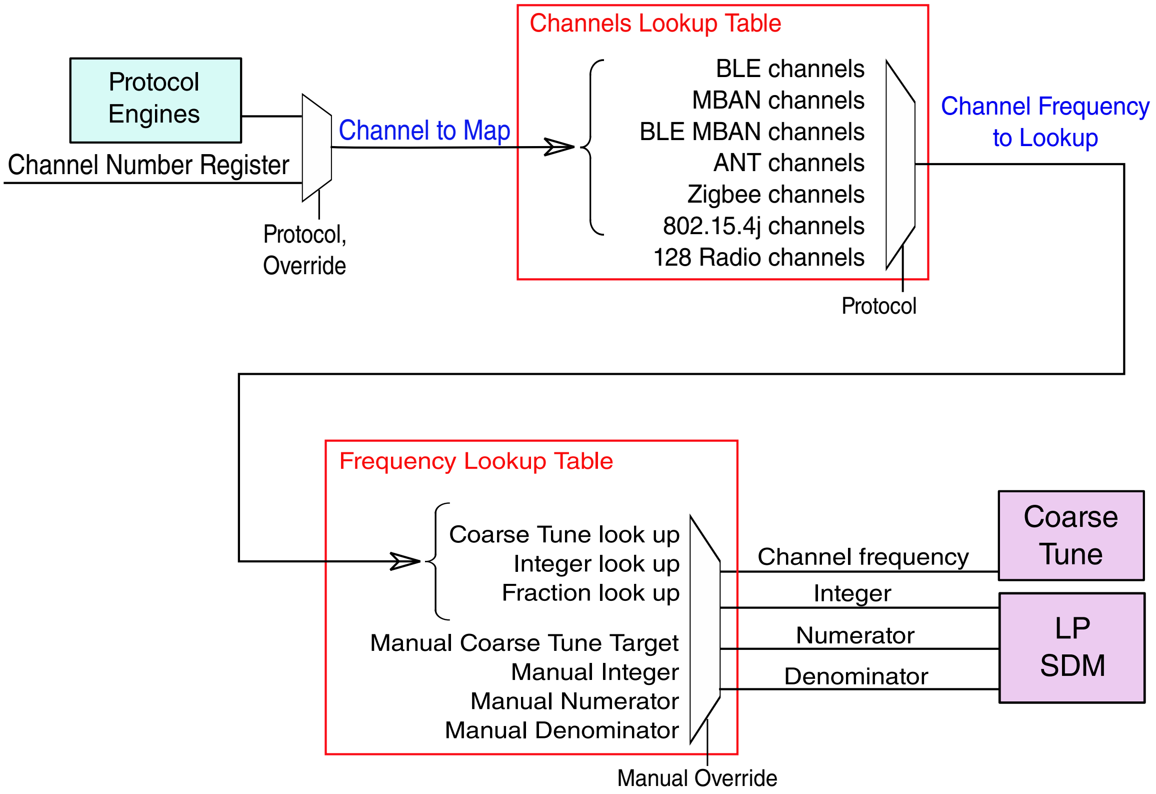


Figure 47-2. Radio Channel Frequency Selection

The carrier frequency is tuned to the RF Protocol channel frequency that is requested by the RF Protocol Engine.

For BLE protocols, the channel selection is done automatically by the BLE link layer so that no software or MCU activity is required for the radio to wake up and hop between BLE channels.

For other protocols, either their link layer needs to support channel number selection in order to hop, or the MCU must be used with software to select new channels.

The channel frequency is synthesized by directly applying the correct Integer, Numerator, and Denominator to the Low Port Sigma Delta Modulator.

47.2.2.1 Manual Control of Radio Channel

The RF Protocol Engine channel selection can be overridden using the XCVR_PLL_CHAN_MAP register.

In channel selection override modes (BOC and ZOC) the CHANNEL_NUM register can be used to directly select a Protocol specific Channel Number, which is then mapped internally to the correct Radio Carrier Frequency for PLL tuning.

The internal mapping of Protocol Channel Numbers to the Radio Carrier Frequency is detailed in the table below.

Table 47-2. Channel Number Mapping to Radio Carrier Frequency (MHz)

Protocol Channel Number	000 BLE	001 BLE in MBAN	010 BLE Overlap MBAN	100 Zigbee	101 802.15.4j
0	2402	2360	2402	2400	2363
1	2404	2361	2404	2400	2368
2	2406	2362	2406	2400	2373
3	2408	2363	2408	2400	2378
4	2410	2364	2410	2400	2383
5	2412	2365	2412	2400	2388
6	2414	2366	2414	2400	2393
7	2416	2367	2416	2400	2367
8	2418	2368	2418	2400	2372
9	2420	2369	2420	2400	2377
10	2422	2370	2422	2400	2382
11	2424	2371	2424	2405	2387
12	2426	2372	2426	2410	2392
13	2428	2373	2428	2415	2397
14	2430	2374	2430	2420	2395
15	2432	2375	2432	2425	2399
16	2434	2376	2434	2430	2399
17	2436	2377	2436	2435	2399
18	2438	2378	2438	2440	2399
19	2440	2379	2440	2445	2399
20	2442	2380	2442	2450	2399
21	2444	2381	2444	2455	2399
22	2446	2382	2446	2460	2399
23	2448	2383	2448	2465	2399
24	2450	2384	2450	2470	2399
25	2452	2385	2452	2475	2399
26	2454	2386	2454	2480	2399
27	2456	2387	2456	2400	2399
28	2458	2388	2458	2400	2399
29	2460	2389	2460	2400	2399
30	2462	2390	2390	2400	2399

Table continues on the next page...

Table 47-2. Channel Number Mapping to Radio Carrier Frequency (MHz) (continued)

Protocol Channel Number	000 BLE	001 BLE in MBAN	010 BLE Overlap MBAN	100 Zigbee	101 802.15.4j
31	2464	2391	2391	2400	2399
32	2466	2392	2392	2400	2399
33	2468	2393	2393	2400	2399
34	2470	2394	2394	2400	2399
35	2472	2395	2395	2400	2399
36	2474	2396	2396	2400	2399
37	2476	2397	2397	2400	2399
38	2478	2398	2398	2400	2399
39	2480	2399	2480 or 2399 *	2400	2399
Default	2400	2399	2399	2400	2399

* The BLE MBAN Channel Remap bit (BMR) controls the frequency mapping in this case.

47.2.2.2 Radio 128 Channels Protocols

The Radio supports two protocols with full access to the 2.360 to 2.487 GHz frequency range.

The Radio Carrier Frequency in these 128 Channel FSK and GFSK protocols can be calculated using the formula below:

Radio Carrier Frequency = (Channel Number + 2360) x 1 MHz

47.2.2.3 Manual Control of Low Port SDM

It is possible to bypass the internal frequency mapping completely and allow software to directly control the LP SDM inputs to select the channel frequency.

The Radio Channel Frequency override is manually selected by setting the SDM_MAP_DIS bit in the PLL_LP_SDM_CTRL1 register along with the LPM_INTG, LPM_NUM, and LPM_DENOM bit fields in the PLL_LP_SDM_CTRL1 and _CTRL2 registers.

Any modulation from the TX Digital will be added to this manual frequency selection.

The Manual carrier frequency selected can be calculated using the formula below:

Radio Carrier Frequency = ((Reference Clock Frequency x 2) x (LPM_INTG + (LPM_NUM / LPM_DENOM))

WARNING : The fraction (LPM_NUM / LPM_DENOM) must be in the range of -0.55 to +0.55 for valid Sigma Delta Modulator operation.

47.2.3 High Port Modulation

The PLL has a High Port Modulator that uses the PLL Baseband Frequency word to modulate the VCO High Port capacitor array during Transmit operations. During Receive operations there is no High Port Modulation and the High Port is held at its mid-point.

47.2.3.1 HPM Calibration and Multiplier

The VCO modulation frequency gain (K_{mod}), in Hz, is a function of the manufacturing process, and the current voltage, temperature and carrier frequency.

Therefore the value of the current VCO High Port capacitor array K_{mod} step size is calculated by doing a calibration before each PLL transmission.

The calibration result is applied to the High Port Modulation Multiplier along with the Baseband Frequency Word and the result is an integer value that is applied as a direct modulation of the VCO High Port capacitor array.

47.2.3.2 HPM Sigma Delta Modulator and Dither

A 6-bit fractional result from the multiplication is accumulated in the High Port Sigma Delta Modulator and is applied to the VCO High Port capacitor LSB array.

The 6-bit fractional result from the High Port Modulation Multiplier is also dithered by a Linear Feedback Shift Register before being processed by the HPM Sigma Delta modulator.

The HPM Sigma Delta modulator is a 2nd-order MASH Sigma Delta Modulator which converts the fractional result to an LSB resolution calculated as the nominal HPM_varactor_size / (2^6).

The resulting 3-state thermometric code is applied to the VCO High Port capacitor LSB array (HPM fraction [u2.0] in figure 1).

47.2.3.3 HPM Delay, Disable, and Inversion

The HPM frequency word has an array of programmable delays available to allow the HPM frequency changes to be matched to the LPM frequency changes and their delay through the PLL Loop Divider and onto the VCO feedback loop.

The High Port Modulation can be delayed at the direct modulation of the VCO High Port capacitor array (HPM_BANK_DELAY), and also at the VCO High Port capacitor LSB array (HPM_SDM_DELAY).

The High Port Modulation can be inverted at the HPM frequency word (HPM_MOD_INV), and also at the HP SDM (HPM_SDM_INV).

And the High Port Modulation can be disabled at the VCO High Port capacitor array (HPM integer [u8.0] in figure 1) using (HPM_BANK_DIS), and at the VCO High Port capacitor LSB array (HPM fraction [u2.0] in figure 1) using (HP_LSB_DIS).

47.2.4 Low Port Modulation

The PLL has a Low Port Modulator that controls the PLL Loop Divider (LPM Loop Divider [u5.0] in figure 1) in order to tune the VCO to the Radio carrier frequency, and in order to add any modulation from the Baseband Frequency Word.

47.2.4.1 Low Port Modulation Scaling

The Low Port Modulator applies a scaling factor to the Baseband Frequency Word that is received from the TX Digital module in order to match the requested frequency modulation range to the Low Port Modulator's Sigma Delta LSB frequency resolution.

47.2.4.2 Sigma Delta LSB Frequency Resolution

The table below shows the PLL LPM SDM Frequency Word LSB size (in Hz) for various Reference Frequencies

Reference Frequency (MHz)	32	36	40	26
PLL LSB (Hz)	244.14	274.66	305.18	198.36

47.2.4.3 LPM Sigma Delta Modulator and Dither

The LPM Sigma Delta Modulator is a 3rd-order Sigma Delta Modulator that places a fractional division in the feedback path of the PLL (Fractional-N Frequency Synthesis).

The input to this SDM is a combination of the Integer and Fraction required for tuning to the requested RF carrier frequency, plus the LPM scaled Baseband Frequency Word.

While in Transmit mode the input to the SDM is naturally dithered by the Low Port modulation, which keeps the SDM from generating a pseudo-periodic output. However, in Receive mode, the SDM can potentially generate idle tones (dominant frequency modes at its output) when the input to the SDM is a constant. And in either Transmit or Receive mode the LP SDM can generate idle tones if the Fraction is near zero.

To mitigate these impairments, the Numerator input to the SDM is dithered whenever the Numerator is in the near zero range { -64 to 63 }, or whenever the Radio is in Receive Mode.

The dither operation is implemented using a 3 bit counter to generate a signed 2-bit integer [s2.0]. The resulting range of -4 to 3 produces a saw tooth pattern with a non-zero average (average = LSB/2) that repeats every Reference Clock Period(Fref) / 8. Each LSB of the dither is nominally equal to Fref/(2^16).

47.2.4.4 LPM SDM Timing

The LPM SDM has three accumulation registers that need to be pre-charged to a valid modulation level before the PLL Loop Divider is engaged and the PLL VCO feedback loop is closed. About 10us is needed to charge the accumulators, but the loop divider sigma delta clock is not available during this time because the VCO High Port capacitor array is being calibrated and the effect of the loop divider would disturb that calibration.

Therefore the SDM Timing module is used to pre-charge the accumulators by engaging the SDM with the reference clock as the clock source and a valid Numerator as the modulation target. After the TSM engages the loop divider, there is a clock switcher circuit that safely switches over to the PLL sigma delta clock for normal operation.

47.2.4.5 LPM Delay, Invert, and Disable

The Low Port Modulator has an array of programmable delays available to allow the HPM frequency changes to be matched to the LPM frequency changes and their delay through the PLL Loop Divider and onto the VCO feedback loop.

The Low Port Modulation can be delayed at the PLL Loop Divider by delaying the input to the LP SDM (LPM_SDM_DELAY).

There is an option to invert the Numerator (LPM_SDM_INV) before it is applied to the SDM in order to invert the sigma delta modulation to the PLL.

And the Low Port Modulation of the LP SDM can be disabled (LPM_SDM_DIS); when the modulation is disabled the LP SDM will continue to tune the VCO to the Radio carrier frequency.

47.2.5 Loss of Lock Detector

The PLL Loss of Lock Detector has three levels of detection, each of which can be enabled as a TSM abort source.

47.2.5.1 Coarse Tuning Frequency Failure

If the result of the Coarse Tune Calibration is outside of an acceptable range there is a likelihood that the PLL will not be able to maintain the VCO in a locked state.

In order to abort a Radio operation early in such a Coarse Tune Failure, the TSM abort needs to be configured to Abort-On-CTUNE fail (ABORT_ON_ CTUNE), and the Coarse Tune Fail Level needs to be set to the maximum allowable count difference (CTUNE_LDF_LEV).

47.2.5.2 Fine Tuning Frequency Failure

As the PLL completes the VCO locking time, and just as it is going On Air, the PLL Ripple Counter can be used to make a Fine Tuning check. The count time can be either 4 or 8 us (for either RX or TX), and a maximum allowable absolute count difference can be selected as a failure level.

In order to abort a Radio operation on such a Fine Tuning Frequency Target Failure, the TSM abort needs to be configured to Abort-On-Frequency-Target fail (ABORT_ON_ FREQ_TARG), and the Frequency Target maximum allowable count difference (for both RX and TX) needs to be set.

47.2.5.3 PLL Cycle Slipping failure

The PLL Cycle Slip detector monitors the out of lock flag from the analog phase detector, and sets an interrupt (CS_FAIL) if the Cycle Slip Flag Counts (CS_FCNT) is reached within the Cycle Slip Flag Observation Window (CS_FW).

If the Cycle Slip Flag Observation Window time (CS_FW) expires before the Cycle Slip Flag Counts (CS_FCNT) is reached, then, if the Cycle Slip Recycle (CS_RC) bit is set, the State Machine will hold in a reset state until the Cycle Slip Wait Time expires (CS_WT). When the Cycle Slip Wait Time expires, then the Cycle Slip detector will again search for the maximum allowed Cycle Slip Flag Counts.

In order to abort a Radio operation on a Cycle-Slipping Failure, the TSM abort needs to be configured to Abort-On-Cycle-Slip fail (ABORT_ON_ CYCLE_SLIP).

47.3 Memory Map and Register Definition

The PLL Digital module memory map and detailed descriptions of all its registers is included Tranceiver(XCVR) register section.

Chapter 48
RX Digital

48.1 About this module

48.1.1 Introduction

The RX DIG module implements the receive digital signal processing functions common to both BLE and ZigBee modes. Generally this includes conditioning the input samples from the ADC and produces outputs to the demodulators.

48.1.2 Features

The rx_dig module includes the following features:

- Decimation Filter
- DC Offset Calibration, Estimation and Correction
- Channel Filter
- AGC & RSSI
- I/Q Mismatch Correction
- Normalizer
- Interpolator

48.1.3 Modes and operations

Table 48-1. RX DIG MODES

Mode	RX_DIG_CTRL	Description
ZigBee	0x220	Enable normalizer, set OSR=8
BLE	0x110	Enable interpolator, set OSR=4
ADC Test	0x004	Set OSR=2, enable raw mode

48.1.4 Block diagram

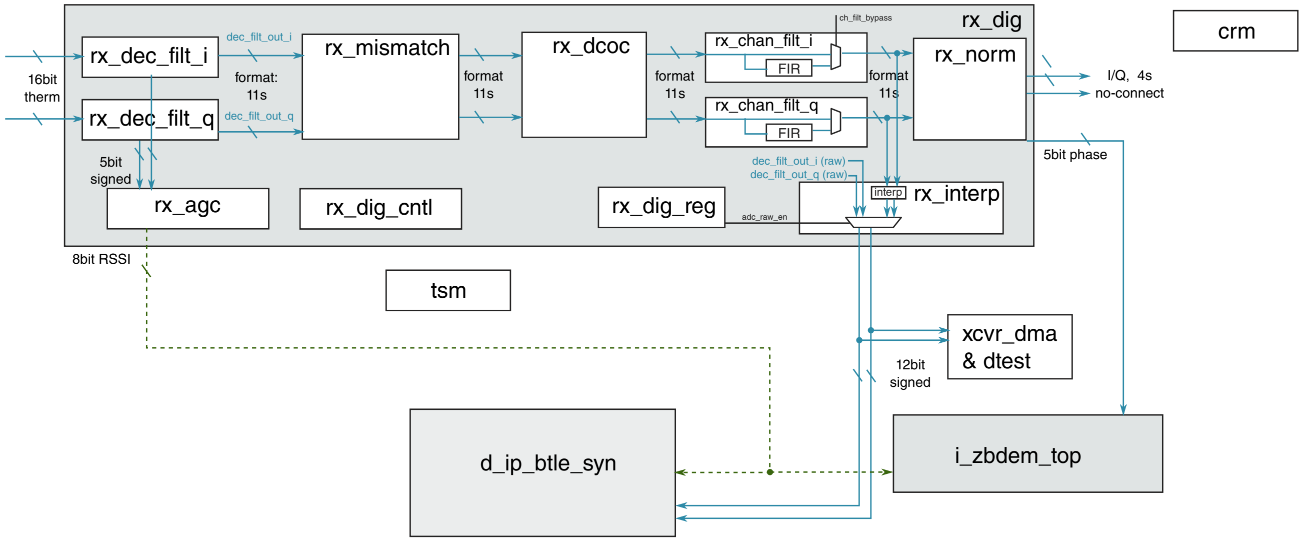


Figure 48-1. RX DIG Block diagram

48.2 Signals

All of the rx_dig I/O signals except for the standard IPS signals are shown in the table below.

Signal	I/O	Description
rx_dig_i[11:0]	O	I channel output
rx_dig_q[11:0]	O	Q channel output
rx_dig_ph[4:0]	O	Phase output
rx_dig_norm_i[4:0]	O	Normalized I output
rx_dig_norm_q[4:0]	O	Normalized Q output
rx_dig_iq_vld	O	I/Q outputs valid
tca_agc[7:0]	O	TZA gain select
bbf_res_tune[3:0]	O	BBF gain select
tza_i_dac[7:0]	O	TZA DCOC DAC code
bbf_i_dac[5:0]	O	BBF DCOC DAC code
tza_q_dac[7:0]	O	TZA DCOC DAC code
bbf_q_dac[5:0]	O	BBF DCOC DAC code

Table continues on the next page...

Signal	I/O	Description
rssi_out[7:0]	O	RSSI out
bbf_pdet_rst_lo	O	Reset to peak detector in analog
bbf_pdet_rst_hi	O	Reset to peak detector in analog
tza_pdet_rst_lo	O	Reset to peak detector in analog
tza_pdet_rst_hi	O	Reset to peak detector in analog
tza_pdet_sel_hi[2:0]	O	Peak detector threshold
tza_pdet_sel_lo[2:0]	O	Peak detector threshold
bbf_pdet_sel_hi[2:0]	O	Peak detector threshold
bbf_pdet_sel_lo[2:0]	O	Peak detector threshold
rssi_ed_0	O	To BLE RSSI Det
rssi_ed_1	O	To BLE AFC ctrl
agc_afc_rst	O	To BLE reset
rssi_adc[7:0]	O	To DTEST
agc_dcoc_cal_rdy	O	To DTEST
dc_cal_complete	O	To DTEST
agc_dcoc_gain_chg	O	To DTEST
curr_agc_idx[4:0]	O	To DTEST
update_correction	O	To DTEST
Idc_est[11:0]	O	To DTEST
Qdc_est[11:0]	O	To DTEST
agc_state[3:0]	O	To DTEST
rx_dig_en	I	rx_dig enable
rx_init	I	rx_dig init
scan_mode	I	scan mode
tsm_rx_dcoc_en	I	dcoc cal en from tsm
tsm_rx_dcoc_init	I	dcoc cal init from tsm
bbf_pdet_out_lo	I	peak detector indicators from analog
bbf_pdet_out_hi	I	peak detector indicators from analog
tza_pdet_out_lo	I	peak detector indicators from analog
tza_pdet_out_hi	I	peak detector indicators from analog
ble_pre_det	I	BLE preamble detect
zb_pre_det	I	ZB preamble detect
orf_rx_access_match	I	BLE access match
zb_lqi_done	I	ZB LQI done
rx_sfd_det	I	ZB LQI done
rx_adc_i[15:0]	I	ADC I channel input
rx_adc_q[15:0]	I	ADC Q channel input
cg1m_en	I	1uS clock gate
clk	I	system clock
reset_b	I	async reset

48.3 Memory Map and register definition

The rx_dig memory map and description of the rx_dig registers is included in Transceiver(XCVR) register section.

48.4 Functional description

The following sections describe functional details of the RX_DIG module.

The RX_DIG module has two overall funtions. Samples at 32MHz are converted from thermometric codes from the ADC to usable formats for the demodulator. Additionally, the module conditions the received sample in various ways, including sending feedback to the analog circuitry.

48.4.1 Clocks

The RX_DIG module uses two clocks, a 32 MHz RF clock (clk) and the IPS gated clock (ips_clk) that is synchronous to the RF oscillator clock. All programming model registers, except for the IQMC_CTRL, IQMC_CAL, DCOC_OFFSET_00-DCOC_OFFSET_26 registers, are clocked by the IPS clock. All other registers are clocked by the RF oscillator clock.

48.4.2 Reset

All RX_DIG registers that require a reset are reset by a global asynchronous reset. The rx_init and tsm_rx_dcoc_init signals also synchronously reset the state of various rx_dig sub-modules.

48.4.3 Decimation Filter

The decimation filter is a 4th order sinc filter that supports oversampling ratios (OSR) of 2, 4 8 and 16.

The decimation filter block takes 16-bit thermometric codes from the ADC as the input samples at 32MHz. After decimation, the samples are output at the appropriate lower rate in 12-bit signed format. Scaling is also applied based on the OSR to maintain consistent headroom.

An OSR of 8 (rx_dec_filt_osr=2) is used for ZigBee mode and an OSR of 4 (rx_dec_filt_osr=1) is used for BLE mode. The decimation filter includes a raw ADC output mode rx_adc_raw_en=1 which bypasses the sinc filter and packs two, unfiltered 5-bit ADC samples into the 12-bit rx_dig outputs rx_dig_i and rx_dig_q. The frequency response of the filter is shown in the figures below.

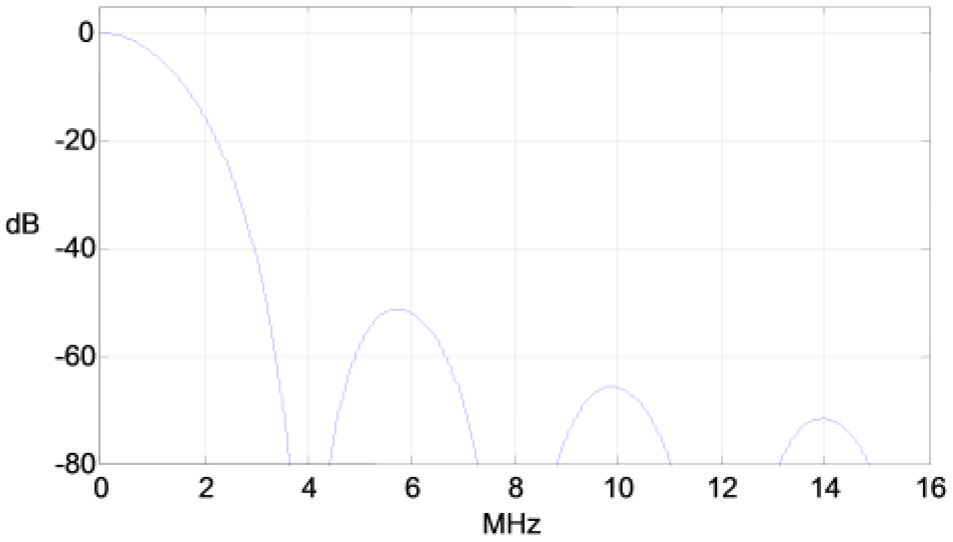


Figure 48-2. ZigBee (OSR=8) Decimation Filter Frequency Response

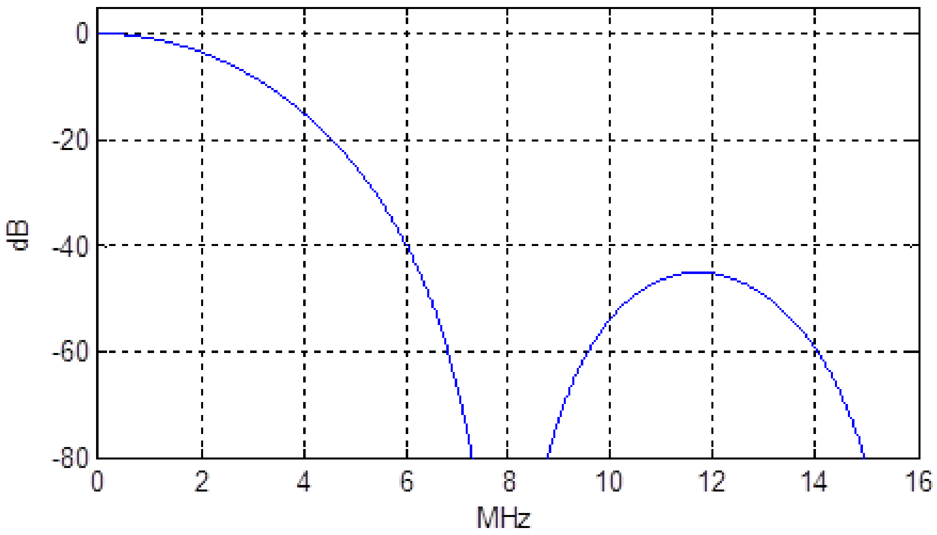


Figure 48-3. BLE (OSR=4) Decimation Filter Frequency Response

48.4.4 Channel Filter

The channel filter is a programmable, 16 tap symmetric FIR filter with 8-bit coefficients. It operates at the decimated sample rate. The coefficient values are programmable and should be configured for the mode of operation (ZigBee or BLE). Example coefficient values are given in the table below.

Coefficient Register	ZigBee	BLE
RX_CHF_COEF0	00	00
RX_CHF_COEF1	00	FF
RX_CHF_COEF2	FF	FD
RX_CHF_COEF3	05	FB
RX_CHF_COEF4	FF	FE
RX_CHF_COEF5	EB	0A
RX_CHF_COEF6	06	1B
RX_CHF_COEF7	4C	27

48.4.5 AGC & RSSI

The AGC & RSSI module has two main functions. It estimates the energy of an incoming signal (RSSI) and it executes the Automatic Gain Control (AGC) by varying the front-end analog gain. These two functions are designed into the same block and the AGC operation relies, at times, on the estimates provided by the RSSI estimator.

48.4.5.1 RSSI Estimator

The RSSI estimator is designed into the overall AGC block. However, it can be enabled and is fully functional even if the AGC is off. The estimator:

- provides a wideband ADC signal level measurement for AGC operation
- generates an accurate in-band signal power level estimate for higher layer processing (e.g., LQI, ED, etc.)
- targets RSSI accuracy with tuneable gains and coefficients

RSSI accuracy is better than +/- 1dB for input signals in the range of -80dBm to 0dBm. This is the intended operating range for the AGC system.

RSSI accuracy is better than +/- 6dB for all input signals in the range of 102dBm to +10dBm.

RSSI estimation is performed using the samples output by the ADC after a thermometric to binary conversion. This signal is first filtered using a low pass 4-tap symmetric FIR filter with un-normalized coefficients [1 3 3 1]. This signal is then decimated by a factor of 4, yeilding a sample rate of 8MHz. The signal magnitude is then estimated with an L1-norm approach as follows:

$$\boldsymbol{Magnitude}(i) = max(|I_i|,|Q_i|) + {}^{3}/_{8} * min(|I_i|,|Q_i|)$$

The estimated magnitude is then passed through a smoothing, low-pass single-tap IIR filter with the transfer function:

$$y[n] = (1-\alpha)*y[n-1] + \alpha*x[n] \text{ ,}$$
$$\text{where } \alpha = 0, {}^{1}/_{2}, {}^{1}/_{4}, {}^{1}/_{8}, {}^{1}/_{16}, {}^{1}/_{32} \text{ (RSSI\_IIR\_WEIGHT)}$$

Following the smoothing filter, the RSSI estimator converts the approximated magnitude to dB. This conversion is acheived using a look-up table that approximates:

$$rssi\_adc\_raw = 20*log_{10}({}^{approx.\ magnitude}/_{ADC\ full\ scale})$$

Finally, the estimator converts the raw dB reading to an estimate for the signal strength at the antenna, using various parameters as follows:

$$rssi\_out = rssi\_adc\_raw - LNM\ gain - BBF\ gain + rssi\_adj$$

For more accurate calculation, when RSSI_USE_VALS=1, the values from the TCA_AGC_VAL and BBF_RES_TUNE_VAL registers are used in the computation of the final dBm estimate. The TCA_AGC_VAL settings are in $^1/_4$ dB increments and represent the actual dB value to be used. The BBF_RES_TUNE_VAL settings are offsets from nominal gain values in $^1/_2$ dB increments.

An additional filter is available to assist in estimation, especially for a CW input signal. This filter restricts the RSSI estimator to a narrow band by filtering the input samples with a single tap IIR filter. This filter has the transfer function:

$$y[n] = (1-\alpha)*y[n-1] + \alpha*x[n]\ ,$$
$$\text{where } \alpha = 0,\ ^1/_8, ^1/_{16}, ^1/_{32}\ \text{(RSSI\_IIR\_CW\_WEIGHT)}$$

The measurement of RSSI will hold briefly if a gain change occurs. This is to avoid invalid, intermediate readings. Additionally, RSSI will be held if RSSI_HOLD_EN=1 and the signal selected by RSSI_HOLD_SRC asserts.

The RSSI system also provides an Energy Detect (ED) feature. This feature simply allows 2 ED signals to assert when a threshold is exceeded. They will deassert when the corresponding hysteresis window has been exited.

48.4.5.2 AGC

The Automatic Gain Control (AGC) system is designed to ensure that the received signal does not become distorted because of clipping in the analog receive chain, while attempting to maintain an optimal signal level for demodulation. There are two locations where gain may be adjusted: analog BBF, and analog LNM. The AGC system manipulates these gains to maintain a targeted ADC input signal level. The targeted ADC signal level is chosen as a compromise between receiver linearity, blocker headroom requirements and the goal to maximize the digital signal SNR that is provided to the protocol-specific PHY demodulators.

Additionally, the AGC system may be manually controlled by setting the USER_BBF_GAIN_EN and USER_LNM_GAIN_EN bits. If these bits are set, the gains selected in BBF_USER_GAIN and LNM_USER_GAIN will be applied.

The key components of the AGC system are: peak detectors in the analog front end, gain control, RSSI estimation, and a state machine. Additionally, the AGC system has appropriate interation with the DCOC block to maintain a coherent state of the gains.

If the manual control of the gains is not enabled, and the AGC system is enbled, the system will automatically control the gains based on peak detector states and/or RSSI measurements.

The AGC sytem operated with a state machine that is depicted in Figure 48-4 and has the following significant modes:

- **IDLE**: No actions are taken. The system waits for either rx_init (for normal receive operation) or cal_init (for DC calibration).
- **DC Calibration**: When cal_init is observed while in IDLE, the AGC enters the DC Calibration state. According to values programmed in XCVR_DCOC_CAL_GAIN, gains will be applied for DC CAL 1, 2, 3 phases and the DCOC block can take measurements of the DC level with those gains applied. During DC Calibraion, the AGC system takes no other action.
- **FAST AGC**: When rx_init is observed while in IDLE, the system advances to FAST AGC. Maximum gain is applied immediately. While in FAST AGC, the high level peak detectors are observed to see if clipping is occurring in the analog. If a high peak detector becomes set, a gain adjustment is applied based on the settings AGC_DOWN_TZA_STEP_SZ, AGC_DOWN_BBF_STEP_SZ, and AGC_GAIN_TBL. Depending on which of the two peak detectors is set, the corresponding step is taken in the gain table. When a change is made, a timer prevents further changes until the gain change has had time to settle. This time is defined by the settings TZA_GAIN_SETTLE_TIME and BBF_GAIN_SETTLE_TIME. This time is also utilized to reset the state of the peak detectors. FAST AGC can be exited by reaching the minimum possible gain setting, or if the fast_expire timer expires. This time is configured with the AGC_FAST_EXPIRE setting.
- **SLOW AGC**: The SLOW AGC state allows for observation of RSSI, while still monitoring the peak detectors. Alternatively, RSSI can be ignored and the low peak detectors can be used to take upward gain steps. These steps are smaller than the steps taken in FAST AGC. Before entering SLOW AGC, there is a PRESLOW state where headroom is quickly checked. If there is not enough ADC headroom, a small downward gain step is taken, and then the system procedes to SLOW AGC. Similar to FAST AGC, SLOW AGC will allow for a gain settling time whenever an adjustment is made. SLOW AGC can be exited because of several condtions. If a high peak detector indicates clipping, the system immediately moves to the FAST

AGC state. If a drastic change in measured RSSI occurs, the system will similarly transition to the FAST AGC state. Finally, if the slow_expire timer reaches the pre-determined time, the system will move to HOLD AGC.

- **HOLD**: The AGC HOLD state will not take any actions unless a high peak detector is set, a large RSSI change is experienced, or the hold_expire timer expires. These conditions cause transitions into approriate states to take action. The expiration of the hold_timer moves the system back to SLOW AGC, where monitoring of the RSSI can be done and gain increases can happen, if appropriate.

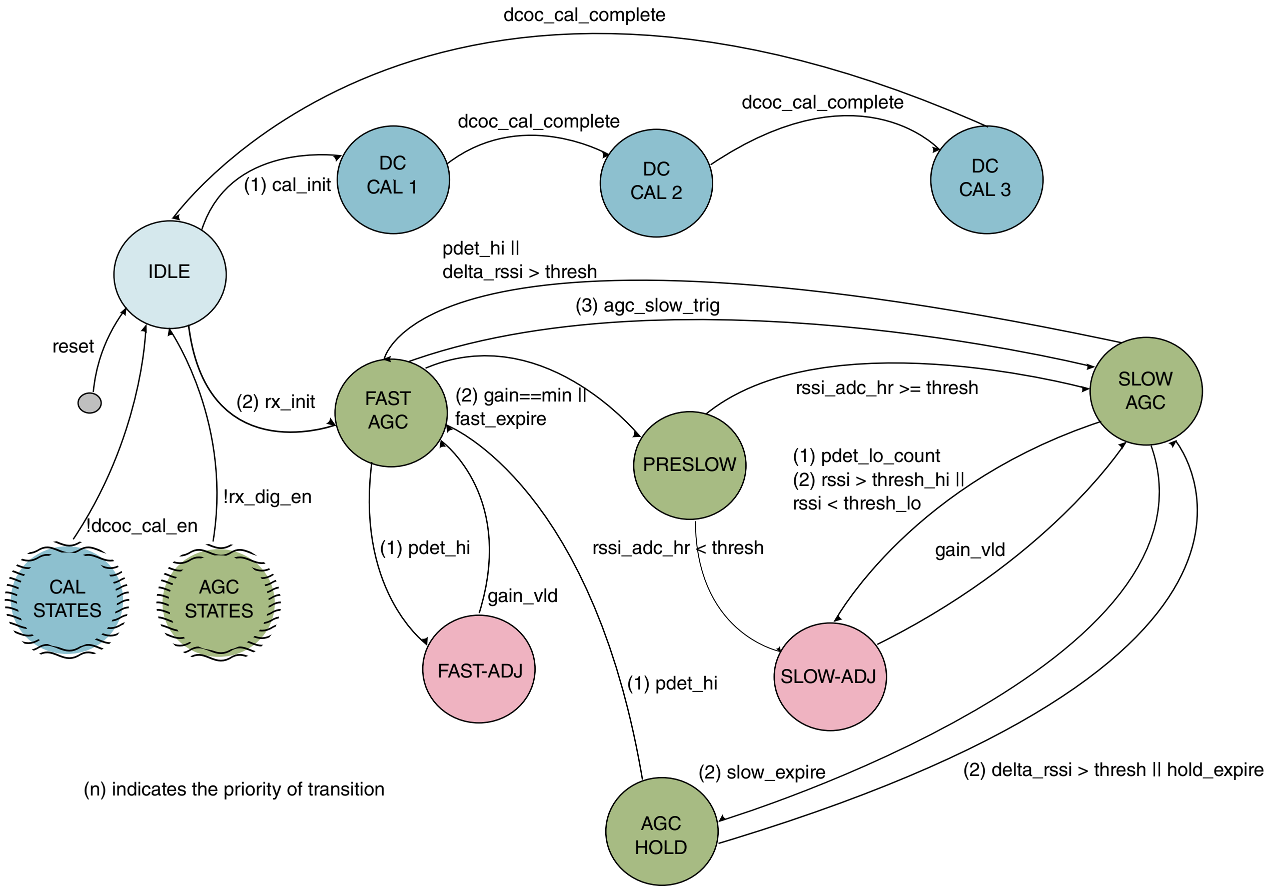


Figure 48-4. AGC State Machine

As previously mentioned, gain steps are performed by striding through the user programmed gain table. There are 27 entries in the table, and each entry contains an index for LNM gain and an index for BBF gain. The mapping of these indicies to codes and nominal gains is described in Table 48-2 and Table 48-3. These values can also be overridden manually and an alternate code can be used for debug purposes. Entry 0 is interpreted as the lowest possible gain setting, and entry 26 the highest.

Table 48-2. LNM Gains

lnm_gain (AGC_GAIN_TBL)	LNM Gain (dB)	tca_agc code
0	-3	FE
1	3	FD
2	9	FB
3	15	F7
4	21	EF
5	27	DF
6	33	BF
7	39	7F
8	45	00
F	--	LNM_ALT_CODE[7:0]

Table 48-3. BBF Gains

bbf_gain (AGC_GAIN_TBL)	BBF Gain (dB)	bbf_res_tune code
0	0	A
1	3	9
2	6	8
3	9	7
4	12	6
5	15	5
6	18	4
7	21	3
8	24	2
9	27	1
A	30	0
F	--	BBF_ALT_CODE[3:0]

48.4.6 DC Offset Correction

DC offsets in a direct conversion receiver can originate in RF (e.g., LO leakage, IP2, etc.), baseband (e.g., operational amplifier offsets, mismatches, etc.) and interfaces (due to DC level mismatches between various RF/analog circuits).

The DC (or origin) offset estimation and correction loop in Apache comprises of two distinct DC estimators. One of the estimators is used during wake-up calibration of DC offsets, while a second adaptive estimator is used during normal receiver operation. Both DC estimates are computed on the Rx I/Q data after the decimation stage. In the Apache receiver, there are three distinct DC offset correction points, which are at the TZA output, BBF output (after first BBF active filter stage) and in the digital domain after the decimation filter. The DC estimation and correction systems operate as follows:

- A DC offset calibration is carried out at receiver warm-up to compute independent I/Q DC offsets at three distinct RX gain points.
- A DC correction is then estimated for all the entries in the AGC gain table using the 3 calibration estimates. Note that only TZA and BBF DC corrections are stored in the AGC gain lookup table.
- During AGC, the system applies a DC offset correction corresponding to the {LNM, BBF} gain combination that is applied.
- DC tracking estimates are reset upon a gain change and the algorithm is seeded with the calibrated DC estimates corresponding to the active gain setting
- Once AGC gain is settled, DC offset tracking estimation continues for a programmed amount of time
- After a pre-programmed interval expires, the tracked DC offset is sampled and a DC offset correction is estimated and applied
- Based on configuration, the DC offset tracking update can be applied once or iterated upon a programmable number of times.

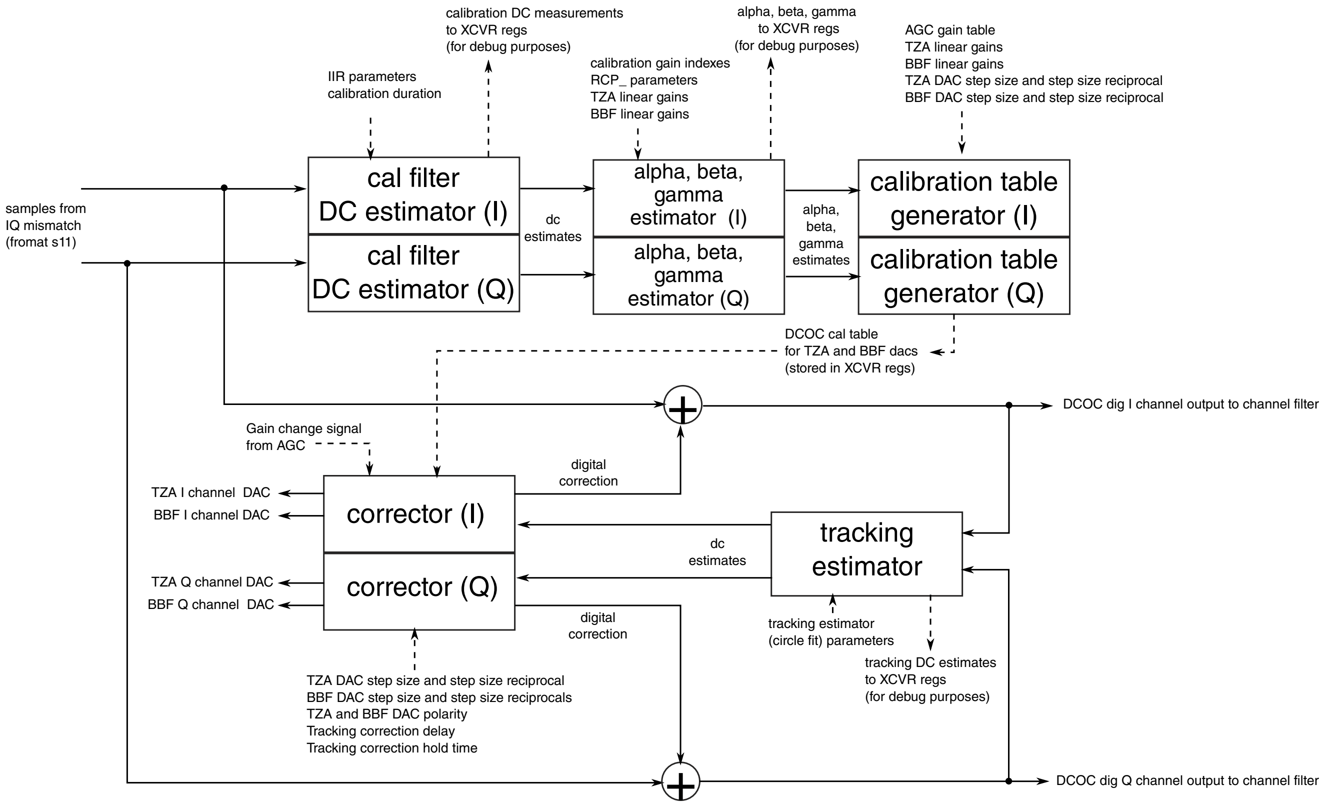


Figure 48-5. RX_DIG DCOC Block Diagram

48.4.6.1 DC Offset Calibration

If the XCVR_RX_DIG_CTRL[RX_DCOC_EN] and XCVR_RX_DIG_CTRL[RX_DCOC_CAL_EN] bits are set, the Apache direct conversion receiver will perform a DC offset calibration for I/Q receive data paths at every RX warm-up. The calibration will compute a table of DC correction values (XCVR_DCOC_OFFSET_n registers) for each of the 27 AGC gain table entries.

The calibration starts when the TSM DCOC_INIT signal pulses high then low.The total calibration time includes the time for each of three DC estimates (DCOC_CAL_DURATION) and overhead to compute the DCOC_OFFSET_n table :

```
calibration time (in µs) = 3*XCVR_DCOC_CTRL_0[DCOC_CAL_DURATION] + (3*27 + 7)/32MHz
```

A value of DCOC_CAL_DURATION=19 results in a calibration time of 59.75us.

After calibration, the DC correction corresponding to AGC gain table entry 26 is applied when TSM rx_dig_en is asserted. AGC gain table entry 26 is the initial gain value used by the AGC, and initializing the DC correction values at this time allows for the DCOC

DACs to settle a bit before the AGC begins. Following that, the DC correction values are updated whenever the AGC changes the gain, and would also be updated if the DCOC tracking mechanism is enabled as described later inDC Offset Tracking.

The figure below illustrates the DC Offset Calibration timing.

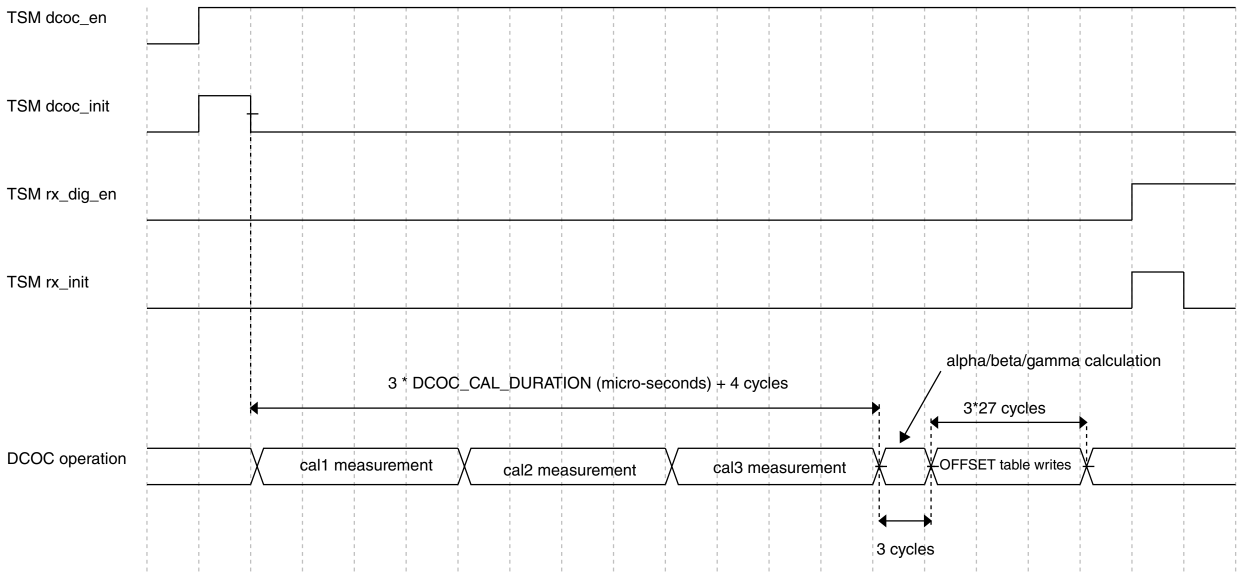


Figure 48-6. DCOC Calibration Timing Diagram

48.4.6.1.1 DC Offset Estimation during Calibration Phase

During the calibration phase, the DCOC makes DC offset estimates at three distinct RX TZA/BBF gain points. The three gain points are programmed in the XCVR_DCOC_CAL_GAIN register, and are described more in the next section. The duration of the DC offset estimate for each of the 3 gain points is programmable up to 31us as defined by XCVR_DCOC_CTRL_0[DCOC_CAL_DURATION].

DC offset estimation during calibration is carried out by using a cascade of 3 single-tap IIR filters of the type

$y_k[n] = (1-\alpha_k) y_k[n-1] + \alpha_k x_k[n]$

where α_i are programmable co-efficients assuming values shown below as programmed in the XCVR_DCOC_CAL_IIR register:

- α_1: {1, 1/4, 1/8, 1/16}
- α_2: {1, 1/4, 1/8, 1/16}
- α_3: {1/4, 1/8, 1/16, 1/32}

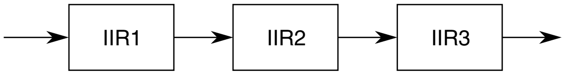


Figure 48-7. DC Offset Estimation using a filter bank during calibration

Using a cascade of 3 IIR filters as shown with α_1=1/16; α_2=1/16; α_3=1/16, the transfer function of the cascaded filter is a narrowband filter with a 3dB bandwidth of approximately 170kHz and a settling time of < 18µs.

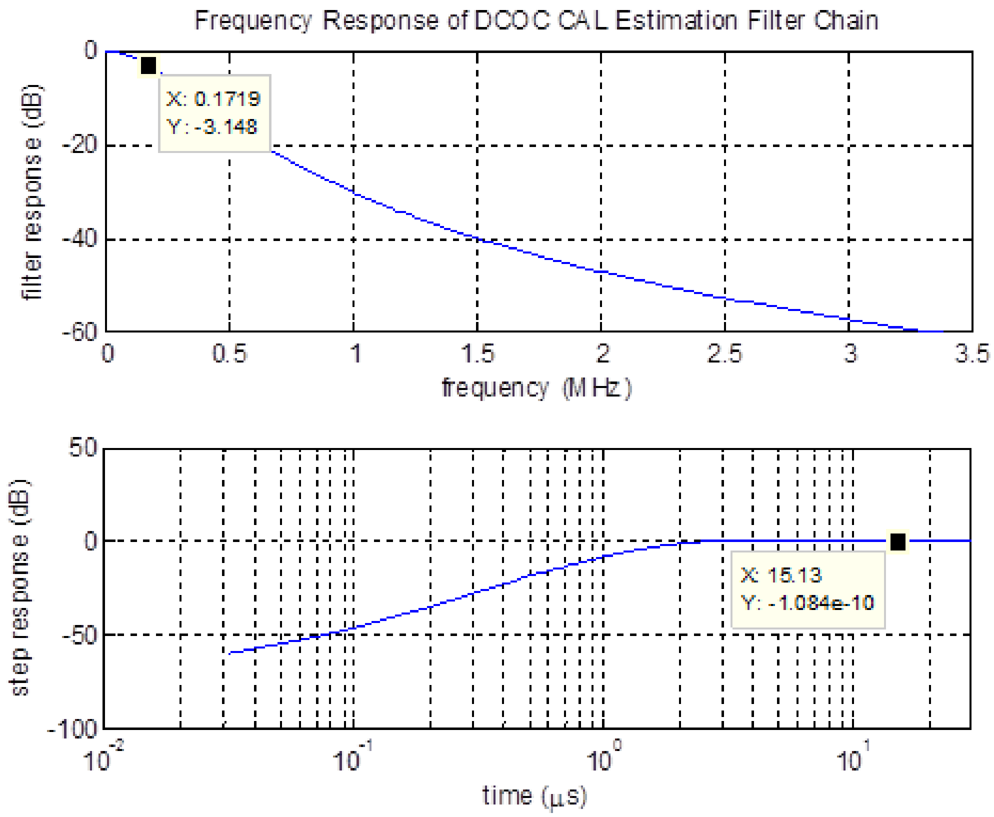


Figure 48-8. Magnitude and Step response of DC Offset Calibration Estimator

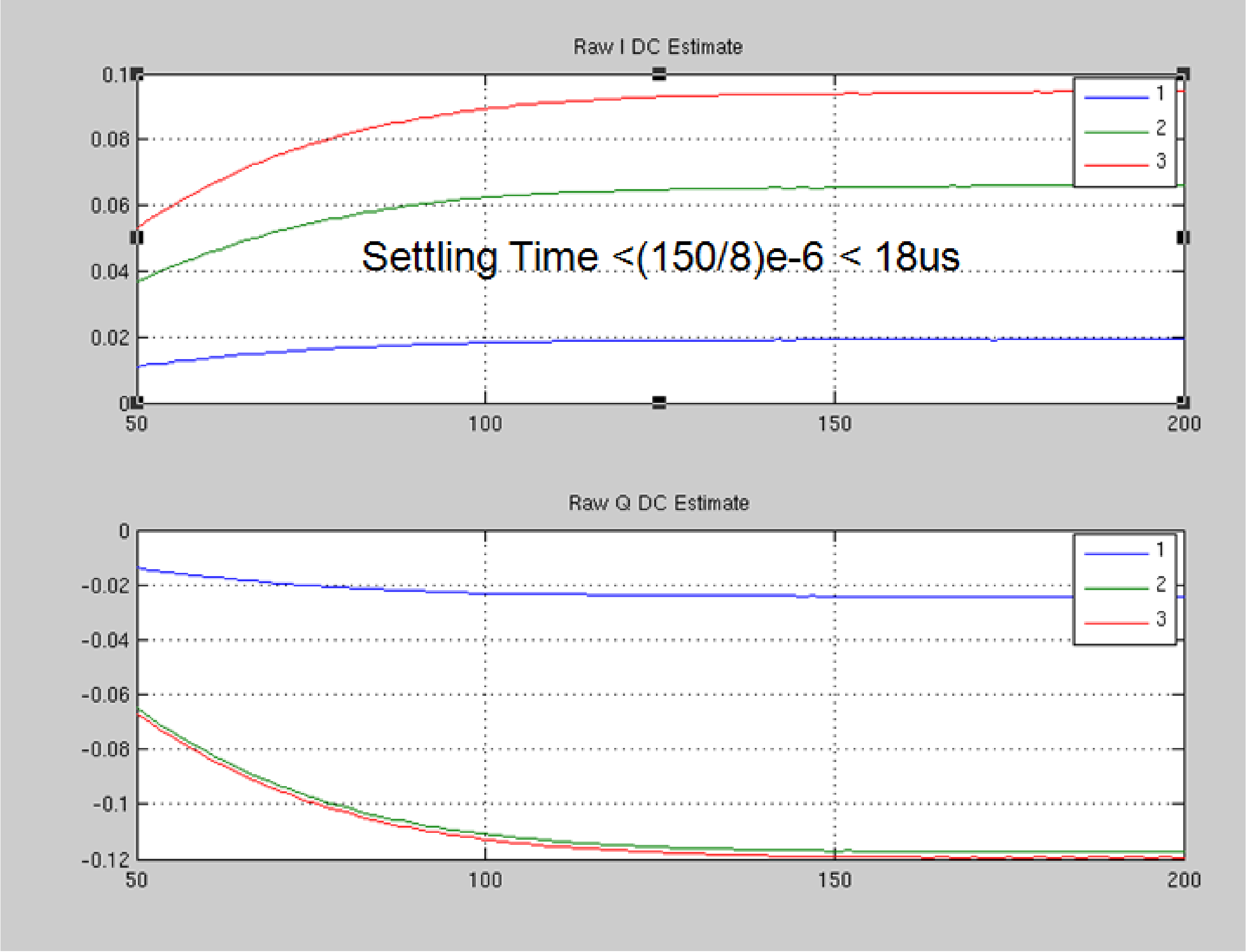


Figure 48-9. 3-step {I,Q} DC estimation during calibration

For debug, the DC offset estimates for the three gain points can be read after calibration from the XCVR_DCOC_CAL1, XCVR_DCOC_CAL2, and XCVR_DCOC_CAL3 registers.

Note that a separate narrow-band DC offset estimator is needed for calibration because a sigma-delta ADC can potentially convert a DC signal to an idle tone. For Apache RxADC, a pure DC stimulus can results in an idle-tones with a frequency of 1.5-2kHz.

The DCOC operates on the decimated output of the ADC. As BLE and ZigBee have different oversample rates, different IIR filter settings are expected to be used assuming a settling time of 18us is desired. The table below shows sets of IIR1/2/3 parameters for BLE and ZigBee which were used during DCOC verification:

- BLE: α_1 = 1/8, α_2=1/16, α_3=1/16
- ZigBee: α_1 = 1/4, α_2 = 1/8, α_3 = 1/8

48.4.6.1.2 DC Estimation at 3 operation gain points of the receiver

Apache ZIF receiver performs a 3 point DC Offset calibration. The calculations shown below are shown for one branch of Rx (I or Q) only. In hardware the calculations are carried out independently for both I and Q branches and stored independently in the AGC/DCOC lookup table

The Apache DC offset calibration scheme approximates the DC offset behavior of the system observed at three distinct gain settings by calculating the DC offset for the three gain points, and decomposing the correction into 3 DC components, which are

- α - An RF component (such as LO feed through), which scales with both LNM and BBF gains
- β - A TZA DC offset which scales with BBF gains
- γ - represents an un-scaled DC offset at the ADC input

These DC components are illustrated in the figure below.

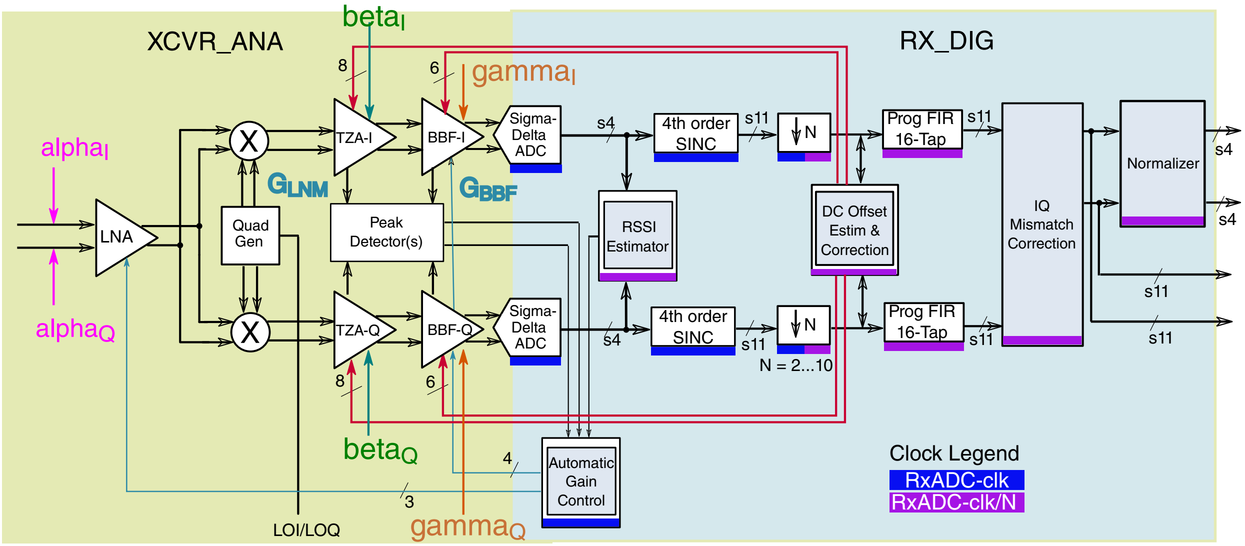


Figure 48-10. Receiver model showing three contributors of DC offset

The DC offset correction after calibration is applied at 2 points in the receiver, which are at

- the TZA output using an 8-bit DAC with LSB = 1.4V/2^8 = 5.5mV
- the BBF output using a 6-bit DACs with LSB = 1.2V/2^6 = 18.7mV

Calibration is performed open-loop. During calibration, the DCOC measures the DC values while the DACs are configured with their default control values which apply an offset of 0V.

The equations below provide the calculations from 3 distinct DC offset measurements to the estimation of the three decomposed DC offset components.

At any given time, the total DC offsets observed at ADC input would follow the formula

$\mathtt{RxDCOC\_pre\_ADC} \cong (\alpha * G_L + \beta) * G_B + \gamma$

where

- G_L – Voltage gain of the LNM stage (in linear domain)
- G_B – Voltage gain of the BBF stage (in linear domain)

For Apache, ADC gain is nominally -1.7dB, though this may need to be adjusted based on actual silicon measurements. There is also a conversion factor from mV to the quantized ADC output of (2^11/1000). Let

```
ADC_gain = 10^(-1.7/20) * (2^11 /1000)
```

Note that the ADC_gain factor is a component of the value programmed for the BBF_DCOC_STEP, BBF_DCOC_STEP_RECIP, DCOC_TZA_STEP_GAIN, and DCOC_TZA_STEP_RECIP bitfields.

The total DC offset observed at the RxDCOC is

$\mathtt{RxDCOC\_post\_ADC} \cong \mathtt{ADC\_gain} * ((\alpha * G_L + \beta) * G_B + \gamma)$

The DC offset calibration is carried at three distinct {LNM, BBF} gain settings, as summarized below.

Table 48-4. DC offset calibration at selected {LNM,BBF} gain settings

DCOC_CAL	LNM linear gain (V/V)	BBF linear gain (V/V)	DC offset Measured at ADC input
DCOC[1]	G_{L_LO}	G_{B_LO}	ADC_gain *((α * G_{L_LO} + β)*G_{B_LO} + γ)
DCOC[2]	G_{L_HI}	G_{B_LO}	ADC_gain *((α * G_{L_HI} + β)*G_{B_LO} + γ)
DCOC[3]	G_{L_LO}	G_{B_HI}	ADC_gain *((α * G_{L_LO} + β)*G_{B_HI} + γ)

The three gain point settings are defined by programming the XCVR_DCOC_CAL_GAIN register. Here is an example:

Table 48-5. Example DCOC_CAL_GAIN programming

DCOC_CAL	DCOC TZA CAL GAIN index	DCOC BBF CAL GAIN index
DCOC[1]	3 (corresponds to G_{L_LO} of 15dB)	5 (corresponds to G_{B_LO} of 15dB)
DCOC[2]	6 (corresponds to G_{L_HI} of 33dB)	5 (corresponds to G_{B_LO} of 15dB)
DCOC[3]	3 (corresponds to G_{L_LO} of 15dB)	10 (corresponds to G_{B_HI} of 30dB)

Calculations of estimates for α , β and γ are carried out post ADC, i.e., ADC_gain is used implicitly in calculations.

Using [1] & [2]

α-hat = α *ADC_gain = (DCOC[2] - DCOC[1]) * [1.0((G$_{L\_HI}$-G$_{L\_LO>}$)*G$_{B\_LO}$)] [4]

Using eq. [1] & eq. [3]

DCOC_tmp = (α-hat * G$_{L\_LO}$ + β-hat)*ADC_gain
= (DCOC[3] - DCOC[1]) * [1.0/(G$_{B\_HI}$ - G$_{B\_LO}$)] [5]

Using eq. [4] & eq. [5]

β-hat= β *ADC_gain = [5] - [4]*G$_{L\_LO}$ [6]

Now, using eq. [5] & eq. [3]

γ-hat = γ *ADC_gain = [3] - [5]* G$_{B\_HI}$ [7]

After calibration, α-hat, β-hat, and γ-hat can be read from the XCVR_DCOC_CAL_ALPHA, XCVR_DCOC_CAL_BETA, and XCVR_DCOC_CAL_GAMMA registers, respectively for debug purposes.

To perform the α-hat/β-hat/γ-hat calculations, the DCOC also uses the following register bitfields:

- XCVR_DCOC_CAL_RCP[ALPHA_CALC_RECIP]. This is the 1.0/((G$_{L\_HI}$-G$_{L\_LO>}$)*G$_{B\_LO}$) factor from equation [4] above.
- XCVR_DCOC_CAL_RCP[DCOC_TMP_CALC_RECIP]. This is the 1.0/(G$_{B\_HI}$-G$_{B\_LO}$) factor from equation [5] above.
- TCA_AGC_LIN_VAL_x (XCVR_TCA_AGC_LIN_VAL_x_y registers). The G$_{L\_LO}$ and G$_{L\_HI}$ gains are found from the appropriate index into this table.
- BBF_RES_TUNE_LIN_VAL_x (XCVR_BBF_RES_TUNE_LIN_VAL_x_y registers). The G$_{B\_LO}$ and G$_{B\_HI}$ gains are found from the appropriate index into this table.

48.4.6.1.3 Calculation of DC Offset Table

Once the estimates for α , β and γ , that is, α-hat, β-hat and γ-hat are computed, the calibrated DC offset correction values for the 27 entries in the AGC gain table (XCVR_AGC_GAIN_TBL_xx_yy registers) can be calculated and stored (in the XCVR_DC_OFFSET_n registers).

The calculations are independently carried out for both the I/Q branches of the transceiver but only one branch is shown below. The computations are as follows for all k=0,1,... 26 entries of the AGC_GAIN_TBL, and where TDAC and BDAC represent the DC offset estimate LSB for TZA and BBA DACs:

DCOC_TZA_TOTAL = α-hat * G$_{Lk}$ + β-hat

TDAC$_k$ = round(DCOC_TZA_TOTAL*[1.0/TZA_dcoc_step_norm])

DCOC_TZA_RESIDUAL = DCOC_TZA_TOTAL - TDAC$_k$ * TZA_dcoc_step_norm

DCOC_BBF_TOTAL = DCOC_TZA_residual *G$_{Bk}$ + γ-hat

BDAC$_k$ = round(DCOC_BBF_TOTAL*[1.0/BBA_dcoc_step])

Information derived from programming of the XCVR_AGC_GAIN_TBL_xx_yy registers		Computed and stored in the XCVR_DC_OFFSET_n registers			
LNM linear gain (V/V)	**BBF linear gain (V/V)**	**TZA-I DAC code**	**BBF-I DAC Code**	**TZA-Q DAC code**	**BBF-Q DAC Code**
G_{L0}	G_{B0}	$TDAC_{I0}$	$BDAC_{I0}$	$TDAC_{Q0}$	$BDAC_{Q0}$
…	…	…	…	…	…
G_{L26}	G_{B26}	$TDAC_{I26}$	$BDAC_{I26}$	$TDAC_{Q26}$	$BDAC_{Q26}$

Note that the values stored in the DC_OFFSET_n table represent the DC offset at the DAC. Normally the DCOC will then apply the negative of this DC offset to the DAC when it applies the correction, but this depends on the XCVR_DCOC_CTRL_1[TZA_CORR_POL] and XCVR_DCOC_CTRL_1[BBA_CORR_POL] bits. Also, note that the DAC code values themselves (which are reflected in the XCVR_DCOC_STAT register) include a bias offset of 0x80 for the TZA DACs and 0x20 for the BBA DACs.

To compute the TDAC and BDAC values for the XCVR_DC_OFFSET_n registers, the DCOC uses the following register bitfields:

- XCVR_DCOC_CTRL_2[BBF_DCOC_STEP_RECIP]. This is the 1.0/ BBA_dcoc_step referred to above.
- XCVR_DCOC_TZA_STEP_0[DCOC_TZA_STEP_RCP]. This is the 1.0/ TZA_dcoc_step_norm referred to above.
- XCVR_DCOC_TZA_STEP_0[DCOC_TZA_STEP]. This is the TZA_dcoc_step_norm referred to above.
- TCA_AGC_LIN_VAL_x (XCVR_TCA_AGC_LIN_VAL_x_y registers). This is used for the G_{Lk} referred to above
- BBF_RES_TUNE_LIN_VAL_x (XCVR_BBF_RES_TUNE_LIN_VAL_x_y registers). This is used for the G_{Bk} referred to above

48.4.6.1.4 Use of DC offset Table After Calibration

After calibration, when the TSM RX_DIG_EN is asserted, if the XCVR_DCOC_CTRL_1[DCOC_CORRECT_EN] bit is 1, the DCOC will apply the value stored in the DC_OFFSET_n table on each AGC gain change. As indicated previously, the value stored in the DC_OFFSET_n register represents the DAC vaue of the DC offset.

If the XCVR_DCOC_CTRL1[TZA_CORR_POL] bit is clear (normal polarity), the DCOC will subtract the TDAC value from the TZA DAC zero bias value of 0x80. If the TZA_CORR_POL bit is set, the DCOC will add the TDAC value to the TZA DAC zero bias value.

If the XCVR_DCOC_CTRL1[BBA_CORR_POL] bit is clear (normal polarity), the DCOC will subtract the BDAC value from the BBA DAC zero bias value of 0x20. If the BBA_CORR_POL bit is set, the DCOC will add the BDAC value to the BBA DAC zero bias value.

The DCOC also allows software to write to the DC_OFFSET_n table. In this case, the XCVR_RX_DIG_CTRL[RX_DCOC_CAL_EN] bit should be cleared so that the DCOC will not overwrite the DC_OFFSET_n table, but XCVR_DCOC_CTRL_1[DCOC_CORRECT_EN] bit should still be set.

48.4.6.2 DC Offset Tracking

The DCOC also supports a closed-loop DC tracking estimator which can be used with or without DC Offset calibration. The tracking estimator applies a DAC correction and also applies a digital correction to the downsampled output of the ADC.

The tracker is used if the XCVR_DCOC_CTRL_0[DCOC_TRACK_EN] bit is programmed to 1 (XCVR_RX_DIG_CTRL[RX_DCOC_EN] and XCVR_DCOC_CTRL_1[DCOC_CORRECT_EN] should also be programmed to 1). The tracking estimator operation is as follows:

- On every AGC gain change, the DCOC will apply the DC offset correction to the DACs from the DCOC_OFFSET_n table, reset the digital correction to zero, and resets the tracker's estimate to either 0 or the current I/Q sample, depending on programming of the XCVR_DCOC_CTRL_1[TRACK_FROM_ZERO] bit.
- The DCOC will periodically adjust the DACs and digital correction at intervals defined by the XCVR_DCOC_CTRL_0[DCOC_CORR_DLY] bitfield until a timeout defined by the XCVR_DCOC_CTRL_0[DCOC_CORR_HOLD_TIME] bitfield is reached.
- The tracking estimator supports two set of parameters so that estimator can be gearshifted from a high bandwidth to a narrow bandwidth configuration after a number of update corrections defined by XCVR_ANA_SPARE[DCOC_TRK_EST_GS_CNT].

The figure below illustrates the timing behavior during DC offset tracking correction after an AGC gain change. If the XCVR_DCOC_CTRL_0[DCOC_CORRECT_SRC] bit were programmed to 0 instead of 1, only the DCOC DAC change at agc_gain_chng pulse would occur and the digital correction applies to the signal through the DCOC would remain at 0. The figure shows an example where trk_est_gs_cnt=2, so the tracking estimator changes its parameters after 2 update correction periods.

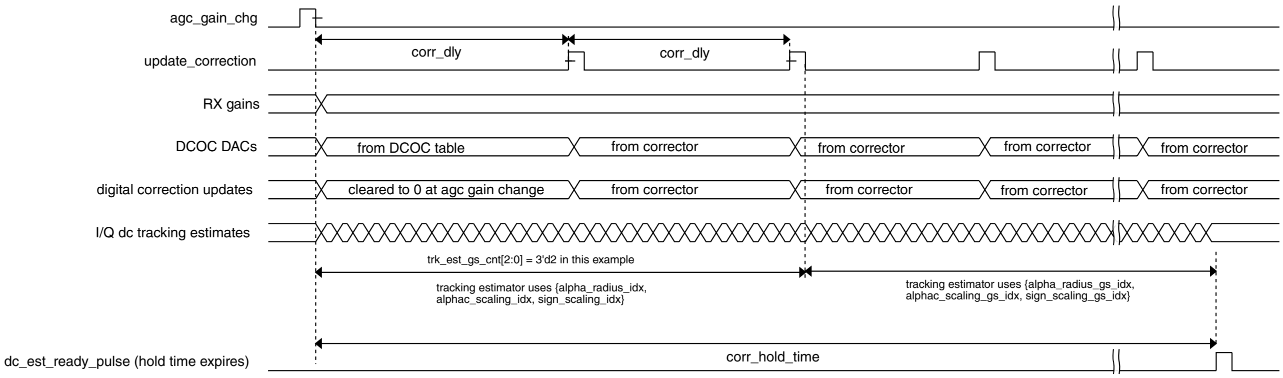

agc_gain_chg
update_correction
RX gains
DCOC DACs
digital correction updates
I/Q dc tracking estimates
dc_est_ready_pulse (hold time expires)
corr_dly
corr_dly
from DCOC table
from corrector
from corrector
from corrector
from corrector
cleared to 0 at agc gain change
from corrector
from corrector
from corrector
from corrector
trk_est_gs_cnt[2:0] = 3'd2 in this example
tracking estimator uses {alpha_radius_idx, alphac_scaling_idx, sign_scaling_idx}
tracking estimator uses {alpha_radius_gs_idx, alphac_scaling_gs_idx, sign_scaling_gs_idx}
corr_hold_time

Figure 48-11. DC Offset Tracking Correction Timing

48.4.6.2.1 DC Offset Estimation during Tracking Phase

Apache has two distinct DC offset estimators: the one use during calibration described previously, and one used for DC tracking during normal receiver operation.

DC Offset estimation during normal receiver operation requires estimation of DC signal in the presence of a constant envelope modulation signal. A circle-fit type of algorithm is used for DC estimation in the presence of desired as well as interfering signals.

The circle-fit algorithm relies on the programmable bitfields shown in the list below. Note that there are two sets of alpha_radius and alphac_scale parameters (but only one sign_scale_idx parameter), which allows the algorithm to gearshift from a high-bandwidth configuration {alpha_radius_idx, alphac_scale_idx, sign_scale_idx} to a low-bandwidth configuration {alpha_radius_gs_idx, alphac_scale_gs_idx, sign_scale_idx} after the number of update corrections defined by trk_est_gs_cnt.

- XCVR_DCOC_CTRL_0[DCOC_ALPHA_RADIUS_IDX], and XCVR_ADC_TEST_CTRL[DCOC_ALPHA_RADIUS_GS_IDX] used as alpha_radius_gs_idx. Step size for radius, assumes values of {1, 1/2, 1/4, 1/8, 1/16, 1/32, or 1/64}. In pseudo-code below, this is referred to as "alpha_radius".
- XCVR_DCOC_CTRL_0[DCOC_ALPHAC_SCALE_IDX], and XCVR_BBF_CTRL[DCOC_ALPHAC_SCALE_GS_IDX] used as alphac_scale_gs_idx . Step size for I/Q center, assumes values of {1/2, 1/4, 1/8 or 1/16}. In psuedo-code below, this is referred to as "alphac_center"

- XCVR_DCOC_CTRL_0[DCOC_SIGN_SCALE_IDX] . sign_scaling. Scaling factor for sign() based correction of the I/Q center, assumes values of {1/2, 1/4, 1/8, or 1/16}. In psuedo-code below, this is referred to as "sign_scaling".
- XCVR_DCOC_CTRL_1[TRACK_FROM_ZERO]. This control bit determines the initial value used by the tracking estimator on each AGC gain change. If TRACK_FROM_ZERO=1, the tracking estimate starts from I/Qcenter=0; if TRACK_FROM_ZERO=0, the tracking estimator starts from I/Qcenter using the current I/Q sample. TRACK_FROM_ZERO=1 can be used in conjunction with calibration, whereas TRACK_FROM_ZERO=0 can be used with or without calibration and also works well at a wider range of RSSI levesl.

Pseudo-code for the circle-fit algorithm in the tracking estimator is as follows:

```
epsilon_I = I - Icenter;
epsilon_Q = Q - Qcenter;
i_abs = abs(epsilon_I);
q_abs = abs(epsilon_Q);
delta_radius = max(i_abs,q_abs)+min(i_abs,q_abs)*3/8;
delta_Icenter = I - Radius*sign(epsilon_I)*sign_scaling;
delta_Qcenter = Q - Radius*sign(epsilon_Q)*sign_scaling;
Icenter = alpha_center*delta_Icenter + (1 - alpha_center)*Icenter;
Qcenter = alpha_center*delta_Qcenter + (1 - alpha_center)*Qcenter;
Radius = alpha_radius*delta_radius + (1 - alpha_radius)*Radius;
```

For debug, the tracker's DC estimate (Icenter, Qcenter) can be read from the XCVR_DCOC_DC_EST register.

The DC correction applied by the tracking algorithm makes use of the following register bitfields:

- XCVR_DCOC_CTRL_1[BBF_DCOC_STEP].
- XCVR_DCOC_CTRL_2[BBF_DCOC_STEP_RECIP].
- XCVR_DCOC_TZA_STEP_RCP (XCVR_DCOC_TZA_STEP_n registers).
- XCVR_DCOC_TZA_STEP (XCVR_DCOC_TZA_STEP_n registers).
- XCVR_DCOC_CTRL1[TZA_CORR_POL]
- XCVR_DCOC_CTRL1[BBA_CORR_POL]

48.4.7 I/Q Mismatch Correction

The I/Q mismatch block corrects for I/Q gain and phase mismatch. It can correct up to +/- 9 degrees of phase mismatch and +/- 4 dB of gain mismatch. After correction, phase mismatch is reduced to less than 0.5 degree and gain mismatch less than 0.3 dB.

Before the receiver is enabled for receiving packets, calibration is run to determine the I/Q phase adjustment coefficient (iqmc_phase_adj) and the I/Q gain adjustment coefficient (iqmc_gain_adj). The gain calibration and phase calibration loops are shown in the figures below.

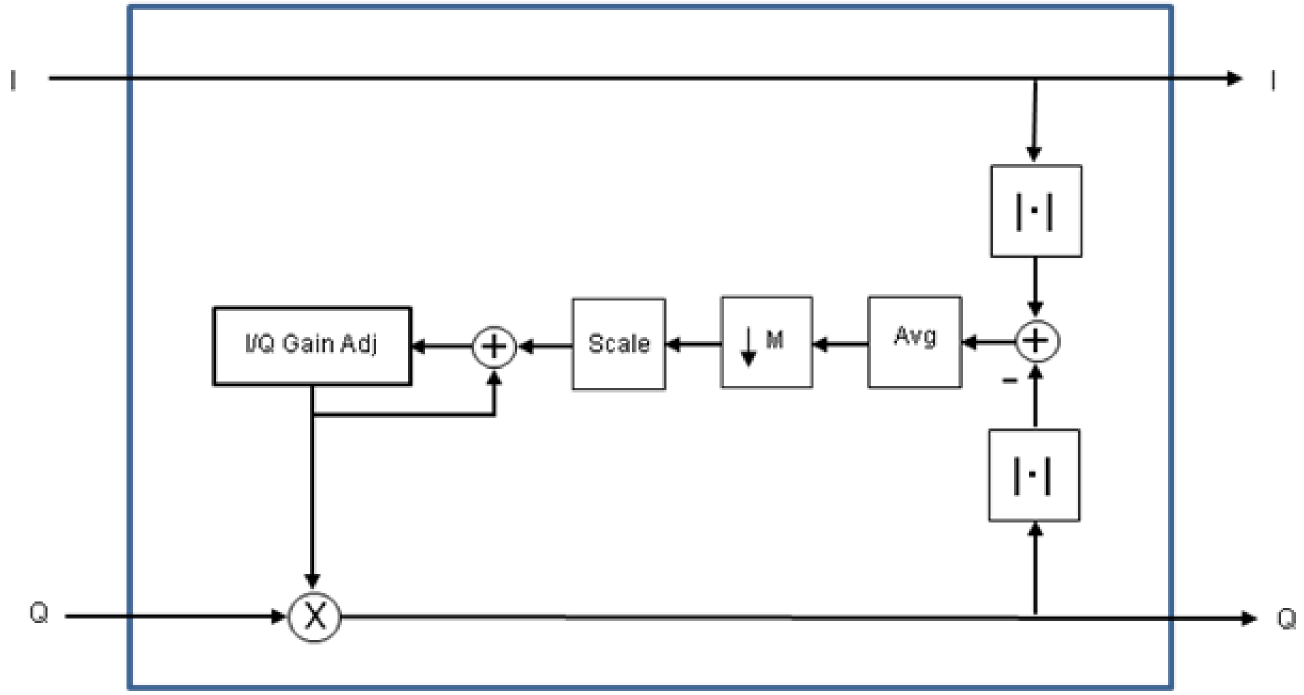


Figure 48-12. I/Q Gain Mismatch Correction Block Diagram

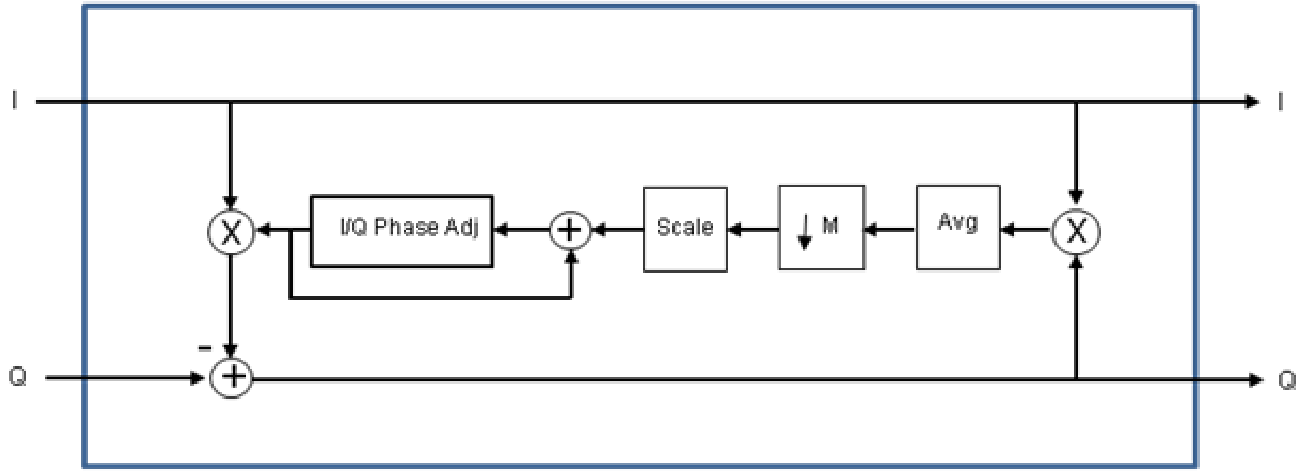


Figure 48-13. I/Q Phase Mismatch Correction Block Diagram

Calibration is enabled by setting the iqmc_cal_en bit. The following values should be used for calibration: iqmc_num_iter=0x80, and rx_dec_filt_osr=2 (OSR of 8). The IQMC_CAL register should be set to its default value 0x0000_0400. A CW input with a 250 kHz (+/-75 kHz) offset and -2 dBm (+/- 6 dBm) level at the ADC I channel input is required for calibration, though the calibration should also work with lower signal levels down to -18 dBm. With the standard number of iterations (iqmc_num_iter=0x80), calibration takes approximately 2 ms. After calibration is completed the IQMC_CAL register is updated and the iqmc_cal_en bit is cleared automatically.

When the receiver is enabled for packet reception, the calibration loops are disabled and the corrections are static. For gain correction, the Q channel is scaled by the gain adjustment coefficient. For phase correction, the I channel is scaled by the phase adjustment coefficient and subtracted from the Q channel. The gain and phase correction values are stored in programming model registers and may be stored in flash to eliminate the need to run I/Q mismatch calibration after each reset.

There is also a separate gain adjusment coefficient, programmed via XCVR_ANA_SPARE[IQMC_DC_GAIN_ADJ], which is used only during DCOC calibration; that is, during the period after the TSM dcoc_en is asserted but before the TSM rx_dig_en is asserted. The applied phase adjustment coefficient is 0 during this time. The XCVR_RX_ANA_CTRL[IQMC_DC_GAIN_ADJ_EN] bit must also be set for this coefficient to be used in favor of iqmc_gain_adj. There is no hardware support to automatically calculate the iqmc_dc_gain_adj term.

48.4.8 Normalizer

The normalizer block converts the 12-bit I/Q input signal into a normalized 5-bit I/Q output and a 5-bit phase output as shown in the block diagram below.

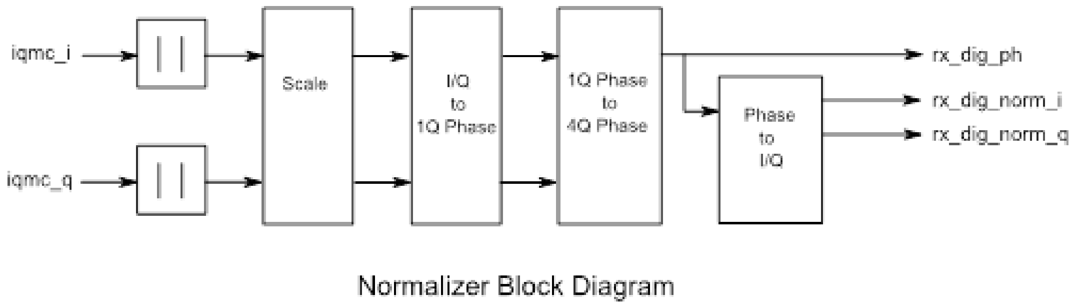


Figure 48-14. Normalizer Block Diagram

The absolute value of each component of the I/Q input is taken and shifted by a common factor to yield the largest 5-bit values which have a zero MSB for both I and Q. This is followed by rounding that yields I and Q values between 0 and 16 decimal. The scaled I/Q values are used as inputs to a one quadrant phase lookup table. The one quadrant table output is converted to a 4 quadrant 5-bit phase value based on the input sign bits. The 5-bit I/Q output is generated by another lookup table. The normalizer has a maximum quantization error of 8.9 degrees and a RMS error of 3.5 degrees for a random input.

48.4.9 Interpolator

The interpolator performs an upsample by 2 operation from 8 MHz to 16 MHz which is required for the FSK demodulator. The interpolator has a 1dB bandwidth of 900 kHz with a full band frequency response shown in the figure below. The interpolation filter has coefficients h[n] = {1, 3, 4, 4, 3, 1}.

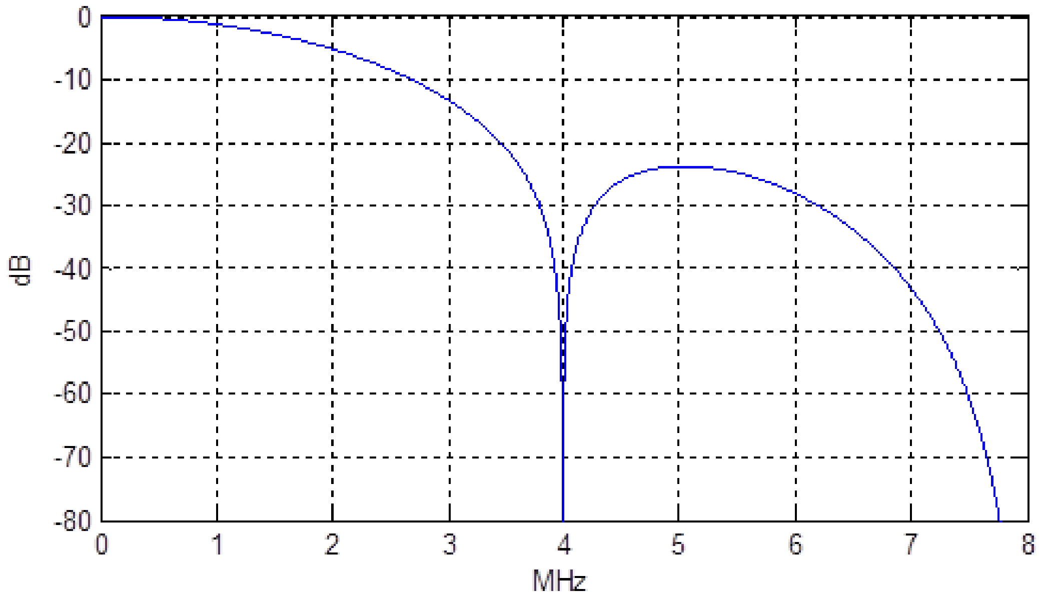


Figure 48-15. Interpolation Filter Response

Chapter 49
TX Digital

49.1 About this module

49.1.1 Introduction

The TX Digital is a module that interfaces to the various supported RF protocols and presents their digital transmission data in a consistent modulated manner to the PLL Digital module.

The TX Digital also includes various DFT modulation options to support validation and testing.

49.1.2 Features

The TX Digital module includes the following features:

Support for various RF Protocols:

- Bluetooth Low Energy
- IEEE 802.15.4

Supports Data Padding for Power Amplifier Ramp-Up and Ramp-Down.

49.1.3 Block diagram

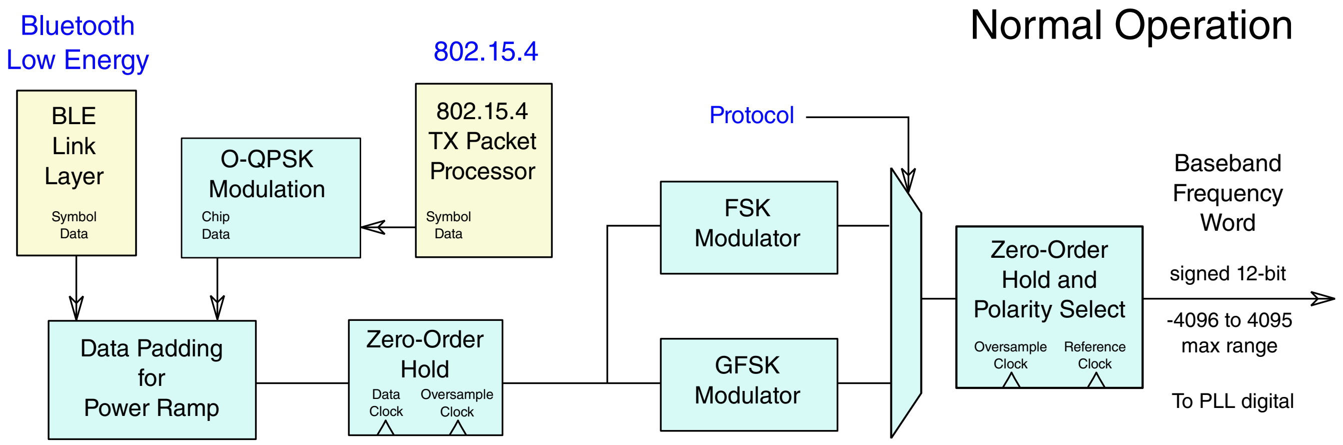


Figure 49-1. TX Digital Block diagram

49.2 Functional Description

The TX Digital is the source of all modulation that is applied to the PLL.

This modulation can be from the GFSK, FSK, or DFT Tone Modulators, it can be pseudo-randomized using the DFT LFSR, or it can be a user specified DFT pattern of Symbols or Chips presented in a wide range of modulation frequencies.

The GFSK modulator can be configured using Modulation Index, Symbol Rate, Reference Clock frequency, and a manual override of the built-in Gaussian filter coefficients.

The FSK modulation level can be directly controlled with a frequency resolution equal to the PLL Frequency Word LSB size (244 Hz for a 32 MHz reference clock).

The Tone Modulator has four discrete modulation levels with a resolution equal to the PLL Frequency Word LSB size; these levels can be selected by software, or be pseudo-randomized using the DFT LFSR.

The DFT LFSR can also generate pseudo-random Symbols for 802.15.4 Symbol to Chip mapping, or the Chips can be generated by the LFSR.

The following sections will discuss these topics in more detail.

49.2.1 Baseband Frequency Word

The TX Digital module supplies the PLL Digital module with a signed 12-bit frequency word containing the integer representation of the desired modulation frequency.

The resulting modulation word can be in the range -4096 to 4095; for a PLL Frequency Word LSB size of 244.14 Hz this gives a maximum +/- 1000 kHz for the modulation frequency range.

The table below shows the PLL Frequency Word LSB size (in Hz) for various Reference Frequencies

Reference Frequency (MHz)	32	36	40	26
PLL LSB (Hz)	244.14	274.66	305.18	198.36

49.2.2 Transmit Power Ramp and Data Padding

The Data Padding module provides a data padding method to support gradually ramping up the Power Amplifier (PA) prior to transmitting RF protocol data, and a data hold method to support gradually ramping down the power after the RF protocol data has been transmitted. This module supports the various RF protocol requirements for power ramping so as not to produce any RF interference that would otherwise corrupt the RF spectrum.

49.2.2.1 Data Padding

Data Padding is done to avoid abrupt steps in frequency modulation and thereby minimize spectral transients during the transition from PA ramping to packet transmission.

Data Padding avoids an abrupt step in modulation during the PA power ramp-up by pre-pending symbols of preamble-like modulation onto the front end of the actual preamble transmission. The PA power ramp-up is done while these additional symbols are being modulated to the PLL, and then after the power ramp-up completes, the modulation is smoothly transitioned into the RF protocol packet transmission at the Target Power level.

This module supports data padding for the various RF protocols and their required packet structures and preambles. For example, the BLE preamble is an alternating 1-0-1-0-1-0-1-0 or 0-1-0-1-0-1-0-1 pattern, depending on the type of packet which is to be transmitted. Data padding patterns are stored in the Data Padding Pattern registers.

The Data Padding Pattern register has two 8-bit register fields to allow for two padding choices, and the chosen bits are modulated onto the PA ramp starting with the lsb of each padding register field.

If the BLE link layer is used by the RF protocol, then the required pattern register is automatically selected. In other cases the DP_SEL bit should be used to select the data padding pattern.

The TSM adjusts the starting time of the TX Digital to account for the length of the power ramp that is selected. In this way the Radio goes On Air at the required time, with a fully ramped Power Amplifier.

At the end of the RF protocol packet transmission, the ending symbol is held during the PA power ramp-down to again ensure there is no abrupt change in modulation power that would cause a spectral transient.

49.2.3 Zero-Order Hold and Polarity Select

The TX Digital has two zero-order hold (ZOH) stages. The first one is used to convert the data bit from the data clock domain to the oversample clock domain.

The second zero-order hold and polarity select stage converts the frequency word from the oversample clock domain to the reference clock domain. There is a choice for the reference clock polarity to use to capture the frequency word {even or odd edge of the reference clock}.

A summary of the ZOH Data Oversample Rates are summarized below for BLE and 802.15.4.

Reference Frequency (MHz)	Clock Division	Oversample Clock (MHz)	BLE Symbol Rate (MHz)	Oversample Rate
32	4	8	1	8
36	4	9	1	9
40	4	10	1	10
26	2	13	1	13

Reference Frequency (MHz)	Clock Division	Oversample Clock (MHz)	802.15.4 Symbol Rate (MHz)	Oversample Rate
32	2	16	2	8
36	2	18	2	9
40	2	20	2	10
26	1	26	2	13

49.3 Memory Map and Register Definition

The TX Digital module memory map and detailed descriptions of all its registers is included in Tranceiver(XCVR) register section.

Chapter 50
Transceiver Sequence Manager (TSM)

50.1 About this module

50.1.1 Introduction

This Block Guide describes the Transceiver Sequence Manager (TSM).

The TSM is a fully-programmable, multi-protocol transceiver sequence manager, which supports Bluetooth Low Energy, 802.15.4, and other 2.4GHz transceivers.

OVERVIEW:

The Transceiver Sequence Manager controls the warmup and warmdown processes for all radio sequences. The TSM provides for a TX sequence, and an RX sequence. For both TX and RX, a warmup and a warmdown sequence is supported. The length of each sequence is programmable, from 1 – 254us. The resolution of the TSM is 1us. Controls for all analog and digital transceiver blocks are provided. The TSM has 56 outputs with which to control the warmup and warmdown processes. Each TSM output enables, or otherwise provides control for, a transceiver-related block. Each TSM output (or group of outputs) has 4 8-bit registers, with which to control the point at which the output asserts during the warmup (1 register for TX, 1 for RX), and the point at which it deasserts (1 register for TX, 1 for RX). Some TSM outputs, which are known in advance to require identical timing, are “ganged together”, to reduce area and required programming. For DFT, and non-mission-mode validation, each TSM output can be put under direct software control, using register overrides. Any sequence can be optionally aborted by software, or by a PLL unlock condition. And, any sequence can be temporarily halted, by setting an optional, programmable breakpoint.

50.1.2 Features

The TSM includes the following features:

- Supports 56 timing-controlled outputs
- 44 of the 56 outputs have fully programmable timing
- Supports TX and RX sequences
- Supports multiple protocols
- Handles software-driven as well as PLL unlock sequence aborts.

50.1.3 Block diagram

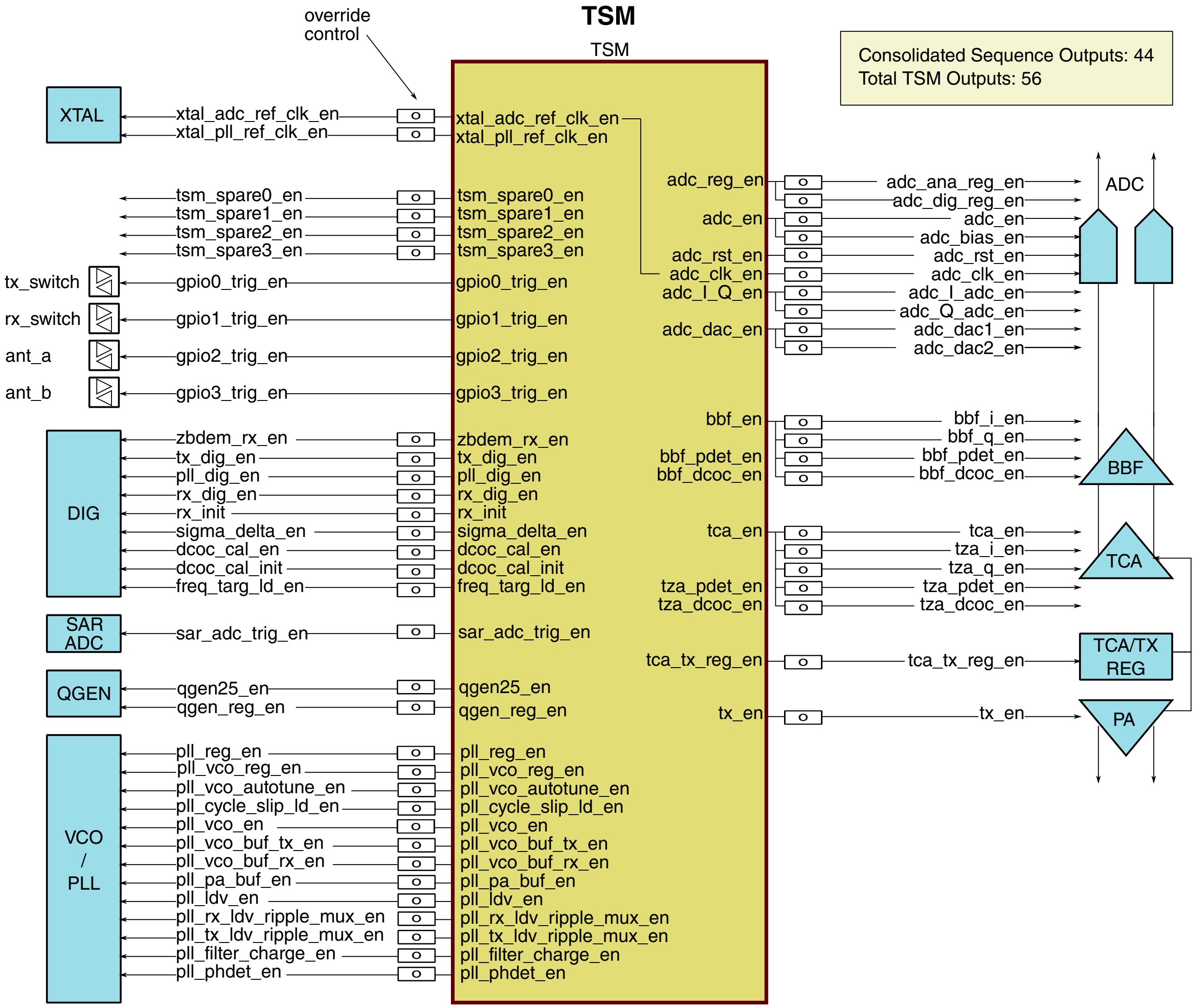


Figure 50-1. Block diagram

50.2 Signals

Signal	I/O	Function
ipg_hard_async_reset_b	I	global async active-low reset
ipg_clk	I	Module IP bus Clock
tsm_clk	I	1mhz clock from CRM divider
tsm_dly_clk	I	same, but delayed by 1 reference clock (ipg_clk) period
tsm_4m_clk	I	4mhz for PA Ramp
signal00_tx_hi[7:0] … signal43_tx_hi[7:0]	I	TX HI time control registers
signal00_tx_lo[7:0] … signal43_tx_lo[7:0] I	I	TX LO time control registers
signal00_rx_hi[7:0] … signal43_rx_hi[7:0]	I	RX HI time control registers
signal00_rx_lo[7:0] … signal43_rx_lo[7:0]	I	RX LO time control registers
end_of_tx_wu[7:0]	I	register END TX warmup
end_of_tx_wd[7:0]	I	register END TX warmdn
end_of_rx_wu[7:0]	I	register END RX warmup
end_of_rx_wd[7:0]	I	register END RX warmdn
bkpt[7:0]	I	register BREAKPOINT
tsm_ovrd_en[53:0]	I	register tsm ovrd enables
tsm_ovrd[53:0]	I	register tsm ovrds
btle_tx_en	I	btle tx enable
btle_rx_en	I	btle rx enable
zigbee_tx_en	I	zigbee tx enable
zigbee_rx_en	I	zigbee rx enable
other_tx_en	I	other tx enable
other_rx_en	I	other rx enable
force_tx_en	I	register mcu force tx enable
force_rx_en	I	register mcu force rx enable
ext_tx_en	I	from gpio tx enable (not used in Apache)
ext_rx_en	I	from gpio rx enable (not used in Apache)
rx_abort_dis	I	register dont abort rx operations
tx_abort_dis	I	register dont abort tx operations

Table continues on the next page...

Signal	I/O	Function
pll_ld_failed_any	I	at least 1 pll unlock abort asserted
pa_ramp_sel[1:0]	I	register duration of ramp
pa_power[3:0]	I	default transmit power from xcvr
pa_bias0[3:0]	I	pa bias table
pa_bias1[3:0]	I	pa bias table
pa_bias2[3:0]	I	pa bias table
pa_bias3[3:0]	I	pa bias table
pa_bias4[3:0]	I	pa bias table
pa_bias5[3:0]	I	pa bias table
pa_bias6[3:0]	I	pa bias table
pa_bias7[3:0]	I	pa bias table
tgt_pwr_src[1:0]	I	select target power source (register)
btle_power[3:0]	I	btle transmit power
pa_pwr_zigbee[3:0]	I	zigbee transmit power
protocol[2:0]	I	protocol (BTLE, ZIGBEE, ANT, etc.) from register
tsm_recycle[1:0]	I	recycle command from ZSM
recycle_count0[7:0]	I	tsm recycle count for RX_CYC state (register)
recycle_count1[7:0]	I	tsm recycle count for RX_PAN1 state (register)
lpps_enable	I	zigbee LPPS mode master enable (register)
lpps_lp_en	I	zigbee LPPS mode timing control (zbdemod)
lpps_allow[7:1]	I	zigbee LPPS mode block enables (register)
tsm_clk_en	O	module clock enable (crm)
pll_reg_en	O	tsm control output
pll_vco_reg_en	O	tsm control output
qgen_reg_en	O	tsm control output
tca_tx_reg_en	O	tsm control output
adc_ana_reg_en	O	tsm control output
adc_dig_reg_en	O	tsm control output
xtal_pll_ref_clk_en	O	tsm control output
xtal_adc_ref_clk_en	O	tsm control output
pll_vco_autotune_en	O	tsm control output
pll_cycle_slip_ld_en	O	tsm control output
pll_vco_en	O	tsm control output
pll_vco_buf_rx_en	O	tsm control output
pll_vco_buf_tx_en	O	tsm control output
pll_pa_buf_en	O	tsm control output
pll_ldv_en	O	tsm control output
pll_rx_ldv_ripple_mux_en	O	tsm control output
pll_tx_ldv_ripple_mux_en	O	tsm control output
pll_filter_charge_en	O	tsm control output
pll_phdet_en	O	tsm control output

Table continues on the next page...

Signal	I/O	Function
qgen25_en	O	tsm control output
tx_en	O	tsm control output
adc_en	O	tsm control output
adc_bias_en	O	tsm control output
adc_clk_en	O	tsm control output
adc_I_adc_en	O	tsm control output
adc_Q_adc_en	O	tsm control output
adc_dac1_en	O	tsm control output
adc_dac2_en	O	tsm control output
adc_rst_en	O	tsm control output
bbf_i_en	O	tsm control output
bbf_q_en	O	tsm control output
bbf_pdet_en	O	tsm control output
bbf_dcoc_en	O	tsm control output
tca_en	O	tsm control output
tza_i_en	O	tsm control output
tza_q_en	O	tsm control output
tza_pdet_en	O	tsm control output
tza_dcoc_en	O	tsm control output
pll_dig_en	O	tsm control output
tx_dig_en	O	tsm control output
rx_dig_en	O	tsm control output
rx_init	O	tsm control output
sigma_delta_en	O	tsm control output
zbdem_rx_en	O	tsm control output
dcoc_en	O	tsm control output
dcoc_init	O	tsm control output
freq_targ_ld_en	O	tsm control output
sar_adc_trig_en	O	tsm control output
tsm_spare0_en	O	tsm control output
tsm_spare1_en	O	tsm control output
tsm_spare2_en	O	tsm control output
tsm_spare3_en	O	tsm control output
gpio0_trig_en	O	tsm control output
gpio1_trig_en	O	tsm control output
gpio2_trig_en	O	tsm control output
gpio3_trig_en	O	tsm control output
tx_mode	O	tx sequence indicator
rx_mode	O	rx sequence indicator
tx_pwr_cntl[3:0]	O	transmit power to pa
pll_abort	O	for zigbee

Table continues on the next page...

Signal	I/O	Function
tsm_wu_complete	O	for zigbee
tsm_idle	O	for zigbee
tsm_count[7:0]	O	for registers and dtest
abort_taken	O	abort trigger to set TSM_ABORT_FLAG
tsm_rx_en	O	for dtest

50.3 Memory Map and register definition

Field	R/W	Description
TSM_TIMING00_TX_HI[7:0] … TSM_TIMING43_TX_HI[7:0]	RW	TX HI timing control register. Sets the point in the TSM TX warmup where the TSM signal controlled by TSM_TIMINGxx will assert. See Section Timing Registers of this Block Guide for a mapping of TSM_TIMINGxx to TSM control outputs.
TSM_TIMING00_TX_LO[7:0] … TSM_TIMING43_TX_LO[7:0]	RW	TX LO timing control register. Sets the point in the TSM TX warmup where the TSM signal controlled by TSM_TIMINGxx will deassert. See Section Timing Registers of this Block Guide for a mapping of TSM_TIMINGxx to TSM control outputs.
TSM_TIMING00_RX_HI[7:0] … TSM_TIMING43_RX_HI[7:0]	RW	RX HI timing control register. Sets the point in the TSM RX warmup where the TSM signal controlled by TSM_TIMINGxx will assert. See Section Timing Registers of this Block Guide for a mapping of TSM_TIMINGxx to TSM control outputs.
TSM_TIMING00_RX_LO[7:0] … TSM_TIMING43_RX_LO[7:0]	RW	RX LO timing control register. Sets the point in the TSM RX warmup where the TSM signal controlled by TSM_TIMINGxx will deassert. See Section Timing Registers of this Block Guide for a mapping of TSM_TIMINGxx to TSM control outputs.
END_OF_TX_WU[7:0]	RW	END TX warmup register. When the TSM counter reaches this value, TSM transitions from the WARMUP phase to the ON phase and holds its count.
END_OF_TX_WD[7:0]	RW	END TX warmdn register. This is the last count of the WARMDOWN phase of the TSM counter. TSM will return to IDLE on the next clock.
END_OF_RX_WU[7:0]	RW	END RX warmup register. When the TSM counter reaches this value, TSM transitions from the WARMUP phase to the ON phase and holds its count.
END_OF_RX_WD[7:0]	RW	END RX warmdn register. This is the last count of the WARMDOWN phase of the TSM counter. TSM will return to IDLE on the next clock.
BKPT[7:0]	RW	BREAKPOINT control register. Breakpoint can be used to temporarily suspend a TSM sequence. During a TSM sequence, when the TSM counter matches the BKPT register value, counting stops and the counter holds in its current state. The breakpoint can be lifted by modifying the BKPT[7:0] register. Once the breakpoint is lifted, the TSM proceeds in the phase it was in prior to the breakpoint match, and the TSM counter begins incrementing again from the point at which the breakpoint occurred
TSM_OVRD_EN[53:0]	RW	TSM override enable bits, once for each of the TSM control outputs. See Section Overrides of this Block Guide.

Table continues on the next page...

<table>
<tr><th>Field</th><th>R/W</th><th>Description</th></tr>
<tr><td>TSM_OVRD[53:0]</td><td>RW</td><td>TSM override value bits, once for each of the TSM control outputs. See Section Overrides of this Block Guide.</td></tr>
<tr><td>TX_ABORT_DIS</td><td>RW</td><td>RX Abort disable. When set, prevents PLL unlock events during TX sequences from aborting the sequence.</td></tr>
<tr><td>RX_ABORT_DIS</td><td>RW</td><td>RX Abort disable. When set, prevents PLL unlock events during RX sequences from aborting the sequence.</td></tr>
<tr><td>ABORT_ON_FREQ_TARG</td><td>RW</td><td>1: allow TSM abort on Frequency Target Unlock Detect
0: don’t allow TSM abort on Frequency Target Unlock Detect</td></tr>
<tr><td>ABORT_ON_CYCLE_SLIP</td><td>RW</td><td>1: allow TSM abort on Cycle Slip Unlock Detect
0: don’t allow TSM abort on Cycle Slip Unlock Detect</td></tr>
<tr><td>ABORT_ON_CTUNE</td><td>RW</td><td>1: allow TSM abort on Coarse Tune Unlock Detect
0: don’t allow TSM abort on Coarse Tune Unlock Detect</td></tr>
<tr><td>PA_RAMP_SEL[1:0]</td><td>RW</td><td>PA_RAMP_SEL bits control duration of PA ramp and ramp rate:
<table>
<tr><th>PA_RAMP_SEL[1:0]</th><th>TOTAL RAMP DURATION</th><th>DURATION OF EACH RAMP STEP</th></tr>
<tr><td>00</td><td>No ramp</td><td>No ramp</td></tr>
<tr><td>01</td><td>2us</td><td>0.25us</td></tr>
<tr><td>10</td><td>4us</td><td>0.5us</td></tr>
<tr><td>11</td><td>8us</td><td>1us</td></tr>
</table></td></tr>
<tr><td>PA_POWER[3:0]</td><td>RW</td><td>PA Target Power. This contents of this register are used as PA target power when TGT_PWR_SRC[1:0] = 00.</td></tr>
<tr><td>PA_BIAS0[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, when TSM tx_en transitions low to high, and then for the duration of the first ramp step. During PA ramp down, the contents of this register are the PA power value during the final ramp step. In both cases, PA_BIAS0 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS1[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the second ramp step. During PA ramp down, the contents of this register are the PA power value during the second-to-last ramp step. In both cases, PA_BIAS1 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS2[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the third ramp step. During PA ramp down, the contents of this register are the PA power value during the third-to-last ramp step. In both cases, PA_BIAS2 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS3[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the fourth ramp step. During PA ramp down, the contents of this register are the PA power value during the fourth-to-last ramp step. In both cases, PA_BIAS3 cannot exceed target power (enforced by PA ramping logic).</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>R/W</th><th>Description</th></tr>
<tr><td>PA_BIAS4[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the fifth ramp step. During PA ramp down, the contents of this register are the PA power value during the fifth-to-last ramp step. In both cases, PA_BIAS4 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS5[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the sixth ramp step. During PA ramp down, the contents of this register are the PA power value during the sixth-to-last ramp step. In both cases, PA_BIAS5 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS6[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the seventh ramp step. During PA ramp down, the contents of this register are the PA power value during the seventh-to-last ramp step. In both cases, PA_BIAS6 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>PA_BIAS7[3:0]</td><td>RW</td><td>If PA ramping is enabled (PA_RAMP_SEL > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the eighth (final) ramp step. During PA ramp down, the contents of this register are the PA power value during the eighth-to-last (first) ramp step. In both cases, PA_BIAS7 cannot exceed target power (enforced by PA ramping logic).</td></tr>
<tr><td>TGT_PWR_SRC[1:0]</td><td>RW</td><td>For determining transmit power, the TGT_PWR_SRC[1:0] bits control target power selection, according to the following table.
<table>
<tr><th>TGT_PWR_SRC[1:0]</th><th>TARGET POWER SOURCE</th></tr>
<tr><td>00</td><td>PA_POWER[3:0] register (XCVR space)</td></tr>
<tr><td>01</td><td>BTLE Link Layer</td></tr>
<tr><td>10</td><td>Zigbee Link Layer (PA_PWR[3:0] register in ZIGBEE space)</td></tr>
<tr><td>11</td><td>PROTOCOL[2:0] bits select target power source</td></tr>
</table></td></tr>
<tr><td>PROTOCOL[2:0]</td><td>RW</td><td>These bits select the current protocol in effect for the entire transceiver, according to the following table.
<table>
<tr><th>PROTOCOL[2:0]</th><th>Selection</th></tr>
<tr><td>000</td><td>Bluetooth Low Energy</td></tr>
<tr><td>001</td><td>BLE in MBAN</td></tr>
<tr><td>010</td><td>BLE overlap MBAN</td></tr>
<tr><td>011</td><td>ANT</td></tr>
<tr><td>100</td><td>Zigbee</td></tr>
<tr><td>101</td><td>802.15.4j</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

Field	R/W	Description
RECYCLE_COUNT0[7:0]	RW	The RECYCLE_COUNT0[7:0] register determines the TSM count value to which the TSM “recycles” when the Zigbee Sequence Manager (ZSM) state “RX_CYC” is reached and the ZSM asserts tsm_recycle[0] to TSM. This register also determines the TSM count value to which the TSM “recycles” when the ZSM state RX_CCCA is reached because tsm_recycle[0] is also asserted in this state. The intention is for this register to be programmed to a TSM count value such that the TSM re-asserts its “rx_init” output, but there are no restrictions on programming this register. See the ZSM Sequence Manager Block Guide Section ZSM/TSM Interaction for more details.
RECYCLE_COUNT1[7:0]	RW	The RECYCLE_COUNT1[7:0] register determines the TSM count value to which the TSM “recycles” when the Zigbee Sequence Manager (ZSM) state “RX_PAN1” is reached and the ZSM asserts tsm_recycle[1] to TSM. The intention is for this register to be programmed to a TSM count value such that the TSM de-asserts, and then re-asserts its “pll_dig_en” output, to effectuate a Dual PAN on-the-fly channel change, but there are no restrictions on programming this register. See the ZSM Sequence Manager Block Guide Section ZSM/TSM Interaction for more details.
LPPS_ENABLE	RW	Master enable for LPPS mode. Allows Zigbee correlators to be duty-cycled during Preamble Search, and selected RF/Analog blocks to be duty-cycled simultaneously. See Section Zigbee Low Power Preamble Search (LPPS), of this Block Guide. 1: LPPS mode enable 0: LPPS mode disabled
LPPS_QGEN25_ALLOW	RW	1: Allow TSM output **qgen25_en** to be duty-cycled during LPPS 0: Disallow TSM output **qgen25_en** to be duty-cycled during LPPS
LPPS_ADC_ALLOW	RW	1: Allow ADC-related TSM outputs {**adc_en**, **adc_bias_en**} to be duty-cycled during LPPS. 0: Disallow ADC-related TSM outputs {**adc_en**, **adc_bias_en**} to be duty-cycled during LPPS.
LPPS_ADC_CLK_ALLOW	RW	1: Allow ADC_CLK-related TSM outputs {**xtal_adc_ref_clk_en**, **adc_clk_en**} to be duty-cycled during LPPS. 0: Disallow ADC-related TSM outputs {**xtal_adc_ref_clk_en**, **adc_clk_en**} to be duty-cycled during LPPS.
LPPS_ADC_I_Q_ALLOW	RW	1: Allow ADC_I/Q-related TSM outputs {**adc_I_adc_en**, **adc_Q_adc_en**} to be duty-cycled during LPPS. 0: Disallow ADC_I/Q-related TSM outputs {**adc_I_adc_en**, **adc_Q_adc_en**} to be duty-cycled during LPPS.
LPPS_ADC_DAC_ALLOW	RW	1: Allow ADC_DAC-related TSM outputs {**adc_dac1_en**, **adc_dac2_en**} to be duty-cycled during LPPS. 0: Disallow ADC_DAC-related TSM outputs {**adc_dac1_en**, **adc_dac2_en**} to be duty-cycled during LPPS.
LPPS_BBF_ALLOW	RW	1: Allow BBF-related TSM outputs {**bbf_i_en**, **bbf_q_en**, **bbf_pdet_en**, **bbf_dcoc_en**} to be duty-cycled during LPPS. 0: Disallow BBF-related TSM outputs {**bbf_i_en**, **bbf_q_en**, **bbf_pdet_en**, **bbf_dcoc_en**} to be duty-cycled during LPPS.
LPPS_TCA_ALLOW	RW	1: Allow TCA-related TSM outputs {**tca_en**, **tza_i_en**, **tza_q_en**, **tza_pdet_en**, **tza_dcoc_en**} to be duty-cycled during LPPS.

Table continues on the next page...

Field	R/W	Description
		0: Disallow TCA-related TSM outputs {**tca_en**, **tza_i_en**, **tza_q_en**, **tza_pdet_en**, **tza_dcoc_en**} to be duty-cycled during LPPS.
TSM_ABORT_FLAG	W1C	Flag that indicates that the TSM has taken an abort due to an enabled PLL unlock condition, since the last time this flag was cleared. Write a ‘1’ to this bit to clear the flag. Writing ‘0’ to the bit, or reading the bit, has no effect on the state of the bit.
TSM_COUNT[7:0]	RO	Reflects the instantaneous value of the TSM counter. Read-only.

50.4 Functional description

50.4.1 Sequence Counter

The TSM supports 1 TX and 1 RX sequence. Each sequence consists of 3 phases:

1. WARMUP phase
2. ON phase
3. WARMDOWN phase

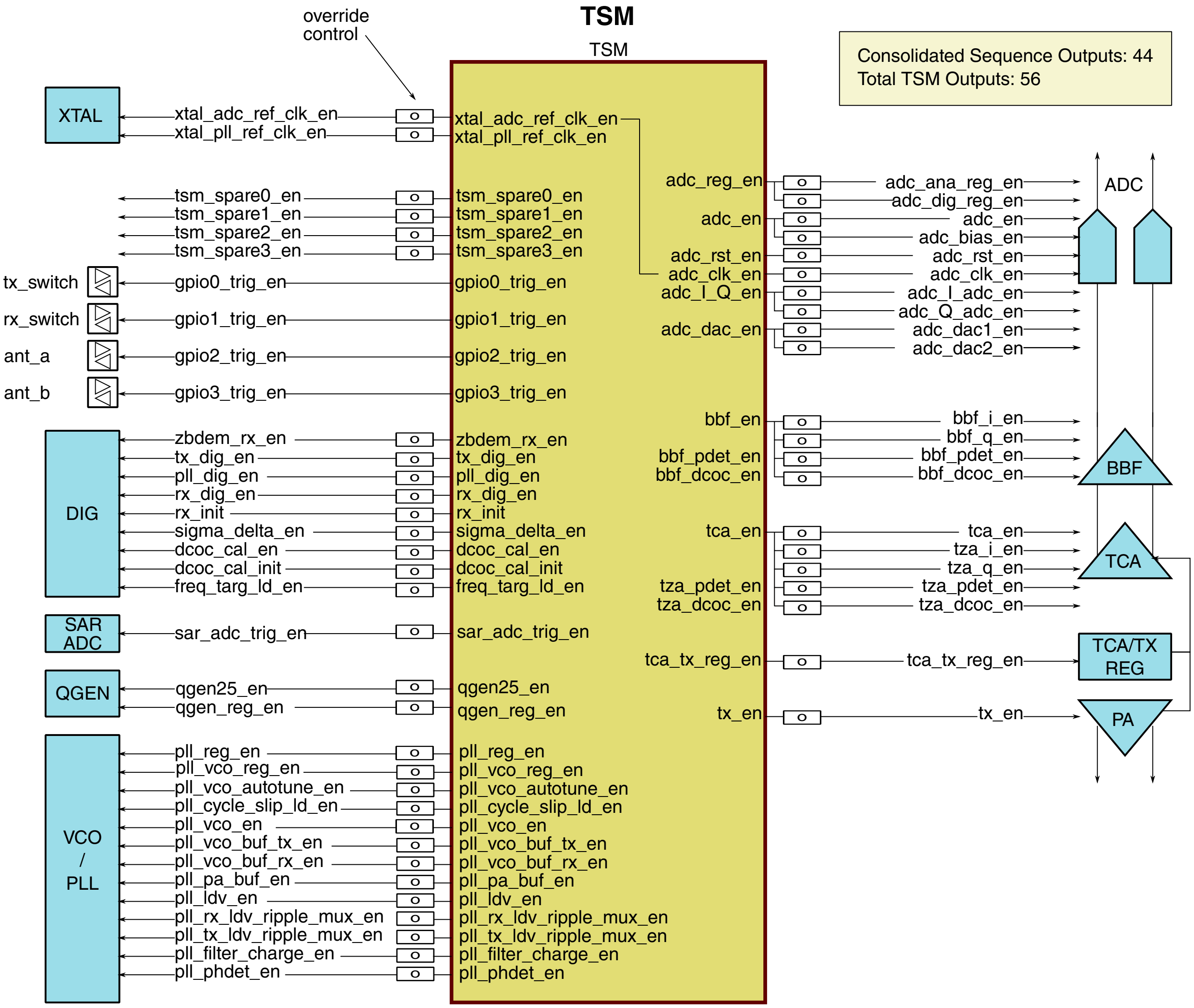
The central element of the TSM is an 8-bit counter. The counter is held at 0 during the idle state. From idle state, any sequence can be launched, by an initiating event. (See Section Sequence Initiation of this Block Guide.) At an initiating event, the TSM counter counts up from 0 to a programmed stop point during the WARMUP phase, determined by the END_OF_TX_WU or END_OF_RX_WU register, depending on whether the sequence is TX or RX. At the end-of-warmup “stop point”, the TSM counter holds its count, and the TSM sequence enters the ON phase. The TSM counter will remain in the ON phase, and hold its count, until the initiating event deasserts, or an abort occurs (See Section Sequence Termination of this Block Guide). When either of these conditions occurs during the ON phase, the TSM will resume counting from the point it was holding during the ON phase, and the sequence will enter the WARMDOWN phase. The counter will continue counting until a programmed stop point is reached. This stop point is determined by the END_OF_TX_WD or END_OF_RX_WD register, depending on whether the sequence is TX or RX. Once this point is reached, the sequence returns to idle, and the TSM counter returns to 0.

The 4 8-bit registers which control the duration of the WARMUP phase (END_OF_TX_WU[7:0] and END_OF_RX_WU[7:0]), and the duration of the WARMDOWN phase (END_OF_TX_WD[7:0] and END_OF_RX_WD[7:0]), reside in the END_OF_SEQ register.

All TSM-controlled outputs are active-high. Any TSM-controlled output can be asserted once during a sequence (either TX or RX sequence), and then deasserted once. Or, the output can be held in a deasserted state for the duration of the sequence. The precise timing for the assertion and deassertion of each TSM-controlled output, for both TX and RX sequences, is determined by 4 8-bit registers assigned to that output (see Section TSM-Controlled Outputs of this Block Guide). The TSM resolution is 1us. This is the update rate for the TSM counter, and sets the granularity with which warmup and warmdown processes can be controlled.

50.4.2 TSM-controlled outputs

.The TSM has 56 timing-controlled outputs, for enabling and control of the various transceiver blocks. The outputs are shown in the following diagram.



In some cases, some of the 56 outputs are grouped together, for outputs which need identical timing. For example, tza_i_en and tza_q_en. These groupings save area, because only 1 set of timing registers is needed to control timing for the entire group. Groupings also reduce the amount of required TSM programming. In all cases, even for grouped outputs, each output has an individual override control, to allow software to take command of any of the 56 TSM-controlled outputs at any time. As a consequence of the groupings, there are only 44 sets of timing registers, to control the 56 outputs. The table below, lists the 56 TSM-controlled outputs, and for the grouped outputs, indicates the signal which controls the timing for the entire group.

SIGNAL INDEX	56 TSM-CONTROLLED OUTPUTS	SIGNAL CONTROLLING GROUP TIMING
0	pll_reg_en	pll_reg_en

Table continues on the next page...

SIGNAL INDEX	56 TSM-CONTROLLED OUTPUTS	SIGNAL CONTROLLING GROUP TIMING
1	pll_vco_reg_en	pll_vco_reg_en
2	qgen_reg_en	qgen_reg_en
3	tca_tx_reg_en	tca_tx_reg_en
4	adc_ana_reg_en	adc_reg_en
5	adc_dig_reg_en	adc_reg_en
6	xtal_pll_ref_clk_en	pll_ref_clk_en
7	xtal_adc_ref_clk_en	adc_clk_en
8	pll_vco_autotune_en	pll_vco_autotune_en
9	pll_cycle_slip_ld_en	pll_cycle_slip_ld_en
10	pll_vco_en	pll_vco_en
11	pll_vco_buf_rx_en	pll_vco_buf_rx_en
12	pll_vco_buf_tx_en	pll_vco_buf_tx_en
13	pll_pa_buf_en	pll_pa_buf_en
14	pll_ldv_en	pll_ldv_en
15	pll_rx_ldv_ripple_mux_en	pll_rx_ldv_ripple_mux_en
16	pll_tx_ldv_ripple_mux_en	pll_tx_ldv_ripple_mux_en
17	pll_filter_charge_en	pll_filter_charge_en
18	pll_phdet_en	pll_phdet_en
19	qgen25_en	qgen25_en
20	tx_en	tx_en
21	adc_en	adc_en
22	adc_bias_en	adc_en
23	adc_clk_en	adc_clk_en
24	adc_I_adc_en	adc_I_Q_en
25	adc_Q_adc_en	adc_I_Q_en
26	adc_dac1_en	adc_dac_en
27	adc_dac2_en	adc_dac_en
28	adc_rst_en	adc_rst_en
29	bbf_i_en	bbf_en
30	bbf_q_en	bbf_en
31	bbf_pdet_en	bbf_en
32	bbf_dcoc_en	bbf_en
33	tca_en	tca_en
34	tza_i_en	tca_en
35	tza_q_en	tca_en
36	tza_pdet_en	tca_en
37	tza_dcoc_en	tca_en
38	pll_dig_en	pll_dig_en
39	tx_dig_en	tx_dig_en
40	rx_dig_en	rx_dig_en

Table continues on the next page...

SIGNAL INDEX	56 TSM-CONTROLLED OUTPUTS	SIGNAL CONTROLLING GROUP TIMING
41	rx_init	rx_init
42	sigma_delta_en	sigma_delta_en
43	zbdem_rx_en	zbdem_rx_en
44	dcoc_en	dcoc_en
45	dcoc_init	dcoc_init
46	freq_targ_ld_en	freq_targ_ld_en
47	sar_adc_trig_en	sar_adc_trig_en
48	tsm_spare0_en	tsm_spare0_en
49	tsm_spare1_en	tsm_spare1_en
50	tsm_spare2_en	tsm_spare2_en
51	tsm_spare3_en	tsm_spare3_en
52	gpio0_trig_en	gpio0_trig_en
53	gpio1_trig_en	gpio1_trig_en
54	gpio2_trig_en	gpio2_trig_en
55	gpio3_trig_en	gpio3_trig_en

For example, TSM-controlled ouputs tca_en, tza_i_en, and tza_q_en, are all members of the same group. The timing registers to program tca_en (TCA_EN_TX_HI, TCA_EN_TX_LO, TCA_EN_RX_HI, TCA_EN_RX_LO), control the timing for all members of the group. Each member of the group, however, has in independent software override.

During the WARMUP and WARMDOWN phases of any sequence, any TSM-controlled output can be programmed to assert once and then deassert once, by programming the timing registers associated with the output. Each output has 4 8-bit timing registers associated with it. The name of the register matches the name of the output:

*OUTPUTNAME*_TX_HI[7:0]
*OUTPUTNAME*_TX_LO[7:0]
*OUTPUTNAME*_RX_HI[7:0]
*OUTPUTNAME*_RX_LO[7:0]

In the case of grouped outputs, the timing registers associated with the signal controlling the group, control timing for the entire group.

During the WARMUP phase of a TX sequence, as the 8-bit TSM counter increments, once the TSM counter equals or exceeds the programmed value of *OUTPUTNAME*_TX_HI[7:0], but is less than the programmed value of *OUTPUTNAME*_TX_LO[7:0], the corresponding TSM output will transition high. During the WARMUP or WARMDOWN phase of the TX sequence, once the TSM counter equals or exceeds the programmed value of *OUTPUTNAME*_TX_LO[7:0], the TSM output will transition low. If the programmed value of *OUTPUTNAME*_TX_HI[7:0], is greater than the length of the TX sequence (set by register END_OF_TX_WD[7:0]), then that signal will never transition high during the TX sequence. A convenient way of ensuring a signal does not assert during a TX sequence, is to set its *OUTPUTNAME*_TX_HI[7:0]=255.

During the WARMUP phase of a RX sequence, as the 8-bit TSM counter increments, once the TSM counter equals or exceeds the programmed value of *OUTPUTNAME*_RX_HI[7:0], but is less than the programmed value of *OUTPUTNAME*_RX_LO[7:0], the corresponding TSM output will transition high. During the WARMUP or WARMDOWN phase of the RX sequence, once the TSM counter equals or exceeds the programmed value of *OUTPUTNAME*_RX_LO[7:0], the TSM output will transition low. If the programmed value of *OUTPUTNAME*_RX_HI[7:0], is greater than the length of the RX sequence (set by register END_OF_RX_WD[7:0]), then that signal will never transition high during the RX sequence. A convenient way of assuring a signal does not assert during a RX sequence, is to set its *OUTPUTNAME*_RX_HI[7:0]=255.

The following diagram depicts the TSM counter, and the control logic used to assert and deassert the TSM-controlled outputs. For clarity, logic for only 2 of the 56 outputs is shown.

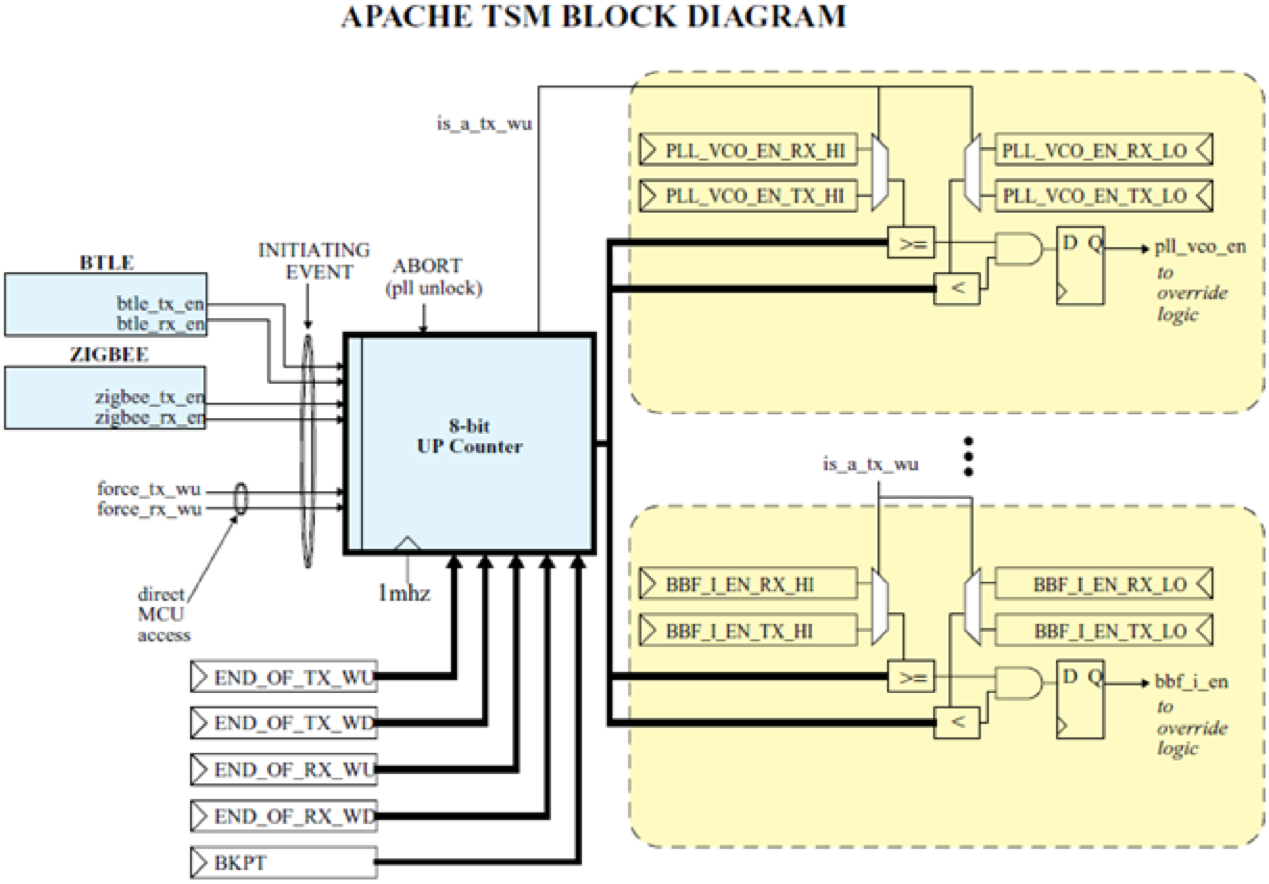


Figure 50-2. TSM Block Diagram

50.4.3 Timing Registers

There are 44 sets of timing registers for the TSM, one set for each TSM-controlled output (or, for the signal controlling the group for the "grouped" outputs). Each register set consists of 4 registers: one to control the assertion time of the output for TX sequences, one to control the assertion time for RX sequence, one to control the deassertion time for TX sequences, and one to control the deassertion time for RX sequences. For TSM-controlled outputs that are grouped, one set of timing registers controls the timing for all members of the group. For each of these 44 sets, the following table lists the register name (TSM_TIMING00 – TSM_TIMING43), and 4 timing register names associated with each.

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING00	PLL_REG_EN_RX_LO[7:0]	PLL_REG_EN_RX_HI[7:0]	PLL_REG_EN_TX_LO[7:0]	PLL_REG_EN_TX_HI[7:0]
TSM_TIMING01	PLL_VCO_REG_EN_RX_LO[7:0]	PLL_VCO_REG_EN_RX_HI[7:0]	PLL_VCO_REG_EN_TX_LO[7:0]	PLL_VCO_REG_EN_TX_HI[7:0]
TSM_TIMING02	QGEN_REG_EN_RX_LO[7:0]	QGEN_REG_EN_RX_HI[7:0]	QGEN_REG_EN_TX_LO[7:0]	QGEN_REG_EN_TX_HI[7:0]
TSM_TIMING03	TCA_TX_REG_EN_RX_LO[7:0]	TCA_TX_REG_EN_RX_HI[7:0]	TCA_TX_REG_EN_TX_LO[7:0]	TCA_TX_REG_EN_TX_HI[7:0]
TSM_TIMING04	ADC_REG_EN_RX_LO[7:0]	ADC_REG_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING05	PLL_REF_CLK_EN_RX_LO[7:0]	PLL_REF_CLK_EN_RX_HI[7:0]	PLL_REF_CLK_EN_TX_LO[7:0]	PLL_REF_CLK_EN_TX_HI[7:0]
TSM_TIMING06	ADC_CLK_EN_RX_LO[7:0]	ADC_CLK_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING07	PLL_VCO_AUTOTUNE_EN_RX_LO[7:0]	PLL_VCO_AUTOTUNE_EN_RX_HI[7:0]	PLL_VCO_AUTOTUNE_EN_TX_LO[7:0]	PLL_VCO_AUTOTUNE_EN_TX_HI[7:0]
TSM_TIMING08	PLL_CYCLE_SLIP_LD_EN_RX_LO[7:0]	PLL_CYCLE_SLIP_LD_EN_RX_HI[7:0]	PLL_CYCLE_SLIP_LD_EN_TX_LO[7:0]	PLL_CYCLE_SLIP_LD_EN_TX_HI[7:0]
TSM_TIMING09	PLL_VCO_EN_RX_LO[7:0]	PLL_VCO_EN_RX_HI[7:0]	PLL_VCO_EN_TX_LO[7:0]	PLL_VCO_EN_TX_HI[7:0]
TSM_TIMING10	PLL_VCO_BUF_RX_EN_RX_LO[7:0]	PLL_VCO_BUF_RX_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING11	n/a	n/a	PLL_VCO_BUF_TX_EN_TX_LO[7:0]	PLL_VCO_BUF_TX_EN_TX_HI[7:0]
TSM_TIMING12	n/a	n/a	PLL_PA_BUF_EN_TX_LO[7:0]	PLL_PA_BUF_EN_TX_HI[7:0]
TSM_TIMING13	PLL_LDV_EN_RX_LO[7:0]	PLL_LDV_EN_RX_HI[7:0]	PLL_LDV_EN_TX_LO[7:0]	PLL_LDV_EN_TX_HI[7:0]
TSM_TIMING14	PLL_RX_LDV_RIPPLE_MUX_EN_RX_LO[7:0]	PLL_RX_LDV_RIPPLE_MUX_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING15	n/a	n/a	PLL_TX_LDV_RIPPLE_MUX_EN_TX_LO[7:0]	PLL_TX_LDV_RIPPLE_MUX_EN_TX_HI[7:0]
TSM_TIMING16	PLL_FILTER_CHARGE_EN_RX_LO[7:0]	PLL_FILTER_CHARGE_EN_RX_HI[7:0]	PLL_FILTER_CHARGE_EN_TX_LO[7:0]	PLL_FILTER_CHARGE_EN_TX_HI[7:0]
TSM_TIMING17	PLL_PHDET_EN_RX_LO[7:0]	PLL_PHDET_EN_RX_HI[7:0]	PLL_PHDET_EN_TX_LO[7:0]	PLL_PHDET_EN_TX_HI[7:0]
TSM_TIMING18	QGEN25_EN_RX_LO[7:0]	QGEN25_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING19	n/a	n/a	TX_EN_TX_LO[7:0]	TX_EN_TX_HI[7:0]
TSM_TIMING20	ADC_EN_RX_LO[7:0]	ADC_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING21	ADC_I_Q_EN_RX_LO[7:0]	ADC_I_Q_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING22	ADC_DAC_EN_RX_LO[7:0]	ADC_DAC_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING23	ADC_RST_EN_RX_LO[7:0]	ADC_RST_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING24	BBF_EN_RX_LO[7:0]	BBF_EN_RX_HI[7:0]	n/a	n/a

Table continues on the next page...

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING25	TCA_EN_RX_LO[7:0]	TCA_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING26	PLL_DIG_EN_RX_LO[7:0]	PLL_DIG_EN_RX_HI[7:0]	PLL_DIG_EN_TX_LO[7:0]	PLL_DIG_EN_TX_HI[7:0]
TSM_TIMING27	n/a	n/a	TX_DIG_EN_TX_LO[7:0]	TX_DIG_EN_TX_HI[7:0]
TSM_TIMING28	RX_DIG_EN_RX_LO[7:0]	RX_DIG_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING29	RX_INIT_RX_LO[7:0]	RX_INIT_RX_HI[7:0]	n/a	n/a
TSM_TIMING30	SIGMA_DELTA_EN_RX_LO[7:0]	SIGMA_DELTA_EN_RX_HI[7:0]	SIGMA_DELTA_EN_TX_LO[7:0]	SIGMA_DELTA_EN_TX_HI[7:0]
TSM_TIMING31	ZBDEM_RX_EN_RX_LO[7:0]	ZBDEM_RX_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING32	DCOC_EN_RX_LO[7:0]	DCOC_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING33	DCOC_INIT_RX_LO[7:0]	DCOC_INIT_RX_HI[7:0]	n/a	n/a
TSM_TIMING34	FREQ_TARG_LD_EN_RX_LO[7:0]	FREQ_TARG_LD_EN_RX_HI[7:0]	FREQ_TARG_LD_EN_TX_LO[7:0]	FREQ_TARG_LD_EN_TX_HI[7:0]
TSM_TIMING35	SAR_ADC_TRIG_EN_RX_LO[7:0]	SAR_ADC_TRIG_EN_RX_HI[7:0]	SAR_ADC_TRIG_EN_TX_LO[7:0]	SAR_ADC_TRIG_EN_TX_HI[7:0]
TSM_TIMING36	TSM_SPARE0_EN_RX_LO[7:0]	TSM_SPARE0_EN_RX_HI[7:0]	TSM_SPARE0_EN_TX_LO[7:0]	TSM_SPARE0_EN_TX_HI[7:0]
TSM_TIMING37	TSM_SPARE1_EN_RX_LO[7:0]	TSM_SPARE1_EN_RX_HI[7:0]	TSM_SPARE1_EN_TX_LO[7:0]	TSM_SPARE1_EN_TX_HI[7:0]
TSM_TIMING38	TSM_SPARE2_EN_RX_LO[7:0]	TSM_SPARE2_EN_RX_HI[7:0]	TSM_SPARE2_EN_TX_LO[7:0]	TSM_SPARE2_EN_TX_HI[7:0]
TSM_TIMING39	TSM_SPARE3_EN_RX_LO[7:0]	TSM_SPARE3_EN_RX_HI[7:0]	TSM_SPARE3_EN_TX_LO[7:0]	TSM_SPARE3_EN_TX_HI[7:0]
TSM_TIMING40	GPIO0_TRIG_EN_RX_LO[7:0]	GPIO0_TRIG_EN_RX_HI[7:0]	GPIO0_TRIG_EN_TX_LO[7:0]	GPIO0_TRIG_EN_TX_HI[7:0]
TSM_TIMING41	GPIO1_TRIG_EN_RX_LO[7:0]	GPIO1_TRIG_EN_RX_HI[7:0]	GPIO1_TRIG_EN_TX_LO[7:0]	GPIO1_TRIG_EN_TX_HI[7:0]
TSM_TIMING42	GPIO2_TRIG_EN_RX_LO[7:0]	GPIO2_TRIG_EN_RX_HI[7:0]	GPIO2_TRIG_EN_TX_LO[7:0]	GPIO2_TRIG_EN_TX_HI[7:0]
TSM_TIMING43	GPIO3_TRIG_EN_RX_LO[7:0]	GPIO3_TRIG_EN_RX_HI[7:0]	GPIO3_TRIG_EN_TX_LO[7:0]	GPIO3_TRIG_EN_TX_HI[7:0]

Some of the TSM outputs have relevance for TX sequences only. For those, timing registers are only provided for TX assertion and deassertion times. For such outputs, the RX timing registers are shown as “n/a” in the table above. The RX timings for such outputs are essentially hardwired to 255, meaning there will be no signal assertion during RX sequences.

Some of the TSM outputs have relevance for RX sequences only. For those, timing registers are only provided for RX assertion and deassertion times. For such outputs, the TX timing registers are shown as “n/a” in the table above. The TX timings for such outputs are essentially hardwired to 255, meaning there will be no signal assertion during TX sequences.

50.4.4 Sequence Initiation

The SoC will include one or more protocol engines. Any of the protocol engines, can launch a TSM sequence. Each protocol engine will have an initiating signal for a TX sequence, and another for an RX sequence. In lieu of a protocol engine, the SoC host can launch a TSM sequence directly. The following table lists the initiating sources supported by the TSM, and the name of the sequence launching signal (TSM input):

Initiating Source	Initiating Signal Names
BTLE	btle_tx_en, btle_rx_en
Zigbee	zigbee_tx_en, zigbee_rx_en
MCU / Host	force_tx_en, force_rx_en

For direct software control, the MCU/Host can initiate a TX sequence warmup directly, by setting the FORCE_TX_EN bit, and later initiate a TX sequence warmdown by clearing the bit. The MCU/Host can initiate a RX sequence warmup directly, by setting the FORCE_RX_EN bit, and later initiate a RX sequence warmdown by clearing the bit. These bits reside in the TSM_CTRL register.

From idle state, the asserting of any initiating event (*source*_tx_en or *source*_rx_en), will begin the TSM WARMUP phase and begin incrementing the TSM counter from 0. The initiating source needs to hold the initiating signal asserted (high) through the course of the WARMUP, and then for the duration of the desired ON phase. Deasserting the initiating signal transitions the TSM to the WARMDOWN phase (see Section “Sequence Termination”).

Needless to say, from idle state, a TX-initiating event (assertion of any TSM *source*_tx_en input) will launch a TX sequence. The TSM output **tx_mode** will go high to indicate a TX sequence is underway. Also, from idle state, an RX-initiating event (assertion of any TSM *source*_rx_en input) will launch an RX sequence. The TSM output **rx_mode** will go high to indicate a RX sequence is underway.

Only 1 initiating source can be allowed to assert at any given time. Under no circumstances should a TX and RX initiating source be asserted simultaneously.

Once a TSM sequence has been launched, the WARMUP phase is entered and TSM counter incrementing will commence. Up-counting will continue until one of the following conditions is met:

1. Deassertion of the initiating event. TSM will transition to WARMDOWN phase
2. Abort. TSM will transition to WARMDOWN phase
3. tsm_count==END_OF_*SEQ* _WU, where *SEQ* = TX or RX. TSM will transition to ON phase
4. Breakpoint. TSM stay in its current phase and hold its count.

A transition from WARMUP to ON phase means TSM will hold its count at END_OF_*SEQ*_WU, where *SEQ* = TX or RX.

A transition from ON to WARMDOWN phase means TSM will cease its hold, and resume counting at END_OF_*SEQ*_WU+1, where *SEQ* = TX or RX. (see Section “Sequence Termination”).

A transition from WARMUP to WARMDOWN phase means the tsm_count will jump to END_OF_*SEQ*_WU+1 (where *SEQ* = TX or RX), and resume counting from there. (see Section “Sequence Termination”).

The 4 8-bit registers which control the duration of the WARMUP phase (END_OF_TX_WU[7:0] and END_OF_RX_WU[7:0]), and the duration of the WARMDOWN phase (END_OF_TX_WD[7:0] and END_OF_RX_WD[7:0]), reside in the END_OF_SEQ register.

50.4.5 Sequence Termination

A sequence may be terminated during the WARMUP on ON phase. A sequence will be terminated when one of the following conditions is met:

1. Deassertion of the initiating event. TSM will transition to WARMDOWN phase
2. Abort. TSM will transition to WARMDOWN phase

Deassertion of the initiating event (*source*_tx_en or *source*_rx_en), will cause a transition to WARMDOWN phase, at which point the TSM counter will resume counting from END_OF_*SEQ*_WU+1 (where *SEQ* = TX or RX).

An abort is caused by a PLL unlock event. An unlock abort will cause a transition to WARMDOWN phase, at which point the TSM counter will resume counting from END_OF_*SEQ*_WU+1 (where *SEQ* = TX or RX). See Section TSM Aborting of this Block Guide.

During the WARMDOWN phase, whether caused by a deassertion of the initiating event, or a PLL unlock, subsequent aborts (PLL unlock events) will not be recognized by the TSM.

50.4.6 Overrides

Each of the 56 TSM-controlled outputs has independent override control, except for the 4 GPIO triggers. In addition, two special TSM outputs, tx_mode and rx_mode, also have their own override controls, for a total of 56 – 4 + 2 = 54 overrides.

Two register bits are assigned to each output, to implement the override function. The bits are named:

*OUTPUTNAME*_OVRD_EN
*OUTPUTNAME*_OVRD

Setting the *OUTPUTNAME*_OVRD_EN=1, allows the *OUTPUTNAME*_OVRD bit to directly control the state of the output. Clearing *OUTPUTNAME*_OVRD_EN=0 returns control of the output to the TSM.

The 54 override bit-pairs, reside in the TSM_OVRD0, TSM_OVRD1, TSM_OVRD2, and TSM_OVRD3 registers of XCVR address space.

Overrides have no effect on TSM operation, TSM counting, sequence initiation, or sequence termination. Overrides may be engaged at any time, during any TSM phase, including during idle state.

50.4.7 Breakpoint

Breakpoint can be used to temporarily suspend a TSM sequence. A breakpoint can be set during the WARMUP or WARMDOWN phase of any TSM sequence. An 8-bit BKPT[7:0] register field resides in the TSM_CTRL register. During a TSM sequence, when the TSM counter matches the BKPT register value, counting stops and the counter holds in its current state. Aborts (PLL unlocks) will be ignored while holding at a breakpoint. Deassertion of the initiating event will also be ignored during the breakpoint. The breakpoint can be lifted by modifying the BKPT[7:0] register. Once the breakpoint is lifted, the TSM proceeds in the phase it was in prior to the breakpoint match, and the TSM counter begins incrementing again from the point at which the breakpoint occurred. Aborts (PLL unlocks) and deassertion of the initiaing event, will then be recognized and handled as per the description in the “Sequence Termination” section. The default value for BKPT[7:0] is 255, which is greater than the length of any allowed sequence, so a breakpoint will not trigger during a sequence, unless a lower value is programmed.

50.4.8 TSM Aborting

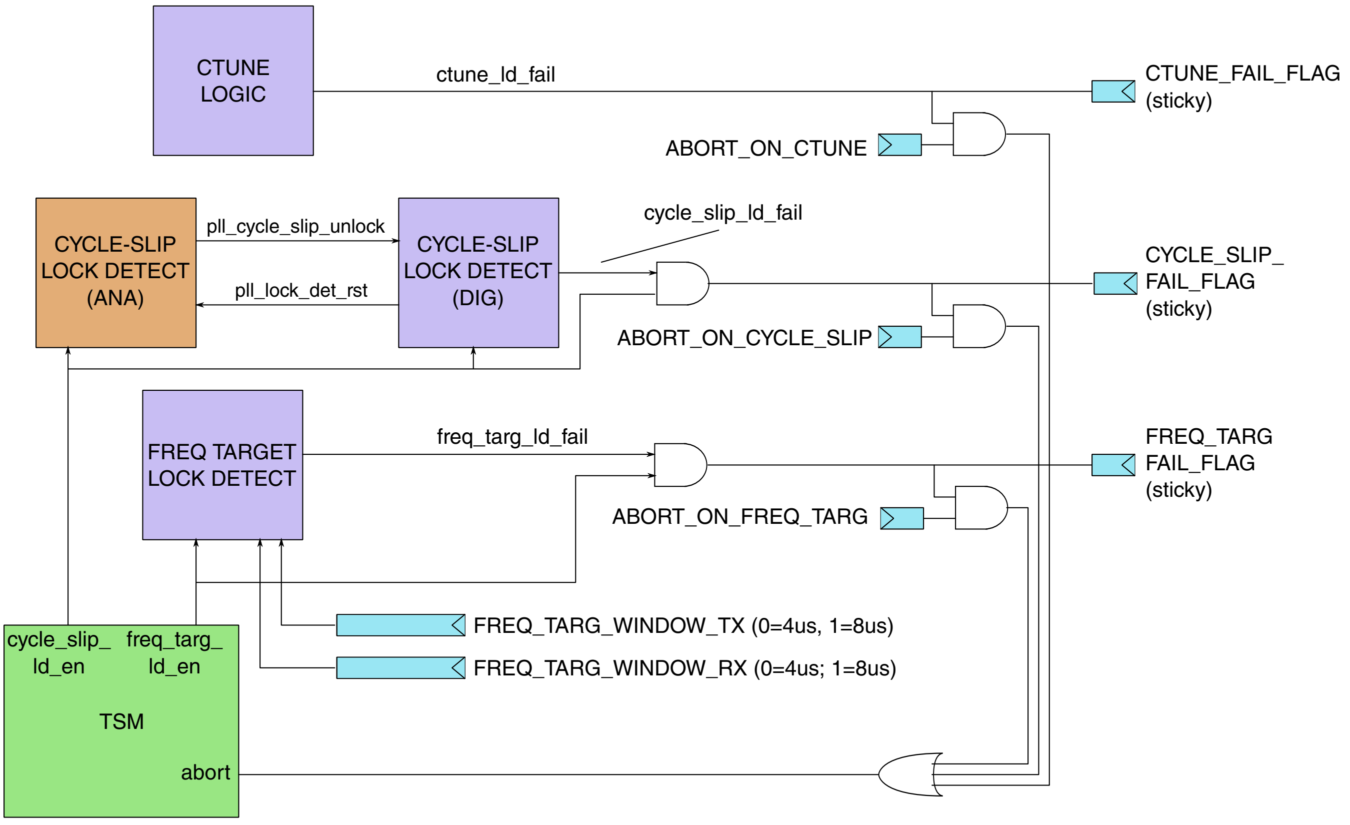
The TSM can be enabled to abort any sequence on a PLL unlock condition. There are 3 mechanisms available for detecting PLL unlock. These are:

1. Coarse Tune Unlock
2. Frequency Target Unlock
3. Cycle Slip Unlock.

Any of these mechanisms, or any combination of them, can be enabled to cause a TSM abort. Each of these mechanisms has a “_FAIL_FLAG” flag in the PLL_LOCK_DETECT register. On an unlock detection by any of these mechanisms, the corresponding “_FAIL_FLAG” flag (sticky bit) will become set, regardless of whether the mechanism is enabled to abort the TSM sequence. The “_FAIL_FLAG” flags are type “write-1-to-clear”.

An overview of the 3 PLL unlock mechanism is shown below.

PLL UNLOCK MONITORING AND ABORT LOGIC



Each of the 3 PLL unlock mechanisms has an “ABORT_ON_” bit, to enable or disable that mechanism from aborting any TSM sequence. Any combination of “ABORT_ON_” bits can be configured.

In addition, the TSM directly controls the timing for 2 of the 3 unlock detect mechanisms: the Cycle Slip Lock Detect and the Frequency Target Lock Detect. (The timing for Coarse Tune Lock Detect mechanism is indirectly controlled by TSM via the pll_dig_en output; the PLL digital state machine is triggered by this signal and directly controls Coarse Tune). Enabling a lock detection mechanism via TSM programming allows an unlock condition to set the respective “_FAIL_FLAG” flag bit; To allow an unlock condition to cause an abort requires the respective “ABORT_ON_” bit to be set.

For each PLL unlock detect mechanism, the table below summarizes the TSM controlling signal for the mechanism, the timing register associated with that signal, and the default timing for the assertion of the controlling signal for both TX and RX sequences.

Lock Detect Mechanism	TSM Controlling Signal	Associated TSM Timing Register	Controlling Signal Assertion Time (TX)	Controlling Signal Assertion Time (RX)
Coarse Tune	pll_dig_en (indirect)	TSM_TIMING26	9us (default)	9us (default)
Freq. Target	freq_targ_ld_en	TSM_TIMING34	101us (default)	51us (default)
Cycle Slip	pll_cycle_slip_ld_en	TSM_TIMING08	103us (default)	51us (default)

For each PLL unlock detect mechanism, the table below indicates the TSM controlling signal, the name of the “_FAIL_FLAG” flag bit, and the name of the “ABORT_ON_” enable bit to allow TSM aborting:

Lock Detect Mechanism	TSM Controlling Signal	Lock Detect Fail Flag (PLL_LOCK_DETECT register)	Abort Enable Bit (TSM_CTRL register)
Coarse Tune	pll_dig_en (indirect)	CTUNE_FAIL_FLAG	ABORT_ON_CTUNE
Freq. Target	freq_targ_ld_en	FREQ_TARG_FAIL_FLAG	ABORT_ON_FREQ_TARG
Cycle Slip	pll_cycle_slip_ld_en	CYCLE_SLIP_FAIL_FLAG	ABORT_ON_CYCLE_SLIP

All “_FAIL_FLAG” flag bits all reside in the PLL_LOCK_DETECT register; all “ABORT_ON_” bits reside in the TSM_CTRL register.

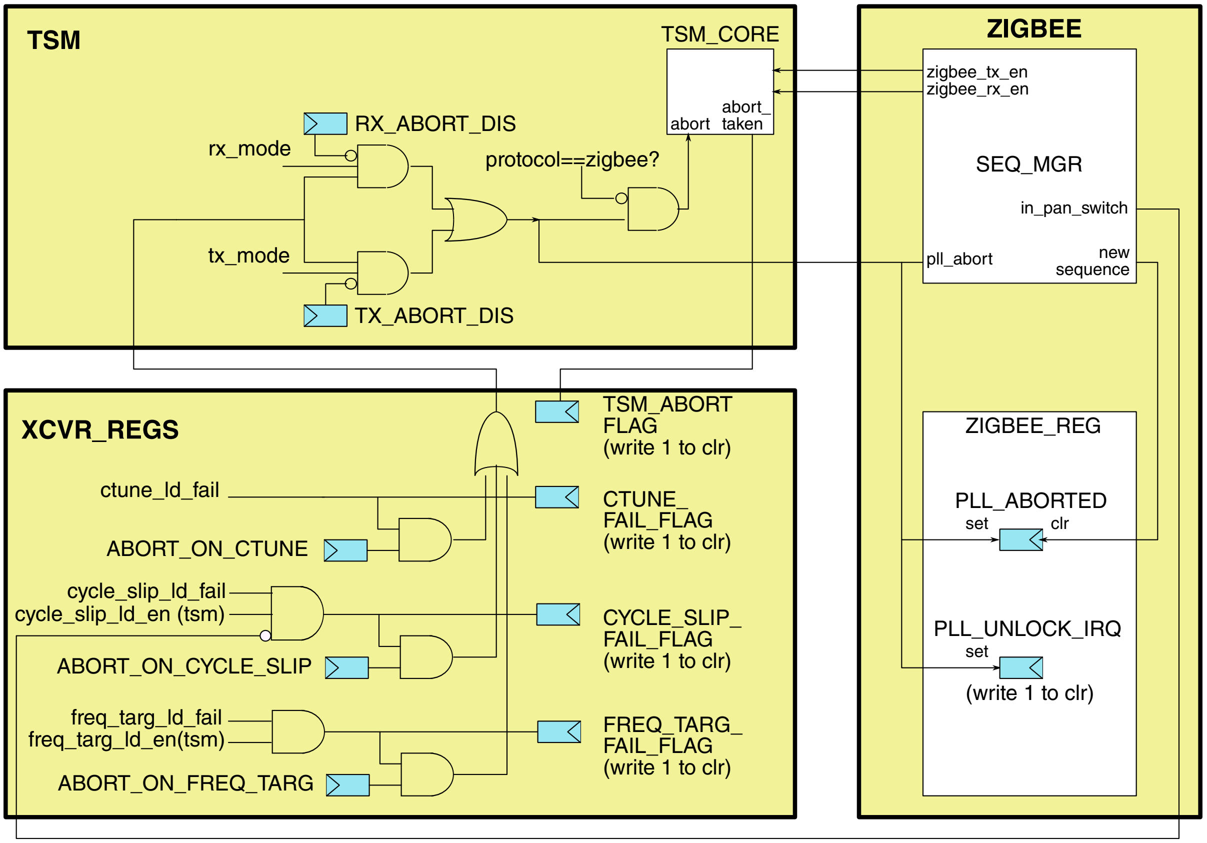
In addition to the controls listed above associated with each unlock source, there are 2 controls to allow TSM aborting to be disabled based on sequence type (TX or RX), instead of unlock source. When TX_ABORT_DIS is set to 1, TX sequences cannot be aborted, regardless of the state of the “ABORT_ON_” bits or any unlock condition. Likewise, when RX_ABORT_DIS is set to 1, RX sequences cannot be aborted, regardless of the state of the “ABORT_ON_” bits or any unlock condition. The TX_ABORT_DIS and RX_ABORT_DIS bits reside in the TSM_CTRL register. The

TX_ABORT_DIS and RX_ABORT_DIS bits have no effect on the setting of the "_FAIL_FLAG" bits on any unlock condition; the TX_ABORT_DIS and RX_ABORT_DIS only affect aborting.

50.4.9 Special Handling for Zigbee Autoseqences

Any of the 3 unlock detect mechanisms can be used to abort a Zigbee autosequence. However the method for aborting a Zigbee autosequence is different than a non-Zigbee abort. For Zigbee, the unlock condition does not abort the sequence directly. There is a higher-level Zigbee Sequence Manager, which handles unlock aborts. For Zigbee, the unlock condition is blocked from aborting the TSM as shown in the following diagram:

ZIGBEE PLL UNLOCK HANDLING



The combined-abort signal is blocked from reaching the TSM, and instead is routed to the Zigbee Sequence Manager (ZSM). This is due to the fact that the ZSM already includes abort-handling logic, which is being re-used from previous Freescale Zigbee products. Keeping the unlock handling intact, facilitates the porting of software from one Freescale Zigbee product to the next.

For Zigbee, the logic which determines which ZSM states allow aborting, is designed into the ZSM state machine. Not all states require an abort-on-unlock. When an unlock event occurs during a ZSM state that requires an abort, the ZSM will deassert its initiating signal to TSM (zigbee_tx_en or zigbee_rx_en), which results in a de facto abort, similar to a TSM-generated abort. For a ZSM abort, the PLL_UNLOCK_IRQ bit will become set, as will the PLL_ABORTED bit. The PLL_UNLOCK_IRQ bit is of type write-1-to-clear, and resides in the IRQSTS1 register of Zigbee space. The PLL_ABORTED bit is a read-only bit, and resides in the ABORT_STS register of Zigbee space; it will self-clear at the start of the next autosequence.

For Zigbee, the TX_ABORT_DIS and RX_ABORT_DIS control bits are still available, and allow/disallow aborting during the TX portions of any autosequence (TX_ABORT_DIS), or the RX portions of any autosequence (RX_ABORT_DIS). Zigbee autosequences are concatenations of TX and RX operations, under the control of the Zigbee Sequence Manager. The ZSM engages the TSM when needed, to perform a TX or RX warmup or warmdown, by asserting or deasserting zigbee_tx_en or zigbee_rx_en.

See the ZSM Sequence Manager Block Guide, Section ZSM/TSM Interaction.

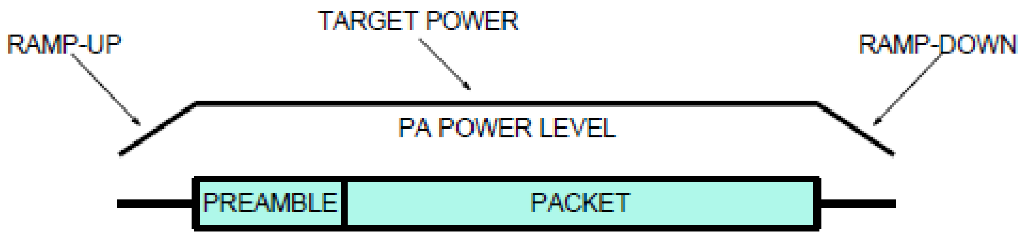
As mentioned previously, for Zigbee the PLL_UNLOCK_IRQ will become set to indicate a PLL unlock event has caused a Zigbee autosequence to be aborted. When Zigbee is not in use, another bit is provided to indicate a non-Zigbee unlock abort (a TSM abort) has occurred. This bit is TSM_ABORT_FLAG. This bit is of type write-1-to-clear, and resides in the PLL_LOCK_DETECT register of transceiver space. This bit will not be set for a Zigbee abort (use PLL_UNLOCK_FLAG or PLL_ABORTED instead).

50.4.10 PA Target Power

The TSM controls the power level that is sent to the PA. Target power level is the power level that is used during packet transmission. Nominally, power level is constant over the course of the packet.

During a packet transmission sequence, prior to the start of preamble, while the PA is being turned on, power level must be ramped up gradually, to prevent unwanted spectral effects. After the packet is complete, and the PA is being turned off, power level must be ramped down slowly for the same reason.

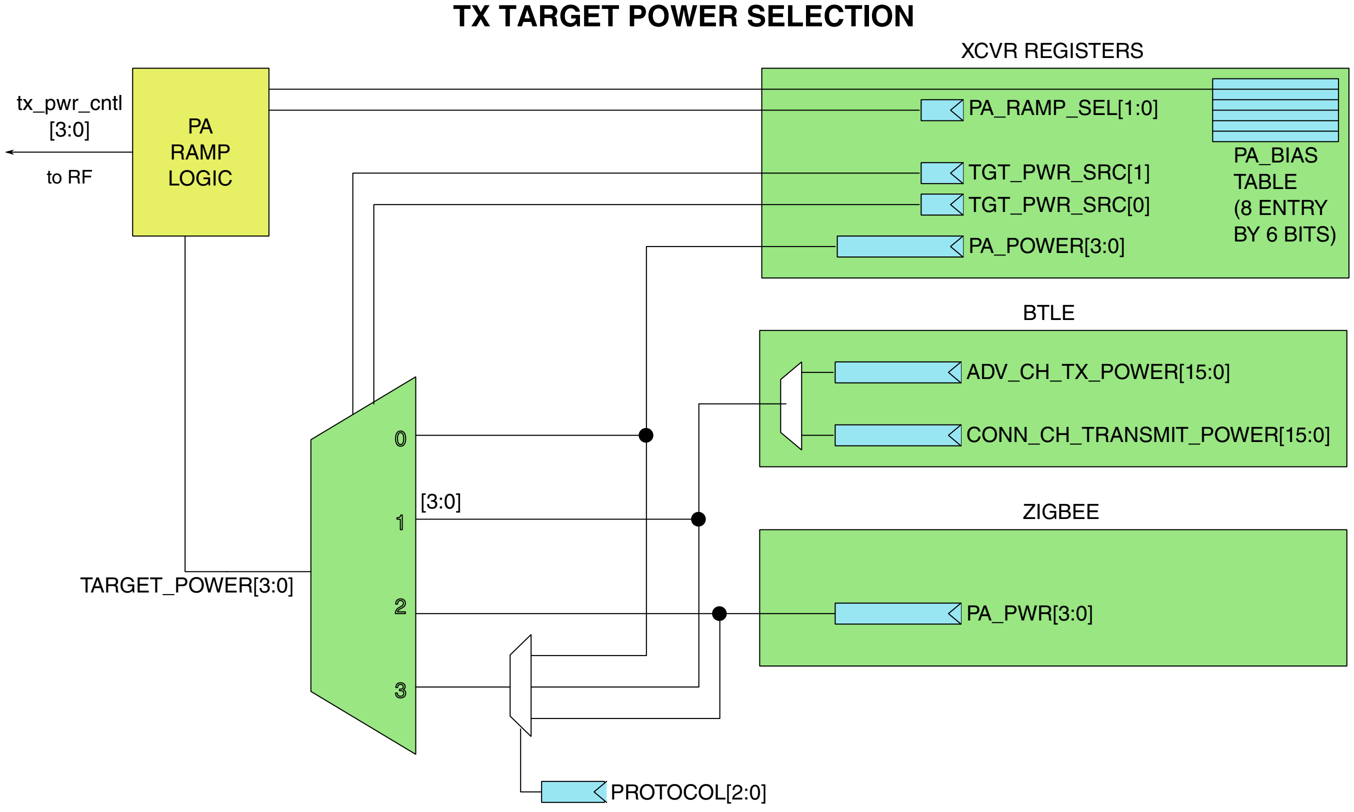
The TSM controls power level to the PA during these 3 phases, as shown below.



Target power, the steady-state power level that is used to transmit the packet, can be selected from one of 3 sources:

1. PA_POWER[3:0] register in XCVR address space
2. BTLE Link Layer (2 different power levels possible, for Advertising and Connection Packets)
3. Zigbee Link Layer (PA_PWR[3:0] register in ZIGBEE address space)The following diagram depicts target power selection.

The following diagram depicts target power selection.



Two register control bits, TGT_PWR_SRC[1:0], control target power selection, according to the following table.

TGT_PWR_SRC[1:0]	TARGET POWER SOURCE
00	PA_POWER[3:0] register (XCVR space)
01	BTLE Link Layer
10	Zigbee Link Layer (PA_PWR[3:0] register in ZIGBEE space)
11	PROTOCOL[2:0] bits select target power source

These bits reside in the XCVR_CTRL register in XCVR address space.

The TSM provides power control information to the PA as a 4-bit output: tx_pwr_cntl[3:0].

50.4.11 PA Ramping

PA ramping is included in the TSM, to prevent abrupt transitions in PA (power amplifier) power, that could cause unwanted spectral transients. During the WARMUP phase of a TX sequence, PA ramping gradually ramps up PA power, between a programmable minimum, and the TX target power. The trajectory of the ramp is programmable, and ramping takes place over the course of 2, 4, or 8us (programmable).

During a TX sequence, PA ramping, if enabled, is triggered by a low-to-high transition on the TSM output tx_en, which is also the enable for the PA (tx buffer). At assertion of tx_en during the TX sequence with ramping enabled, the TSM will enable the PA, with a minimum power level determined by the register PA_BIAS0[3:0]. After that, power level will increment according to a 8-deep pre-programmed register table (PA_BIAS Table). Incrementing will occur at a programmable rate. At each ramp step, PA power level will be taken from the next entry in the PA_BIAS Table, from 0-7. After the eighth and final ramp step, PA power level will be the selected target power. The target power will remain in place for the duration of the packet transmission, i.e., the “ON” phase of the TSM sequence.

The ramp trajectory is determined by the contents of the PA_BIAS Table. The PA_BIAS table contains register fields PA_BIAS0[3:0] … PA_BIAS7[3:0]. These register fields can be found in the PA_BIAS_TBL0 and PA_BIAS_TBL1 registers in XCVR address space. The contents of the PA_BIAS table are as shown:

PA_BIAS Table
PA_BIAS0[3:0]
PA_BIAS1[3:0]

Table continues on the next page...

PA_BIAS Table
PA_BIAS2[3:0]
PA_BIAS3[3:0]
PA_BIAS4[3:0]
PA_BIAS5[3:0]
PA_BIAS6[3:0]
PA_BIAS7[3:0]

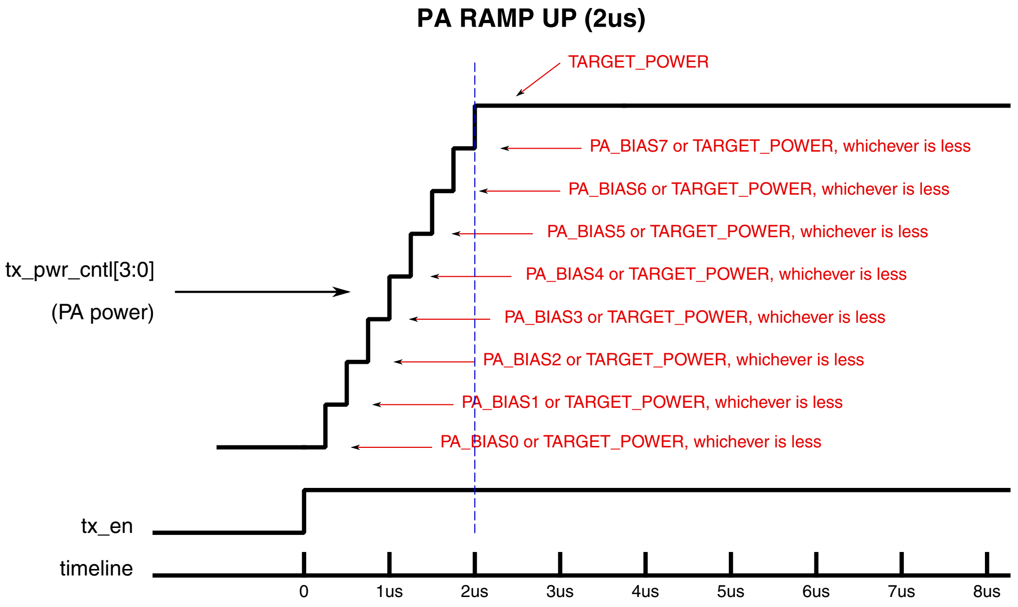
At each step of the ramp-up, the selected entry of the PA_BIAS Table is compared against target_power. The lesser of the two becomes the output power for that ramp step. This allows the same PA_BIAS table to be used for various target_power levels, without re-programming the table.

The ramp rate is controlled by the register bit PA_RAMP_SEL[1:0] bits of the TSM_CTRL register in XCVR address space. For each setting, the table below describes the total ramp duration, and the duration of each ramp step.

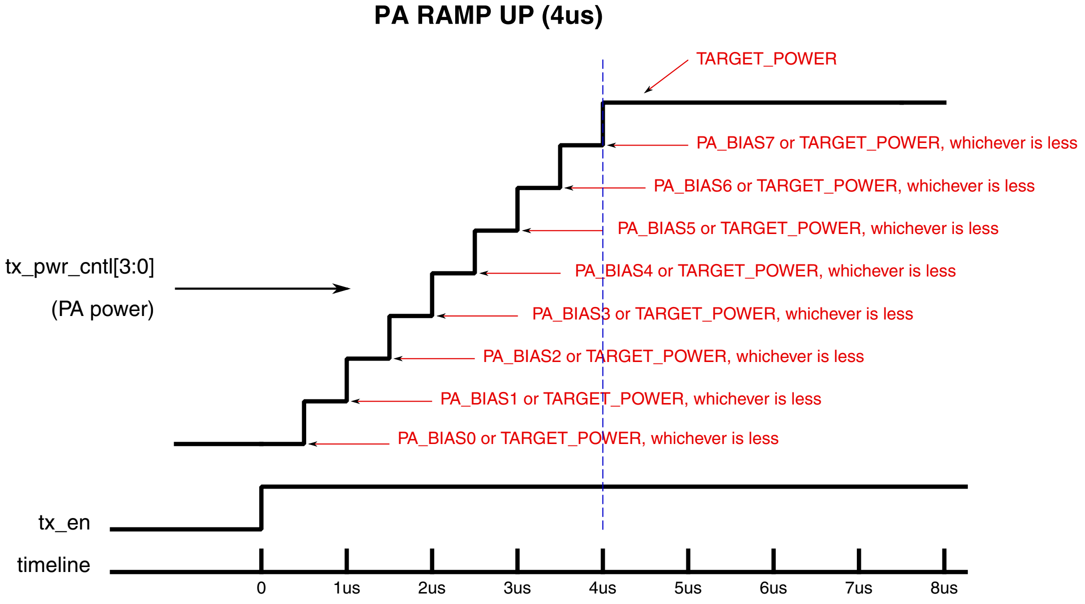
PA_RAMP_SEL[1:0]	TOTAL RAMP DURATION	DURATION OF EACH RAMP STEP
00	No ramp	No ramp
01	2us	0.25us
10	4us	0.5us
11	8us	1us

When PA_RAMP_SEL[1:0] is set to 00, there is no ramping, and tx_pwr_cntl[3:0] tracks the selected Target Power at all time (see Section PA Target Power of this Block Guide).

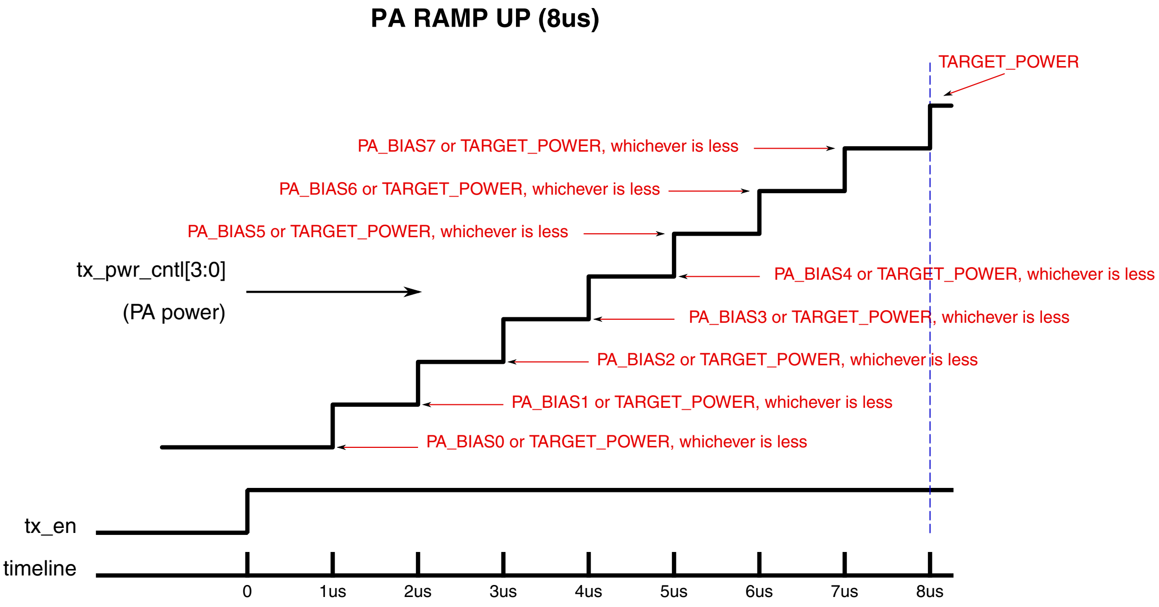
When PA_RAMP_SEL[1:0] is set to 01, a 2us ramp is enabled, as shown:


PA RAMP UP (2us)
TARGET_POWER
PA_BIAS7 or TARGET_POWER, whichever is less
PA_BIAS6 or TARGET_POWER, whichever is less
PA_BIAS5 or TARGET_POWER, whichever is less
PA_BIAS4 or TARGET_POWER, whichever is less
PA_BIAS3 or TARGET_POWER, whichever is less
PA_BIAS2 or TARGET_POWER, whichever is less
PA_BIAS1 or TARGET_POWER, whichever is less
PA_BIAS0 or TARGET_POWER, whichever is less
tx_pwr_cntl[3:0]
(PA power)
tx_en
timeline
0
1us
2us
3us
4us
5us
6us
7us
8us

When PA_RAMP_SEL[1:0] is set to 10, a 4us ramp is enabled, as shown:


PA RAMP UP (4us)
TARGET_POWER
PA_BIAS7 or TARGET_POWER, whichever is less
PA_BIAS6 or TARGET_POWER, whichever is less
PA_BIAS5 or TARGET_POWER, whichever is less
PA_BIAS4 or TARGET_POWER, whichever is less
PA_BIAS3 or TARGET_POWER, whichever is less
PA_BIAS2 or TARGET_POWER, whichever is less
PA_BIAS1 or TARGET_POWER, whichever is less
PA_BIAS0 or TARGET_POWER, whichever is less
tx_pwr_cntl[3:0]
(PA power)
tx_en
timeline
0
1us
2us
3us
4us
5us
6us
7us
8us

When PA_RAMP_SEL[1:0] is set to 11, an 8us ramp is enabled, as shown:


PA RAMP UP (8us)
TARGET_POWER
PA_BIAS7 or TARGET_POWER, whichever is less
PA_BIAS6 or TARGET_POWER, whichever is less
PA_BIAS5 or TARGET_POWER, whichever is less
tx_pwr_cntl[3:0]
(PA power)
PA_BIAS4 or TARGET_POWER, whichever is less
PA_BIAS3 or TARGET_POWER, whichever is less
PA_BIAS2 or TARGET_POWER, whichever is less
PA_BIAS1 or TARGET_POWER, whichever is less
PA_BIAS0 or TARGET_POWER, whichever is less
tx_en
timeline
0
1us
2us
3us
4us
5us
6us
7us
8us

PA ramp-down is the mirror image of ramp-up. Ramp-down is triggered by the deassertion of the sequence-initiating event (i.e.., btle_tx_en, zigbee_tx_en, or force_tx_en). At deassertion of the initiating event, output power will hold at target_power for 1us. Then, output power will step through the PA_BIAS Table in reverse order, starting at PA_BIAS7, and ending at PA_BIAS0. At each step of the ramp-down, the selected entry of the PA_BIAS Table is compared against target_power. The lesser of the two becomes the output power. The PA_RAMP_SEL[1:0] register controls the ramp-down rate in the same manner that it controls the ramp-up rate. After the last step of the ramp-down (PA_BIAS0 level), tx_en deasserts to disable the PA.

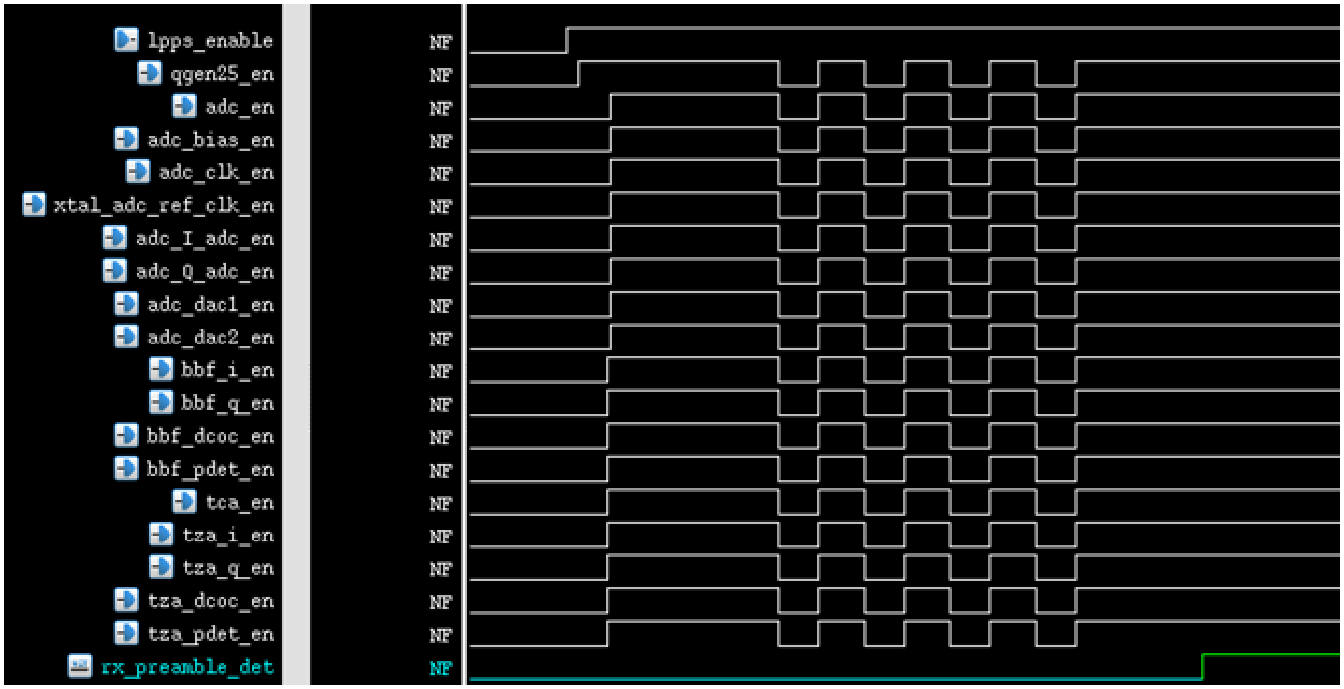
50.4.12 LPPS

To reduce power consumption during Zigbee preamble-search receive state, a Low Power Preamble Search (LPPS) mode can be optionally enabled. When engaged, the correlators are disabled approximately 50% of the time during preamble search. The same signal used to dynamically enable/disable the correlators, can be used to dynamically enable/ disable selected analog and RF blocks in unison, to further reduce power consumption. The LPPS_CTRL register in XCVR space, has one bit to master-enable LPPS mode (LPPS_ENABLE), and 7 other bits to allow selected analog and RF blocks to be “duty-cycled” in sync with the correlators. The bits of LPPS_CTRL, and their functions, are described in the following table.

THIS LPPS_CTRL REGISTER BIT...	... ALLOWS THESE TSM OUTPUTS TO BE DUTY-CYCLED DURING PREAMBLE SEARCH
LPPS_ENABLE	MASTER ENABLE FOR LPPS MODE. ALLOWS CORRELATORS TO BE DUTY-CYCLED
LPPS_QGEN25_ALLOW	qgen25_en
LPPS_ADC_ALLOW	adc_en, adc_bias_en
LPPS_ADC_CLK_ALLOW	xtal_adc_ref_clk_en, adc_clk_en
LPPS_ADC_I_Q_ALLOW	adc_I_adc_en, adc_Q_adc_en
LPPS_ADC_DAC_ALLOW	adc_dac1_en, adc_dac2_en
LPPS_BBF_ALLOW	bbf_i_en, bbf_q_en, bbf_pdet_en, bbf_dcoc_en
LPPS_TCA_ALLOW	tca_en, tza_i_en, tza_q_en, tza_pdet_en, tza_dcoc_en

Note that the penalty for the reduced power consumption in LPPS mode during preamble search, is a slight degradation in sensitivity. Note also that the TSM override bits, also override LPPS functionality. The override bits provide ultimate control over all TSM outputs.

An example of LPPS duty-cycling of all the eligible analog and RF outputs, during preamble detection, is shown in the following diagram. After preamble is detected, LPPS mode is automatically disabled and the correlators and all RF and analog functions are operating full time during packet reception, as shown.



By default, LPPS mode is not enabled.

50.4.13 TSM Programming Restrictions

The following restrictions are imposed on TSM programming. Violating the restrictions may result in unpredictable behavior.

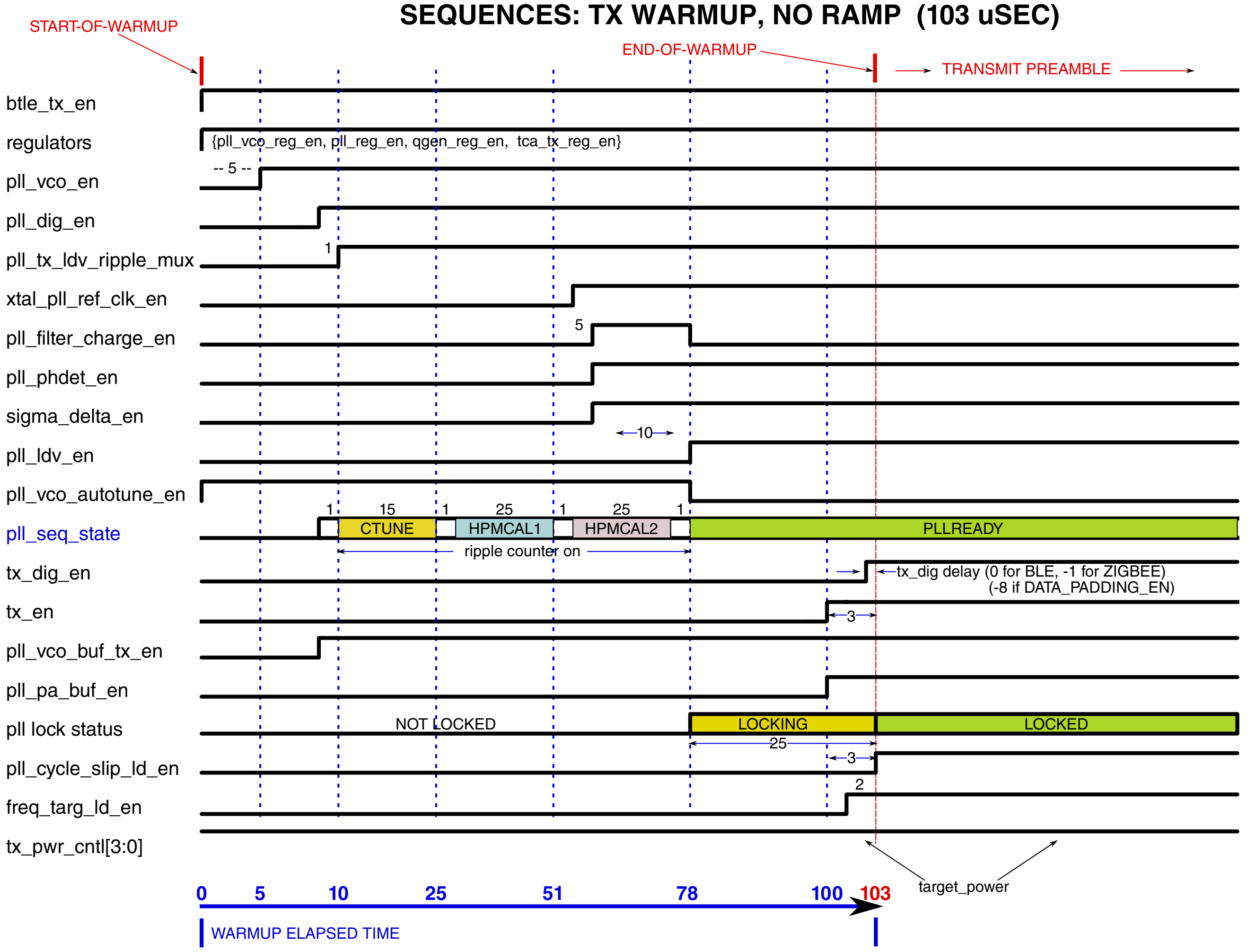
1. The maximum value for END_OF_TX_WU[7:0] is 253
2. The maximum value for END_OF_RX_WU[7:0] is 253
3. The maximum value for END_OF_TX_WD[7:0] is 254
4. The maximum value for END_OF_RX_WD[7:0] is 254
5. END_OF_TX_WD[7:0] – END_OF_TX_WU[7:0] >= 1
6. END_OF_RX_WD[7:0] – END_OF_RX_WU[7:0] >= 1
7. BKPT[7:0] != END_OF_TX_WD[7:0]
8. BKPT[7:0] != END_OF_RX_WD[7:0]
9. For any TSM output: *OUTPUTNAME* _TX_HI[7:0] <= *OUTPUTNAME* _TX_LO[7:0]
10. For any TSM output: *OUTPUTNAME* _RX_HI[7:0] <= *OUTPUTNAME* _RX_LO[7:0]

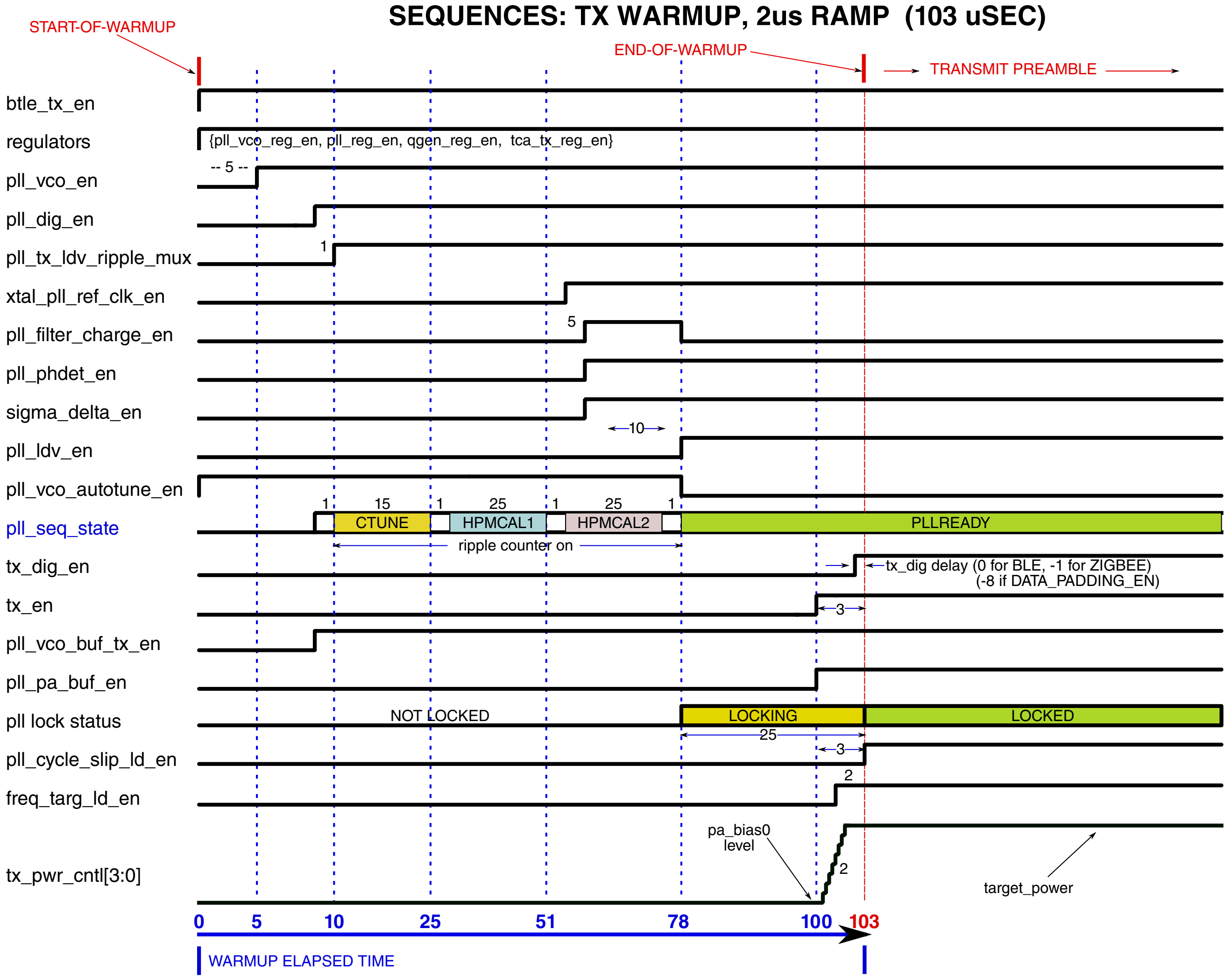
Restrictions 9 and 10 above, if violated, will result in the TSM-controlled output held in a deasserted state throughout the sequence.

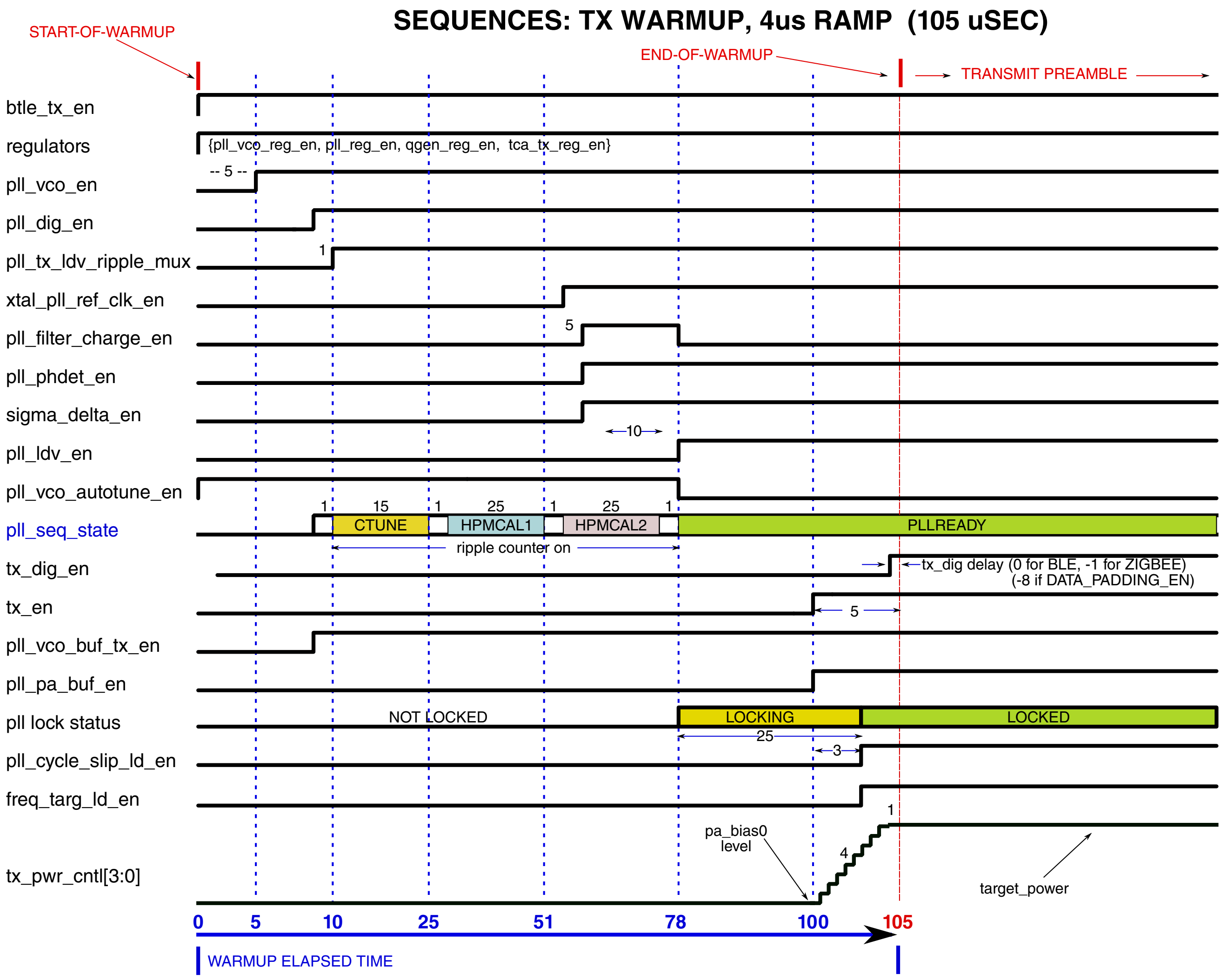
To guarantee non-assertion of a particular TSM-controlled output throughout the course of a sequence, simply program its *OUTPUTNAME*_*SEQ*_HI = *OUTPUTNAME*_*SEQ*_LO = 255 (where *SEQ* = TX or RX).

50.4.14 Default Sequences

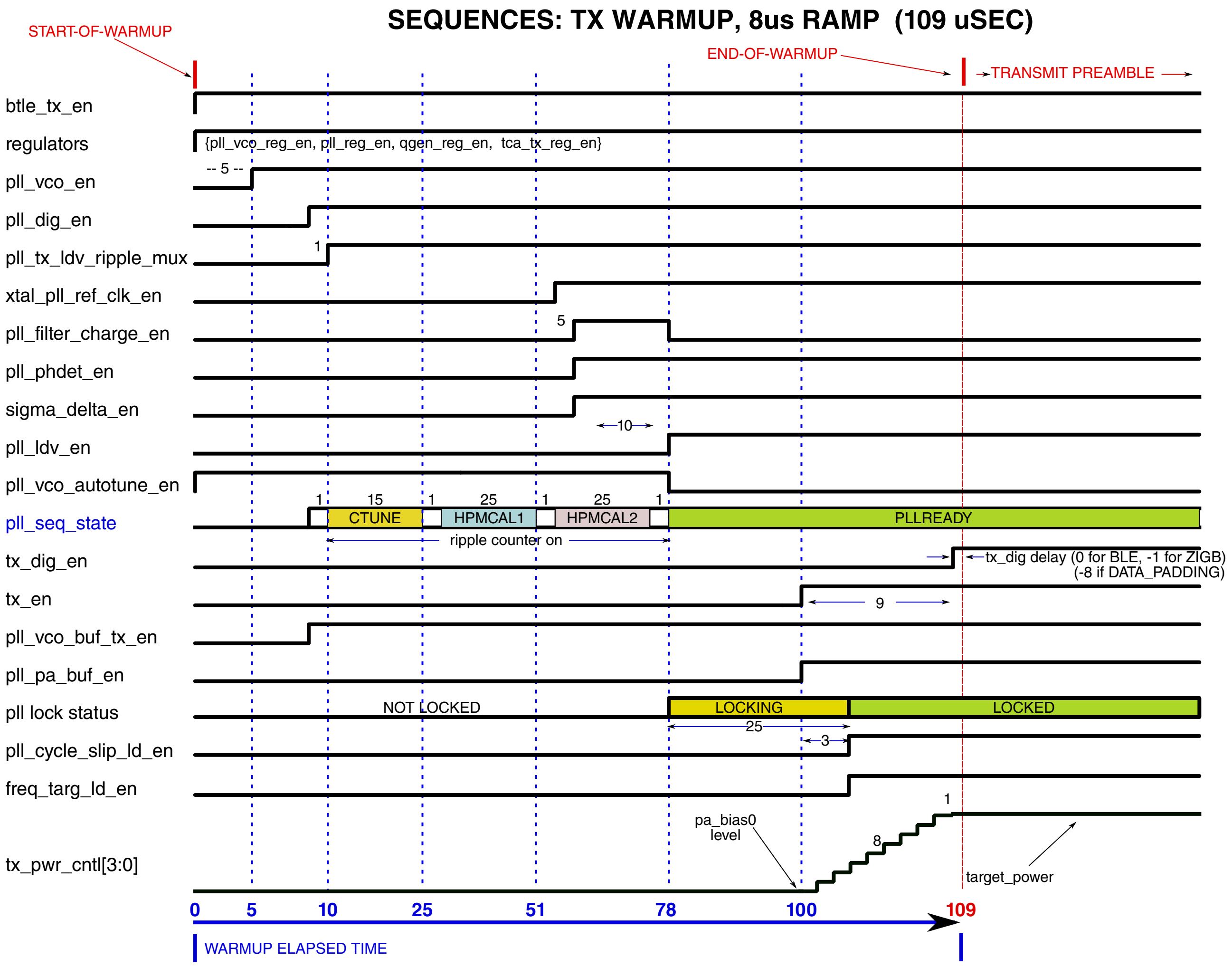
SEQUENCES: TX WARMUP, NO RAMP (103 uSEC)

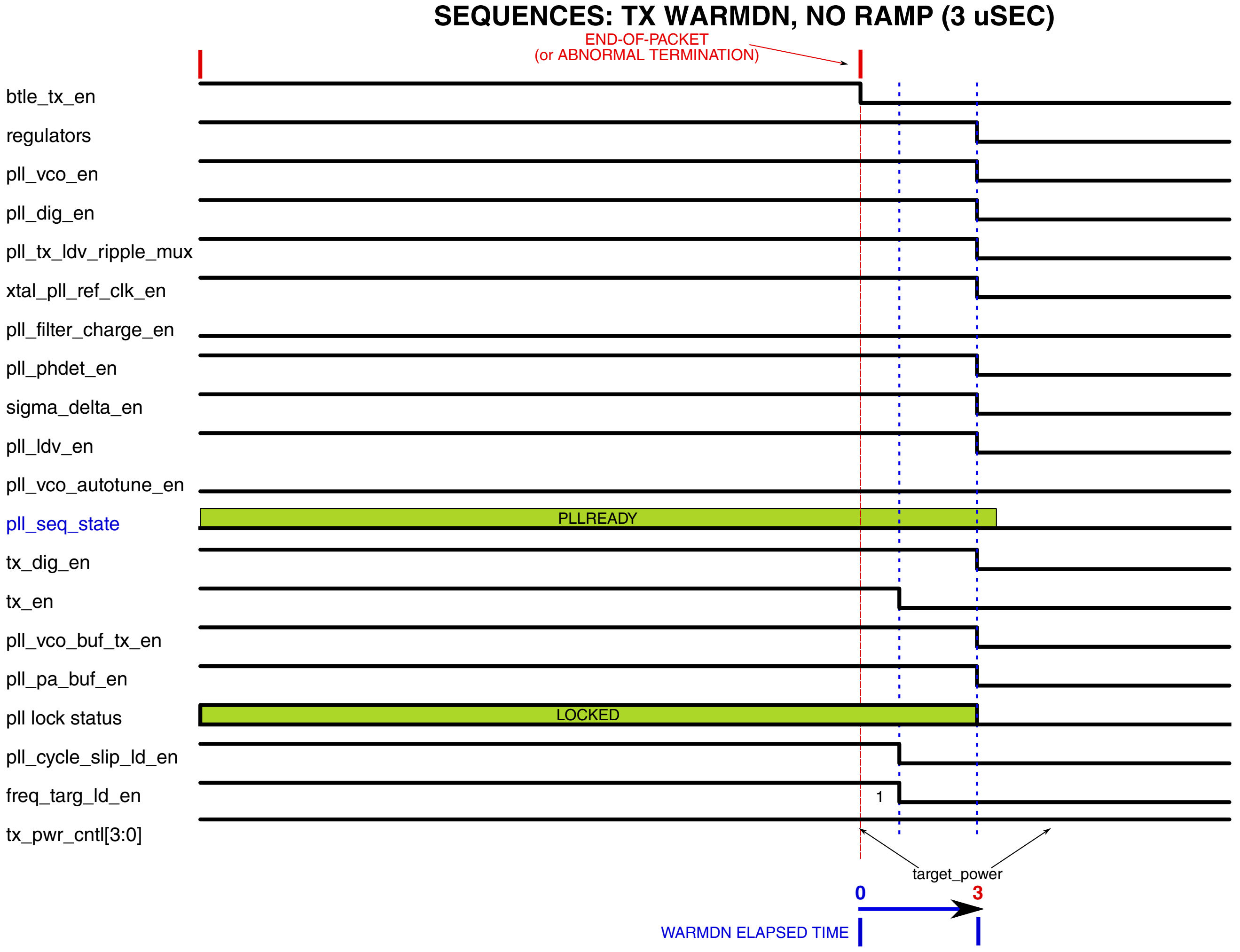


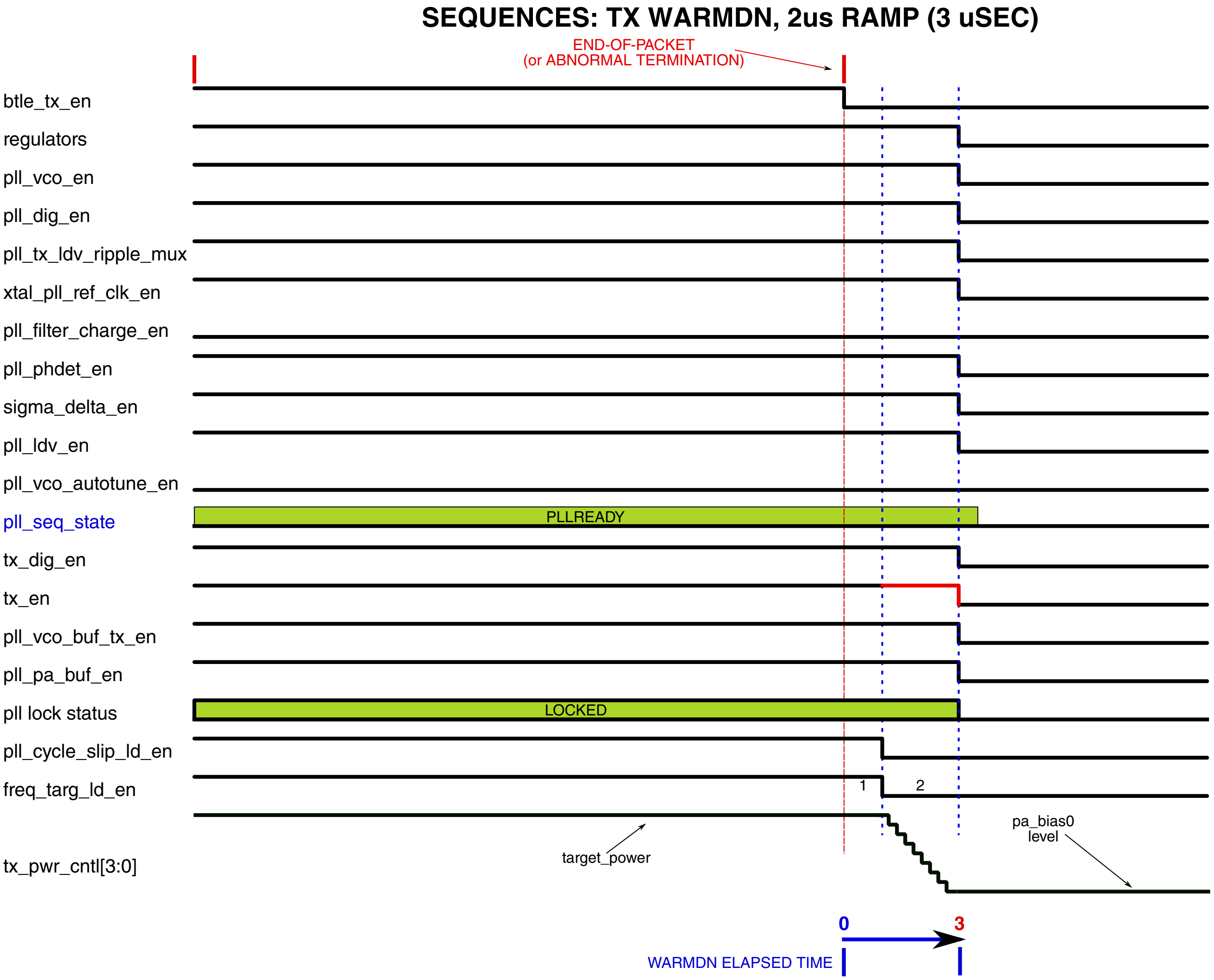

SEQUENCES: TX WARMUP, 2us RAMP (103 uSEC)
START-OF-WARMUP
END-OF-WARMUP
TRANSMIT PREAMBLE
btle_tx_en
regulators
{pll_vco_reg_en, pll_reg_en, qgen_reg_en, tca_tx_reg_en}
pll_vco_en
-- 5 --
pll_dig_en
pll_tx_ldv_ripple_mux
1
xtal_pll_ref_clk_en
pll_filter_charge_en
5
pll_phdet_en
sigma_delta_en
10
pll_ldv_en
pll_vco_autotune_en
pll_seq_state
1
15
1
25
1
25
1
CTUNE
HPMCAL1
HPMCAL2
PLLREADY
ripple counter on
tx_dig_en
tx_dig delay (0 for BLE, -1 for ZIGBEE)
(-8 if DATA_PADDING_EN)
tx_en
3
pll_vco_buf_tx_en
pll_pa_buf_en
pll lock status
NOT LOCKED
LOCKING
LOCKED
25
pll_cycle_slip_ld_en
3
freq_targ_ld_en
2
pa_bias0
level
2
tx_pwr_cntl[3:0]
target_power
0
5
10
25
51
78
100
103
WARMUP ELAPSED TIME

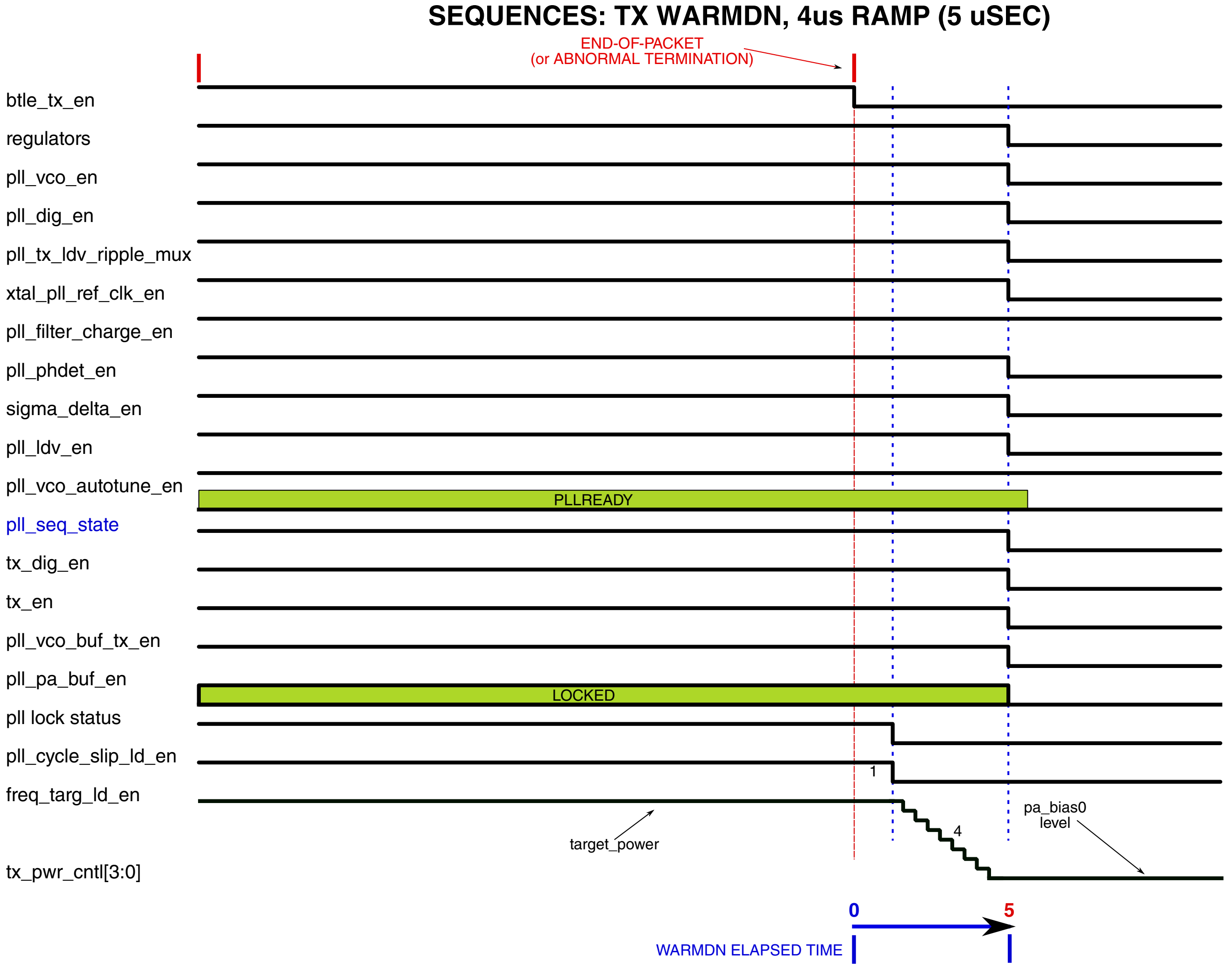


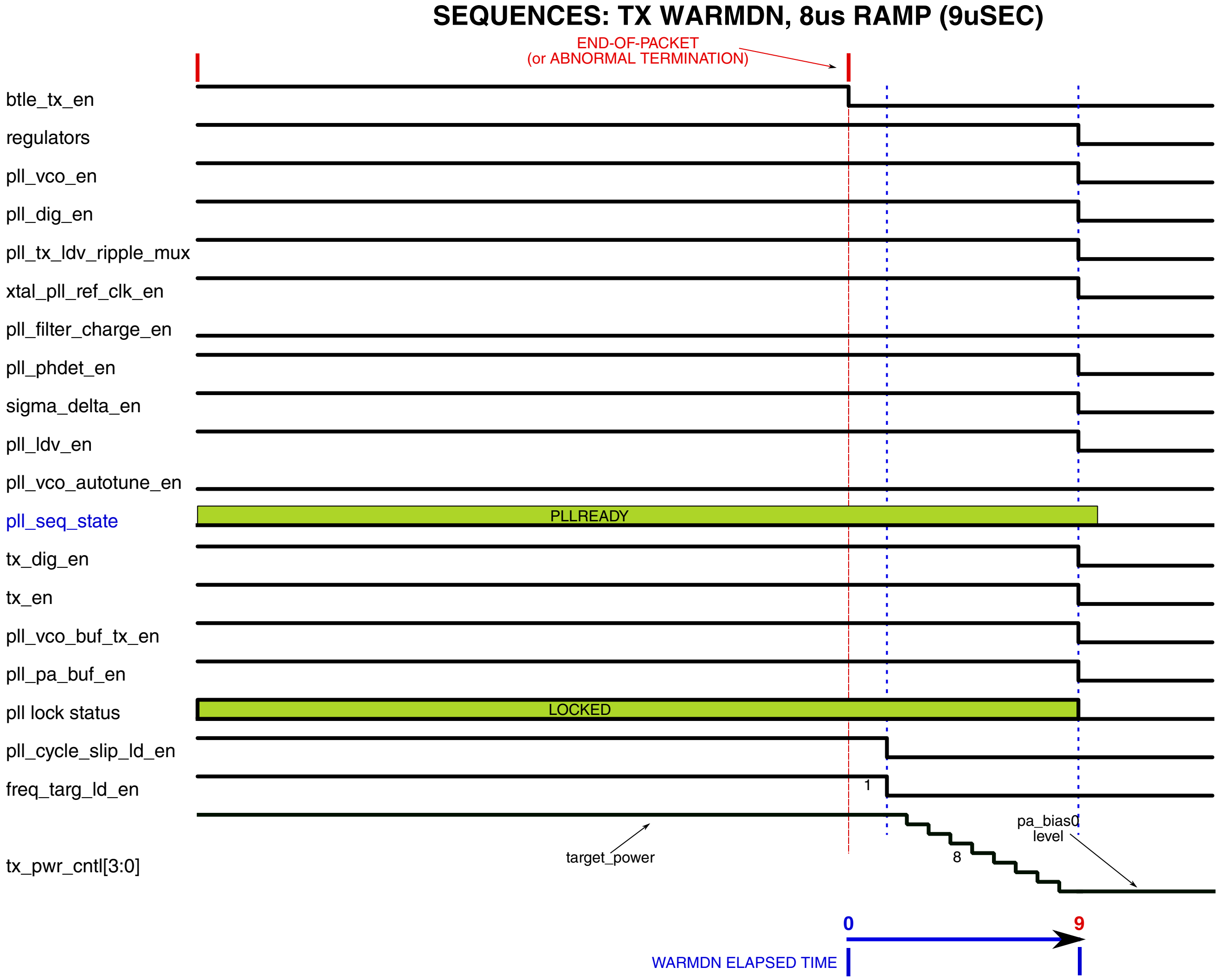
SEQUENCES: TX WARMUP, 4us RAMP (105 uSEC)

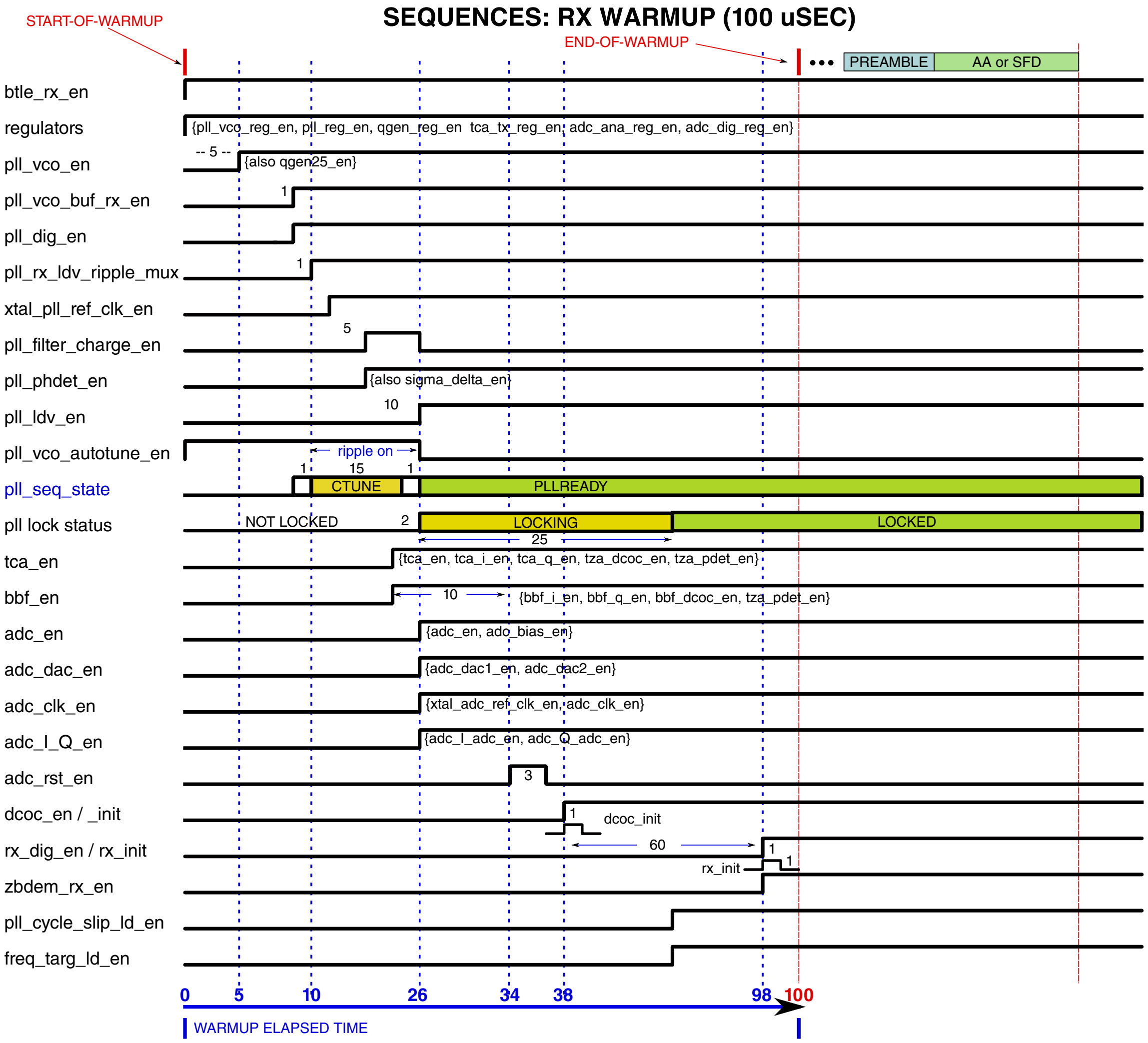

SEQUENCES: TX WARMUP, 8us RAMP (109 uSEC)
START-OF-WARMUP
END-OF-WARMUP
TRANSMIT PREAMBLE
btle_tx_en
regulators
{pll_vco_reg_en, pll_reg_en, qgen_reg_en, tca_tx_reg_en}
pll_vco_en
-- 5 --
pll_dig_en
pll_tx_ldv_ripple_mux
1
xtal_pll_ref_clk_en
pll_filter_charge_en
5
pll_phdet_en
sigma_delta_en
10
pll_ldv_en
pll_vco_autotune_en
pll_seq_state
1
15
1
25
1
25
1
CTUNE
HPMCAL1
HPMCAL2
PLLREADY
ripple counter on
tx_dig_en
tx_dig delay (0 for BLE, -1 for ZIGB)
(-8 if DATA_PADDING)
tx_en
9
pll_vco_buf_tx_en
pll_pa_buf_en
pll lock status
NOT LOCKED
LOCKING
LOCKED
25
3
pll_cycle_slip_ld_en
freq_targ_ld_en
tx_pwr_cntl[3:0]
pa_bias0
level
8
1
target_power
0
5
10
25
51
78
100
109
WARMUP ELAPSED TIME


SEQUENCES: TX WARMDN, NO RAMP (3 uSEC)
END-OF-PACKET
(or ABNORMAL TERMINATION)
btle_tx_en
regulators
pll_vco_en
pll_dig_en
pll_tx_ldv_ripple_mux
xtal_pll_ref_clk_en
pll_filter_charge_en
pll_phdet_en
sigma_delta_en
pll_ldv_en
pll_vco_autotune_en
pll_seq_state
PLLREADY
tx_dig_en
tx_en
pll_vco_buf_tx_en
pll_pa_buf_en
pll lock status
LOCKED
pll_cycle_slip_ld_en
freq_targ_ld_en
1
tx_pwr_cntl[3:0]
target_power
0
3
WARMDN ELAPSED TIME


SEQUENCES: TX WARMDN, 2us RAMP (3 uSEC)
END-OF-PACKET
(or ABNORMAL TERMINATION)
btle_tx_en
regulators
pll_vco_en
pll_dig_en
pll_tx_ldv_ripple_mux
xtal_pll_ref_clk_en
pll_filter_charge_en
pll_phdet_en
sigma_delta_en
pll_ldv_en
pll_vco_autotune_en
pll_seq_state
PLLREADY
tx_dig_en
tx_en
pll_vco_buf_tx_en
pll_pa_buf_en
pll lock status
LOCKED
pll_cycle_slip_ld_en
freq_targ_ld_en
1
2
tx_pwr_cntl[3:0]
target_power
pa_bias0
level
0
3
WARMDN ELAPSED TIME


SEQUENCES: TX WARMDN, 4us RAMP (5 uSEC)
END-OF-PACKET
(or ABNORMAL TERMINATION)
btle_tx_en
regulators
pll_vco_en
pll_dig_en
pll_tx_ldv_ripple_mux
xtal_pll_ref_clk_en
pll_filter_charge_en
pll_phdet_en
sigma_delta_en
pll_ldv_en
pll_vco_autotune_en
PLLREADY
pll_seq_state
tx_dig_en
tx_en
pll_vco_buf_tx_en
pll_pa_buf_en
LOCKED
pll lock status
pll_cycle_slip_ld_en
1
freq_targ_ld_en
target_power
4
pa_bias0
level
tx_pwr_cntl[3:0]
0
5
WARMDN ELAPSED TIME


SEQUENCES: TX WARMDN, 8us RAMP (9uSEC)
END-OF-PACKET
(or ABNORMAL TERMINATION)
btle_tx_en
regulators
pll_vco_en
pll_dig_en
pll_tx_ldv_ripple_mux
xtal_pll_ref_clk_en
pll_filter_charge_en
pll_phdet_en
sigma_delta_en
pll_ldv_en
pll_vco_autotune_en
pll_seq_state
PLLREADY
tx_dig_en
tx_en
pll_vco_buf_tx_en
pll_pa_buf_en
pll lock status
LOCKED
pll_cycle_slip_ld_en
freq_targ_ld_en
1
tx_pwr_cntl[3:0]
target_power
8
pa_bias0
level
0
9
WARMDN ELAPSED TIME


SEQUENCES: RX WARMUP (100 uSEC)
START-OF-WARMUP
END-OF-WARMUP
•••
PREAMBLE
AA or SFD
btle_rx_en
regulators
{pll_vco_reg_en, pll_reg_en, qgen_reg_en tca_tx_reg_en, adc_ana_reg_en, adc_dig_reg_en}
pll_vco_en
-- 5 --
{also qgen25_en}
pll_vco_buf_rx_en
1
pll_dig_en
pll_rx_ldv_ripple_mux
1
xtal_pll_ref_clk_en
pll_filter_charge_en
5
pll_phdet_en
{also sigma_delta_en}
pll_ldv_en
10
pll_vco_autotune_en
ripple on
1
15
1
pll_seq_state
CTUNE
PLLREADY
pll lock status
NOT LOCKED
2
LOCKING
LOCKED
25
tca_en
{tca_en, tca_i_en, tca_q_en, tza_dcoc_en, tza_pdet_en}
bbf_en
10
{bbf_i_en, bbf_q_en, bbf_dcoc_en, tza_pdet_en}
adc_en
{adc_en, adc_bias_en}
adc_dac_en
{adc_dac1_en, adc_dac2_en}
adc_clk_en
{xtal_adc_ref_clk_en, adc_clk_en}
adc_I_Q_en
{adc_I_adc_en, adc_Q_adc_en}
adc_rst_en
3
dcoc_en / _init
1
dcoc_init
rx_dig_en / rx_init
60
1
rx_init
1
zbdem_rx_en
pll_cycle_slip_ld_en
freq_targ_ld_en
0
5
10
26
34
38
98
100
WARMUP ELAPSED TIME

50.4.15 Clocks

The TSM uses 4 clocks:

1. ipg_clk
2. tsm_clk
3. tsm_dly_clk
4. tsm_4m_clk

The “**ipg_ck”,** is a gated 32mhz clock from the reference oscillator. This clock only used to re-clock the tx_pwr_cntl[3:0] output, to ensure no glitching or skew between the bits, on this power control bus to the PA.

The main clock is the 1MHz **tsm_clk**. This clock establishes the TSM timebase (1us), and runs the TSM counter, as well as the control and abort-handling logic.

Another 1MHz clock is **tsm_dly_clk**. This clock is used to re-clock the 56 TSM-controlled outputs. This clock is time-shifted (delayed) by 1 ipg_clk cycle relative to tsm_clk.

The 4Mhz clock **tsm_4m_clk** runs the PA ramping logic.

All 4 TSM clocks derive from the same 32MHz source, the reference oscillator. They are divided down from this reference frequency in the CRM (Clocks and Resets Module). Since all 4 are in the same clock domain, they will be balanced (skew-controlled) during clock-tree synthesis. There there are no clock-domain-crossings or asynchronous interfaces in the TSM.

50.4.16 Reset

The TSM has a single, active-low, asynchronous reset input: **ipg_hard_async_reset_b** . At integration, this reset should be tied to the master reset for the entire transceiver. There are no special reset requirements.

50.4.17 Interrupts

This TSM generates no interrupts.

Chapter 51
Oscillator

51.1 About this module

51.1.1 Introduction

This document describes the XTAL32m design. This oscillator is based on the Pierce oscillator configuration, in which the voltage gain or the transconductance of the amplifier is controlled by the bias current. Figure 51-1 shows the crystal oscillator input/output signals.

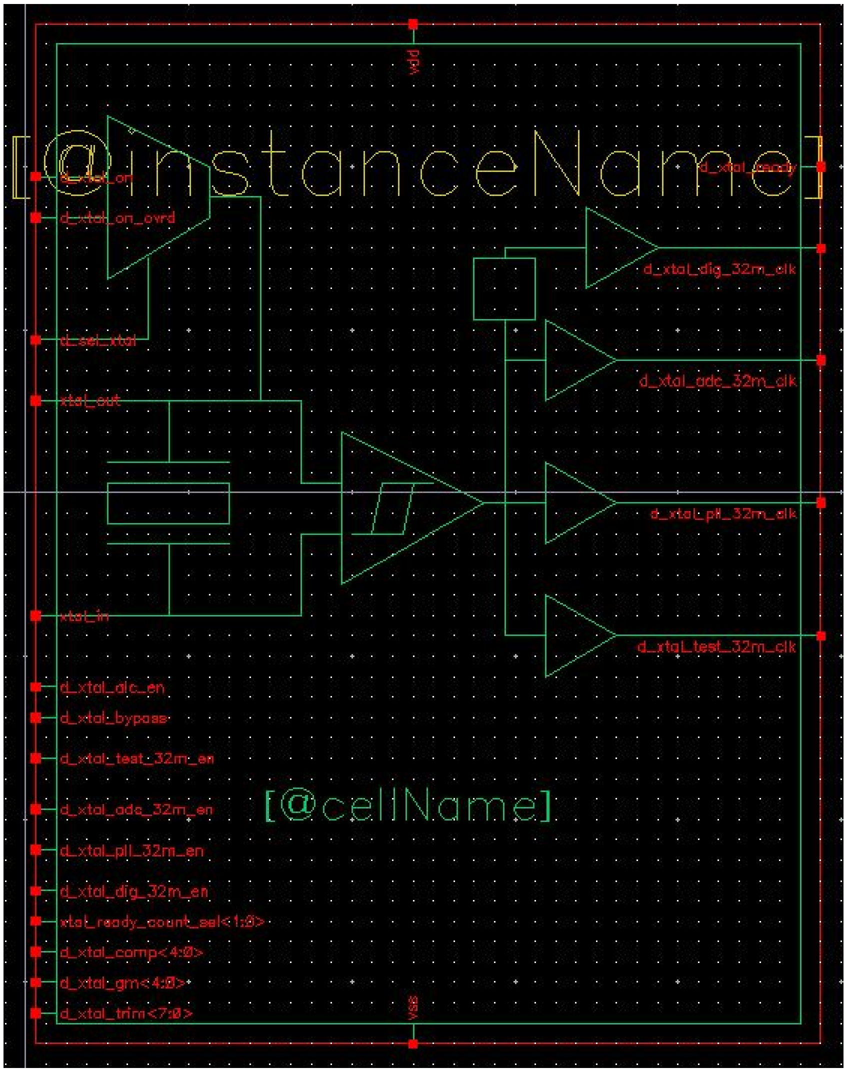


Figure 51-1. Symbol

A simplified version of the top level schematic is shown in Figure 51-2. It contains two main parts, the Xtal core block and the digital sub-blocks around it. The core block contains the crystal oscillator driver itself and its associated sub-blocks. The main digital sub-blocks are a timer and trim capacitor which will be discussed later.

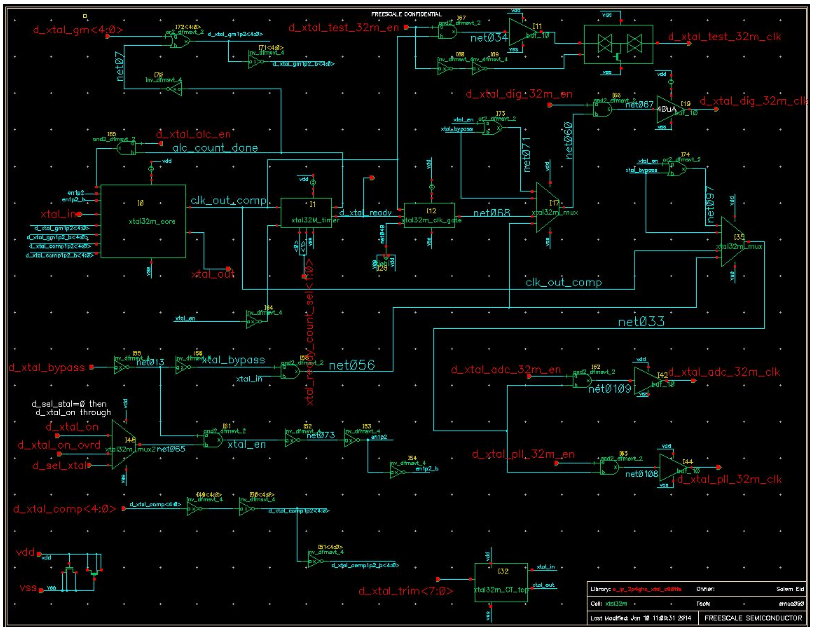


Figure 51-2. Top Level Schematic

51.1.2 Features and Electrical Specifications

- Supply voltage: 1.2V typ
- Temperature range -40C, 125C
- Pierce oscillator type
- Automatic level control (ALC) to control the oscillator sinusoidal amplitude
- 32MHz crystal operation
- Output buffer duty cycle: 50 ± 15%
- Startup time 256us, a kick circuit is implemented to ensure quick startup
- Phase noise: -126 dBc/Hz from 10KHz and higher
- Xtal ready signal is provided to indicate the crystal clock is valid
- A bypass mode: disables the XTAL32M circuit, allows external clock injection
- Trim capacitor circuit (8-bit) to adjusts the crystal load capacitors (18fF/LSB)
- Crystal model used, 32MHz, Lx=13.2mH, Cx=1.9fF, Rx=60Ω max with Cshunt = 0.6pF
- Crystal resonator load capacitance 9PF typical, package shunt capacitance 2.8PF

51.1.3 Signals

The table below describes the signals on the boundary of the module.

Signal	I/O	Function
xtal_in	I	Crystal connection, to be connected to pad EXTAL32M
xtal_out	O	Crystal connection, to be connected to pad Xtal32M
vdd	I/O	Analog supply. 1.2V
vss	I/O	Analog ground, connected to XTAL32m_GND
d_xtal_comp<4:0>	I	Control switches for current level in the comparator, default value is 01010
d_xtal_gm<4:0>	I	Defines the gm value for the crystal driver (typ: 00011)
d_xtal_bypass	I	XTAL32m bypass mode 0 – normal operating mode (crystal is connected). 1 – bypass mode, external 32MHz clock referenced to VBAT is applied to EXTAL32M pin. The XTAL32m block should be disabled (d_xtal_en=0)
d_xtal_on	I	XTAL32m enable as well as Band Gap 0 – disabled, 1- oscillator block is enabled XTAL32m should be disabled (=0) for the bypass mode to work
d_xtal_on_ovrd	I	xtal_on_overdrive
d_sel_xtal	I	Xtal Enable Select 0 – d_xtal_on used as enable, 1 – d_xtal_on_ovrd used a enable
d_xtal_ready	O	32MHz clock ready signal sent to the digital. 0 – clock is not ready, 1 – Clock is valid and active.
d_xtal_alc_en	I	Enable the Automatic Level Control (ALC) 0 – disabled, 1 – ALC is enabled
d_xtal_trim<7:0>	I	Crystal capacitor trim. 18fF per LSB
d_xtal_test_32m_clk	O	32MHz test clock, special low jitter route to CLK_OUT mux
d_xtal_test_32m_en	I	0 – disable, 1 – enable test clock to mux at padring
d_xtal_dig_32m_clk	O	clock for digital baseband
d_xtal_dig_32m_en	I	enable digital clock
d_xtal_adc_32m_clk	O	clock for ADC
d_xtal_adc_32m_en	I	ADC clock enable
d_xtal_pll_32m_clk	O	clock for PLL
d_xtal_pll_32m_clk_b	O	180° out of phase clock for PLL
d_xtal_pll_32m_en	I	PLL clock enable
xtal_ready_count_sel<1:0>	I	Selects the number of clock counts after which xtal_ready goes high - 00 after 1024 clock counts, 01 after 2048 counts, 10 after 4096 counts, and 11 after 8192 count
sel_alc_256	I	Controls when the alc is done, 0 gives 256uS delay or 8192 clk count, 1 gives 512uS or 16384 clk count
iso	I	0 - normal operating mode, 1 - ISO mode: XTAL regulator turned OFF (This is an input to the low-to-high levelshifters indicating that the lower supply is not available)

51.2 Major building blocks

51.2.1 Major building blocks

The major blocks that constitute the XTAL32m are the core (xtal32m_core) and the digital sub-circuits (xtal_timer & xtal32m_CT_top).

51.2.2 xtal32m_core circuit

The simplified schematic of xtal32m_core block is shown in the figure below. It consists of xtal32m_amp, xtal32m_bias_gen, xtal32m_alc, xtal32m_comp, and xtal32m_kick circuits. The power domain of this block is 1.2V vbat voltage.

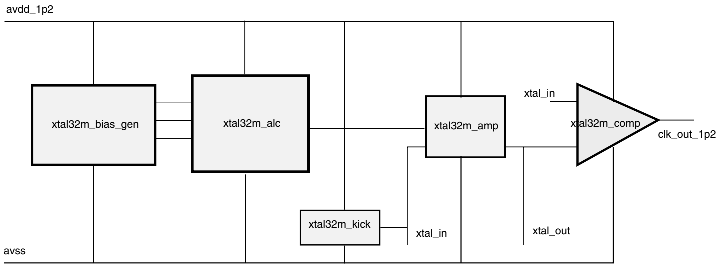


Figure 51-3. xtal32m_core Schematic

51.2.3 xtal32m_amp circuit

The amplifier designed is of a transconductance type. It consists of a large NMOS device biased by a current mirror. The source of the current is the amplitude level control (ALC) circuit. There is a 5-bit register “d_xtal_gm1p2<4:0>” that can modify the current of the transconductance transistor. This capability is added so that the current can be selected to satisfy the various crystals (external components) that could be used. The default value for the register is (00011) which would provide around 100uA to the transconductance transistor. The current for each setting is around:

Setting	Current
00000	25uA
00001	50uA
00011	100uA
00111	200uA
01111	350uA
11111	550uA

51.2.4 xtal32m_alc circuit

The ALC loop reduces the current in the transconductor amplifier in response to an increase of the oscillation amplitude, developed at "ac_input" pin (connected to xtal_in at top). Thus, the gain of the transconductor amplifier is constantly controlled to maintain a certain oscillation level (typical 300mVpp). The inputs to this block are the xtal_in (“ac_input”) and a filtered version (“reference”) of this signal. The “ac_input” drives the gate of an envelope detector, operating on the negative peak of ac_input signal. The “reference” signal is a filtered "ac_input" signal so only the dc signal is extracted. The output of this block is “pbias” which drives the mirror in the amplifier block.

51.2.5 xtal32m_comp circuit

The inputs to the comparator are the "xtal_out" and "xtal_in" signals which constitute the voltage across the crystal. This differential sinusoidal signal is converted to a single ended digital signal “out_comp”. The bias current for the comparator is programmable by “d_xtal_comp1p2<4:0>” register. There are two modes of operation: high performance (low phase noise, high current) and a low power mode. The default values are “01111” for high performance and “00011” for low power mode when only the clock is needed to drive the digital see-of-gates. The current levels for the current source are:

Setting	Current
00000	4uA
00001	6uA
00011	10uA
00111	18uA
01111	30uA
11111	52uA

51.2.6 xtal32m_bias_gen circuit

This bias circuit is a direct port from the Havasu project crystal oscillator driver circuit. This bias circuit is a self-bias circuit where the current is generated as delta(Vgs) voltage over a resistor. This bias circuit provides all the currents needed in the crystal circuit.

51.2.7 xtal32m_kick circuit

The kick-off circuit provides a startup current pulse in the crystal resonator branch. The kick-off circuit concept is to add a switch between "xtal_in" pad and Vdd, and close the switch during a time close to one period of the resonator frequency. The startup pulse starts at the onset of “en1p2_b” signal, that is as the enable signal goes from high to low, the output PMOS device will be ON for the pulse duration mainly determined by the capacitor at the input of the NOR gate.

51.2.8 xtal32M_timer_3v circuit

Figure below shows the timer circuit. This timer circuit controls the 32MHz clock gate that goes to the digital sea-of-gates. The idea is as the crystal sinusoidal signal is building amplitude, the comparator output will not have a stable clock. The counter is designed to count 1024/2048/4096/8192 cycles (@32MHz) before the clock is released to the digital.

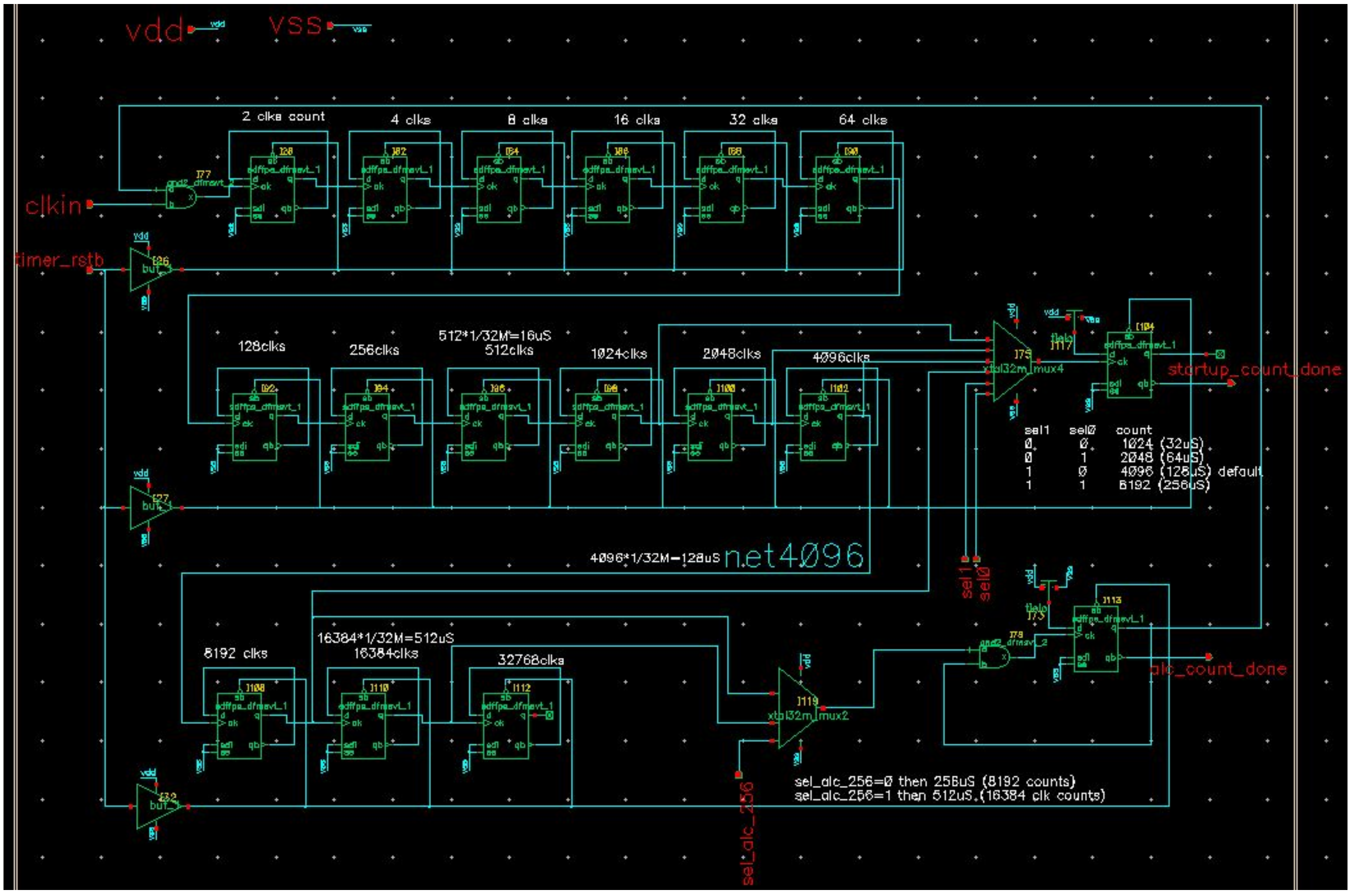


Figure 51-4. xtal32M_timer_3v Schematic

51.2.9 xtal32m_CT_top circuit

The xtal32m_CT_top block comprise of two xtal32m_CT capacitor banks as shown in the figure below. Each of these xtal32m_CT capacitor bank is connected to the crystal oscillator pins (xtal_in and xtal_out). The total capacitance of the capacitor bank is around 4.6pF with 8-bit trim register (d_xtal_trim<7:0>) and with default value of “01110111”. This default value corresponds to a 2.3pF capacitance for the capacitor bank. The LSB capacitor value is around 18fF. This LSB value corresponds to a frequency shift of 7Hz from the 32MHz crystal oscillator frequency.

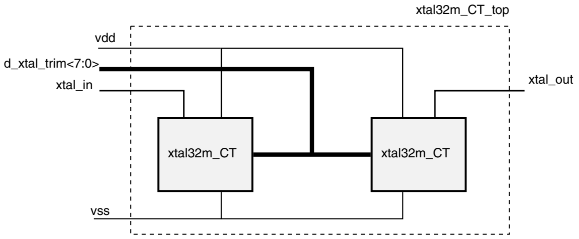


Figure 51-5. xtal32m_CT_top Schematic

Chapter 52
Transceiver Registers

52.1 Transceiver Memory map/register definition

This section describes the registers and data structures in the Transceiver module as instantiated in the **Radio**. The base address of the module depends on the particular memory map of the MCU. The addresses presented here are absolute addresses.

XCVR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C000	RX Digital Control (XCVR_RX_DIG_CTRL)	32	R/W	0000_0000h	52.1.1/1097
4005_C004	AGC Control 0 (XCVR_AGC_CTRL_0)	32	R/W	0000_0000h	52.1.2/1099
4005_C008	AGC Control 1 (XCVR_AGC_CTRL_1)	32	R/W	0000_0000h	52.1.3/1100
4005_C00C	AGC Control 2 (XCVR_AGC_CTRL_2)	32	R/W	0000_0000h	52.1.4/1101
4005_C010	AGC Control 3 (XCVR_AGC_CTRL_3)	32	R/W	0000_0000h	52.1.5/1103
4005_C014	AGC Status (XCVR_AGC_STAT)	32	R	0000_0000h	52.1.6/1104
4005_C018	RSSI Control 0 (XCVR_RSSI_CTRL_0)	32	R/W	0000_0000h	52.1.7/1105
4005_C01C	RSSI Control 1 (XCVR_RSSI_CTRL_1)	32	R/W	0000_0000h	52.1.8/1107
4005_C020	DCOC Control 0 (XCVR_DCOC_CTRL_0)	32	R/W	0000_0000h	52.1.9/1107
4005_C024	DCOC Control 1 (XCVR_DCOC_CTRL_1)	32	R/W	0000_0000h	52.1.10/ 1109
4005_C028	DCOC Control 2 (XCVR_DCOC_CTRL_2)	32	R/W	0000_0000h	52.1.11/ 1110
4005_C02C	DCOC Control 3 (XCVR_DCOC_CTRL_3)	32	R/W	0000_0000h	52.1.12/ 1111
4005_C030	DCOC Control 4 (XCVR_DCOC_CTRL_4)	32	R/W	0000_0000h	52.1.13/ 1111
4005_C034	DCOC Calibration Gain (XCVR_DCOC_CAL_GAIN)	32	R/W	0000_0000h	52.1.14/ 1112
4005_C038	DCOC Status (XCVR_DCOC_STAT)	32	R	See section	52.1.15/ 1113
4005_C03C	DCOC DC Estimate (XCVR_DCOC_DC_EST)	32	R	0000_0000h	52.1.16/ 1113

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C040	DCOC Calibration Reciprocals (XCVR_DCOC_CAL_RCP)	32	R/W	0000_0000h	52.1.17/ 1114
4005_C04C	IQMC Control (XCVR_IQMC_CTRL)	32	R/W	See section	52.1.18/ 1115
4005_C050	IQMC Calibration (XCVR_IQMC_CAL)	32	R/W	See section	52.1.19/ 1115
4005_C054	TCA AGC Step Values 3..0 (XCVR_TCA_AGC_VAL_3_0)	32	R/W	See section	52.1.20/ 1116
4005_C058	TCA AGC Step Values 7..4 (XCVR_TCA_AGC_VAL_7_4)	32	R/W	See section	52.1.21/ 1116
4005_C05C	TCA AGC Step Values 8 (XCVR_TCA_AGC_VAL_8)	32	R/W	See section	52.1.22/ 1117
4005_C060	BBF Resistor Tune Values 7..0 (XCVR_BBF_RES_TUNE_VAL_7_0)	32	R/W	0000_0000h	52.1.23/ 1117
4005_C064	BBF Resistor Tune Values 10..8 (XCVR_BBF_RES_TUNE_VAL_10_8)	32	R/W	0000_0000h	52.1.24/ 1118
4005_C068	TCA AGC Linear Gain Values 2..0 (XCVR_TCA_AGC_LIN_VAL_2_0)	32	R/W	0000_0000h	52.1.25/ 1119
4005_C06C	TCA AGC Linear Gain Values 5..3 (XCVR_TCA_AGC_LIN_VAL_5_3)	32	R/W	0000_0000h	52.1.26/ 1119
4005_C070	TCA AGC Linear Gain Values 8..6 (XCVR_TCA_AGC_LIN_VAL_8_6)	32	R/W	0000_0000h	52.1.27/ 1120
4005_C074	BBF Resistor Tune Values 3..0 (XCVR_BBF_RES_TUNE_LIN_VAL_3_0)	32	R/W	0000_0000h	52.1.28/ 1121
4005_C078	BBF Resistor Tune Values 7..4 (XCVR_BBF_RES_TUNE_LIN_VAL_7_4)	32	R/W	0000_0000h	52.1.29/ 1121
4005_C07C	BBF Resistor Tune Values 10..8 (XCVR_BBF_RES_TUNE_LIN_VAL_10_8)	32	R/W	0000_0000h	52.1.30/ 1122
4005_C080	AGC Gain Tables Step 03..00 (XCVR_AGC_GAIN_TBL_03_00)	32	R/W	0000_0000h	52.1.31/ 1123
4005_C084	AGC Gain Tables Step 07..04 (XCVR_AGC_GAIN_TBL_07_04)	32	R/W	0000_0000h	52.1.32/ 1123
4005_C088	AGC Gain Tables Step 11..08 (XCVR_AGC_GAIN_TBL_11_08)	32	R/W	0000_0000h	52.1.33/ 1124
4005_C08C	AGC Gain Tables Step 15..12 (XCVR_AGC_GAIN_TBL_15_12)	32	R/W	0000_0000h	52.1.34/ 1125
4005_C090	AGC Gain Tables Step 19..16 (XCVR_AGC_GAIN_TBL_19_16)	32	R/W	0000_0000h	52.1.35/ 1126
4005_C094	AGC Gain Tables Step 23..20 (XCVR_AGC_GAIN_TBL_23_20)	32	R/W	0000_0000h	52.1.36/ 1127
4005_C098	AGC Gain Tables Step 26..24 (XCVR_AGC_GAIN_TBL_26_24)	32	R/W	0000_0000h	52.1.37/ 1127
4005_C0A0	DCOC Offset (XCVR_DCOC_OFFSET_00)	32	R/W	0000_0000h	52.1.38/ 1128

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C0A4	DCOC Offset (XCVR_DCOC_OFFSET_01)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0A8	DCOC Offset (XCVR_DCOC_OFFSET_02)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0AC	DCOC Offset (XCVR_DCOC_OFFSET_03)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0B0	DCOC Offset (XCVR_DCOC_OFFSET_04)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0B4	DCOC Offset (XCVR_DCOC_OFFSET_05)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0B8	DCOC Offset (XCVR_DCOC_OFFSET_06)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0BC	DCOC Offset (XCVR_DCOC_OFFSET_07)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0C0	DCOC Offset (XCVR_DCOC_OFFSET_08)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0C4	DCOC Offset (XCVR_DCOC_OFFSET_09)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0C8	DCOC Offset (XCVR_DCOC_OFFSET_10)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0CC	DCOC Offset (XCVR_DCOC_OFFSET_11)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0D0	DCOC Offset (XCVR_DCOC_OFFSET_12)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0D4	DCOC Offset (XCVR_DCOC_OFFSET_13)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0D8	DCOC Offset (XCVR_DCOC_OFFSET_14)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0DC	DCOC Offset (XCVR_DCOC_OFFSET_15)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0E0	DCOC Offset (XCVR_DCOC_OFFSET_16)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0E4	DCOC Offset (XCVR_DCOC_OFFSET_17)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0E8	DCOC Offset (XCVR_DCOC_OFFSET_18)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0EC	DCOC Offset (XCVR_DCOC_OFFSET_19)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0F0	DCOC Offset (XCVR_DCOC_OFFSET_20)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0F4	DCOC Offset (XCVR_DCOC_OFFSET_21)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C0F8	DCOC Offset (XCVR_DCOC_OFFSET_22)	32	R/W	0000_0000h	52.1.38/ 1128

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C0FC	DCOC Offset (XCVR_DCOC_OFFSET_23)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C100	DCOC Offset (XCVR_DCOC_OFFSET_24)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C104	DCOC Offset (XCVR_DCOC_OFFSET_25)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C108	DCOC Offset (XCVR_DCOC_OFFSET_26)	32	R/W	0000_0000h	52.1.38/ 1128
4005_C110	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_00)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C114	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_01)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C118	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_02)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C11C	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_03)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C120	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_04)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C124	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_05)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C128	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_06)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C12C	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_07)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C130	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_08)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C134	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_09)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C138	DCOC TZA DC step (XCVR_DCOC_TZA_STEP_10)	32	R/W	0000_0000h	52.1.39/ 1129
4005_C16C	DCOC Calibration Alpha (XCVR_DCOC_CAL_ALPHA)	32	R	0000_0000h	52.1.40/ 1129
4005_C170	DCOC Calibration Beta (XCVR_DCOC_CAL_BETA)	32	R	0000_0000h	52.1.41/ 1130
4005_C174	DCOC Calibration Gamma (XCVR_DCOC_CAL_GAMMA)	32	R	0000_0000h	52.1.42/ 1130
4005_C178	DCOC Calibration IIR (XCVR_DCOC_CAL_IIR)	32	R/W	0000_0000h	52.1.43/ 1131
4005_C180	DCOC Calibration Result (XCVR_DCOC_CAL1)	32	R	0000_0000h	52.1.44/ 1132
4005_C184	DCOC Calibration Result (XCVR_DCOC_CAL2)	32	R	0000_0000h	52.1.44/ 1132
4005_C188	DCOC Calibration Result (XCVR_DCOC_CAL3)	32	R	0000_0000h	52.1.44/ 1132

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C1A0	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF0)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1A4	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF1)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1A8	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF2)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1AC	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF3)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1B0	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF4)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1B4	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF5)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1B8	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF6)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C1BC	Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF7)	32	R/W	0000_0000h	52.1.45/ 1133
4005_C200	TX Digital Control (XCVR_TX_DIG_CTRL)	32	R/W	0000_0140h	52.1.46/ 1133
4005_C204	TX Data Padding Pattern (XCVR_TX_DATA_PAD_PAT)	32	R/W	7FFF_55AAh	52.1.47/ 1136
4005_C208	TX GFSK Modulation Control (XCVR_TX_GFSK_MOD_CTRL)	32	R/W	0301_4000h	52.1.48/ 1137
4005_C20C	TX GFSK Filter Coefficients 2 (XCVR_TX_GFSK_COEFF2)	32	R/W	C063_0401h	52.1.49/ 1138
4005_C210	TX GFSK Filter Coefficients 1 (XCVR_TX_GFSK_COEFF1)	32	R/W	BB29_960Dh	52.1.50/ 1139
4005_C214	TX FSK Modulation Scale (XCVR_TX_FSK_MOD_SCALE)	32	R/W	07FF_1800h	52.1.51/ 1140
4005_C218	TX DFT Modulation Pattern (XCVR_TX_DFT_MOD_PAT)	32	R/W	0000_0000h	52.1.52/ 1140
4005_C21C	TX DFT Tones 0 and 1 (XCVR_TX_DFT_TONE_0_1)	32	R/W	1000_0FFFh	52.1.53/ 1141
4005_C220	TX DFT Tones 2 and 3 (XCVR_TX_DFT_TONE_2_3)	32	R/W	1E00_01FFh	52.1.54/ 1141
4005_C228	PLL Modulation Overrides (XCVR_PLL_MOD_OVRD)	32	R/W	0000_0000h	52.1.55/ 1142
4005_C22C	PLL Channel Mapping (XCVR_PLL_CHAN_MAP)	32	R/W	0000_0200h	52.1.56/ 1143
4005_C230	PLL Lock Detect (XCVR_PLL_LOCK_DETECT)	32	R/W	0020_2600h	52.1.57/ 1146
4005_C234	PLL High Port Modulation Control (XCVR_PLL_HP_MOD_CTRL)	32	R/W	0084_0000h	52.1.58/ 1148
4005_C238	PLL HPM Calibration Control (XCVR_PLL_HPM_CAL_CTRL)	32	R	4000_02A2h	52.1.59/ 1150

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C23C	PLL Cycle Slip Lock Detect Configuration and HPM Calibration 1 (XCVR_PLL_LD_HPM_CAL1)	32	R	4430_0000h	52.1.60/ 1151
4005_C240	PLL Cycle Slip Lock Detect Configuration and HPM Calibration 2 (XCVR_PLL_LD_HPM_CAL2)	32	R	0210_0000h	52.1.61/ 1153
4005_C244	PLL HPM SDM Fraction (XCVR_PLL_HPM_SDM_FRACTION)	32	R/W	01FF_0000h	52.1.62/ 1154
4005_C248	PLL Low Port Modulation Control (XCVR_PLL_LP_MOD_CTRL)	32	R/W	0808_0000h	52.1.63/ 1155
4005_C24C	PLL Low Port SDM Control 1 (XCVR_PLL_LP_SDM_CTRL1)	32	R/W	0026_0026h	52.1.64/ 1157
4005_C250	PLL Low Port SDM Control 2 (XCVR_PLL_LP_SDM_CTRL2)	32	R/W	0200_0000h	52.1.65/ 1158
4005_C254	PLL Low Port SDM Control 3 (XCVR_PLL_LP_SDM_CTRL3)	32	R/W	0400_0000h	52.1.66/ 1158
4005_C258	PLL Low Port SDM Numerator Applied (XCVR_PLL_LP_SDM_NUM)	32	R	0E20_0000h	52.1.67/ 1159
4005_C25C	PLL Low Port SDM Denominator Applied (XCVR_PLL_LP_SDM_DENOM)	32	R	0400_0000h	52.1.68/ 1159
4005_C260	PLL Delay Matching (XCVR_PLL_DELAY_MATCH)	32	R/W	0000_0201h	52.1.69/ 1160
4005_C264	PLL Coarse Tune Control (XCVR_PLL_CTUNE_CTRL)	32	R/W	0000_0000h	52.1.70/ 1161
4005_C268	PLL Coarse Tune Count 6 (XCVR_PLL_CTUNE_CNT6)	32	R	0000_0000h	52.1.71/ 1162
4005_C26C	PLL Coarse Tune Counts 5 and 4 (XCVR_PLL_CTUNE_CNT5_4)	32	R	0000_0000h	52.1.72/ 1162
4005_C270	PLL Coarse Tune Counts 3 and 2 (XCVR_PLL_CTUNE_CNT3_2)	32	R	0000_0000h	52.1.73/ 1163
4005_C274	PLL Coarse Tune Counts 1 and 0 (XCVR_PLL_CTUNE_CNT1_0)	32	R	0000_0000h	52.1.74/ 1163
4005_C278	PLL Coarse Tune Results (XCVR_PLL_CTUNE_RESULTS)	32	R	0962_0040h	52.1.75/ 1164
4005_C280	Transceiver Control (XCVR_XCVR_CTRL)	32	R/W	See section	52.1.76/ 1165
4005_C284	Transceiver Status (XCVR_XCVR_STATUS)	32	R	See section	52.1.77/ 1167
4005_C288	Soft Reset (XCVR_SOFT_RESET)	32	R	0000_0000h	52.1.78/ 1169
4005_C290	Overwrite Version (XCVR_OVERWRITE_VER)	32	R/W	0000_0000h	52.1.78/ 1170
4005_C294	DMA Control (XCVR_DMA_CTRL)	32	R/W	0000_0000h	52.1.79/ 1171
4005_C298	DMA Data (XCVR_DMA_DATA)	32	R	0000_0000h	52.1.80/ 1172

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C29C	Digital Test Control (XCVR_DTEST_CTRL)	32	R/W	0000_0000h	52.1.81/ 1172
4005_C2A0	Packet Buffer Control Register (XCVR_PB_CTRL)	32	R/W	0000_0000h	52.1.82/ 1175
4005_C2C0	Transceiver Sequence Manager Control (XCVR_TSM_CTRL)	32	R/W	FF00_0000h	52.1.83/ 1176
4005_C2C4	End of Sequence Control (XCVR_END_OF_SEQ)	32	R/W	6564_6A67h	52.1.84/ 1178
4005_C2C8	TSM Override 0 (XCVR_TSM_OVRD0)	32	R/W	0000_0000h	52.1.85/ 1179
4005_C2CC	TSM Override 1 (XCVR_TSM_OVRD1)	32	R/W	0000_0000h	52.1.86/ 1182
4005_C2D0	TSM Override 2 (XCVR_TSM_OVRD2)	32	R/W	0000_0000h	52.1.87/ 1186
4005_C2D4	TSM Override 3 (XCVR_TSM_OVRD3)	32	R/W	0000_0000h	52.1.88/ 1189
4005_C2D8	PA Power (XCVR_PA_POWER)	32	R/W	0000_0000h	52.1.89/ 1191
4005_C2DC	PA Bias Table 0 (XCVR_PA_BIAS_TBL0)	32	R/W	0000_0000h	52.1.90/ 1191
4005_C2E0	PA Bias Table 1 (XCVR_PA_BIAS_TBL1)	32	R/W	0000_0000h	52.1.91/ 1192
4005_C2E4	Recycle Count Register (XCVR_RECYCLE_COUNT)	32	R/W	0000_0826h	52.1.92/ 1193
4005_C2E8	TSM_TIMING00 (XCVR_TSM_TIMING00)	32	R/W	6500_6A00h	52.1.93/ 1194
4005_C2EC	TSM_TIMING01 (XCVR_TSM_TIMING01)	32	R/W	6500_6A00h	52.1.94/ 1195
4005_C2F0	TSM_TIMING02 (XCVR_TSM_TIMING02)	32	R/W	6500_6A00h	52.1.95/ 1195
4005_C2F4	TSM_TIMING03 (XCVR_TSM_TIMING03)	32	R/W	6500_6A00h	52.1.96/ 1196
4005_C2F8	TSM_TIMING04 (XCVR_TSM_TIMING04)	32	R/W	6500_FFFFh	52.1.97/ 1197
4005_C2FC	TSM_TIMING05 (XCVR_TSM_TIMING05)	32	R/W	650B_6A3Fh	52.1.98/ 1197
4005_C300	TSM_TIMING06 (XCVR_TSM_TIMING06)	32	R/W	651A_FFFFh	52.1.99/ 1198
4005_C304	TSM_TIMING07 (XCVR_TSM_TIMING07)	32	R/W	1A00_4E00h	52.1.100/ 1199
4005_C308	TSM_TIMING08 (XCVR_TSM_TIMING08)	32	R/W	6533_6867h	52.1.101/ 1199
4005_C30C	TSM_TIMING09 (XCVR_TSM_TIMING09)	32	R/W	6505_6A05h	52.1.102/ 1200

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C310	TSM_TIMING10 (XCVR_TSM_TIMING10)	32	R/W	6509_FFFFh	52.1.103/ 1201
4005_C314	TSM_TIMING11 (XCVR_TSM_TIMING11)	32	R/W	FFFF_6A09h	52.1.104/ 1201
4005_C318	TSM_TIMING12 (XCVR_TSM_TIMING12)	32	R/W	FFFF_6A64h	52.1.105/ 1202
4005_C31C	TSM_TIMING13 (XCVR_TSM_TIMING13)	32	R/W	651A_6A4Eh	52.1.106/ 1202
4005_C320	TSM_TIMING14 (XCVR_TSM_TIMING14)	32	R/W	650A_FFFFh	52.1.107/ 1203
4005_C324	TSM_TIMING15 (XCVR_TSM_TIMING15)	32	R/W	FFFF_6A0Ah	52.1.108/ 1204
4005_C328	TSM_TIMING16 (XCVR_TSM_TIMING16)	32	R/W	1A10_4E44h	52.1.109/ 1204
4005_C32C	TSM_TIMING17 (XCVR_TSM_TIMING17)	32	R/W	6510_6A44h	52.1.110/ 1205
4005_C330	TSM_TIMING18 (XCVR_TSM_TIMING18)	32	R/W	6505_FFFFh	52.1.111/ 1206
4005_C334	TSM_TIMING19 (XCVR_TSM_TIMING19)	32	R/W	FFFF_6864h	52.1.112/ 1206
4005_C338	TSM_TIMING20 (XCVR_TSM_TIMING20)	32	R/W	651A_FFFFh	52.1.113/ 1207
4005_C33C	TSM_TIMING21 (XCVR_TSM_TIMING21)	32	R/W	651A_FFFFh	52.1.114/ 1207
4005_C340	TSM_TIMING22 (XCVR_TSM_TIMING22)	32	R/W	651A_FFFFh	52.1.115/ 1208
4005_C344	TSM_TIMING23 (XCVR_TSM_TIMING23)	32	R/W	651A_FFFFh	52.1.116/ 1209
4005_C348	TSM_TIMING24 (XCVR_TSM_TIMING24)	32	R/W	6518_FFFFh	52.1.117/ 1209
4005_C34C	TSM_TIMING25 (XCVR_TSM_TIMING25)	32	R/W	6518_FFFFh	52.1.118/ 1210
4005_C350	TSM_TIMING26 (XCVR_TSM_TIMING26)	32	R/W	6509_6A09h	52.1.119/ 1210
4005_C354	TSM_TIMING27 (XCVR_TSM_TIMING27)	32	R/W	FFFF_6A67h	52.1.120/ 1211
4005_C358	TSM_TIMING28 (XCVR_TSM_TIMING28)	32	R/W	6562_FFFFh	52.1.121/ 1212
4005_C35C	TSM_TIMING29 (XCVR_TSM_TIMING29)	32	R/W	6362_FFFFh	52.1.122/ 1212
4005_C360	TSM_TIMING30 (XCVR_TSM_TIMING30)	32	R/W	6510_6A44h	52.1.123/ 1213
4005_C364	TSM_TIMING31 (XCVR_TSM_TIMING31)	32	R/W	6562_FFFFh	52.1.124/ 1214

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C368	TSM_TIMING32 (XCVR_TSM_TIMING32)	32	R/W	6526_FFFFh	52.1.125/ 1214
4005_C36C	TSM_TIMING33 (XCVR_TSM_TIMING33)	32	R/W	2726_FFFFh	52.1.126/ 1215
4005_C370	TSM_TIMING34 (XCVR_TSM_TIMING34)	32	R/W	6533_6865h	52.1.127/ 1215
4005_C374	TSM_TIMING35 (XCVR_TSM_TIMING35)	32	R/W	FFFF_FFFFh	52.1.128/ 1216
4005_C378	TSM_TIMING36 (XCVR_TSM_TIMING36)	32	R/W	FFFF_FFFFh	52.1.129/ 1217
4005_C37C	TSM_TIMING37 (XCVR_TSM_TIMING37)	32	R/W	FFFF_FFFFh	52.1.130/ 1217
4005_C380	TSM_TIMING38 (XCVR_TSM_TIMING38)	32	R/W	FFFF_FFFFh	52.1.131/ 1218
4005_C384	TSM_TIMING39 (XCVR_TSM_TIMING39)	32	R/W	FFFF_FFFFh	52.1.132/ 1219
4005_C388	TSM_TIMING40 (XCVR_TSM_TIMING40)	32	R/W	FFFF_FFFFh	52.1.133/ 1219
4005_C38C	TSM_TIMING41 (XCVR_TSM_TIMING41)	32	R/W	FFFF_FFFFh	52.1.134/ 1220
4005_C390	TSM_TIMING42 (XCVR_TSM_TIMING42)	32	R/W	FFFF_FFFFh	52.1.135/ 1221
4005_C394	TSM_TIMING43 (XCVR_TSM_TIMING43)	32	R/W	FFFF_FFFFh	52.1.136/ 1221
4005_C3C0	CORR_CTRL (XCVR_CORR_CTRL)	32	R/W	See section	52.1.137/ 1222
4005_C3C4	PN_TYPE (XCVR_PN_TYPE)	32	R/W	0000_0001h	52.1.138/ 1223
4005_C3C8	PN_CODE (XCVR_PN_CODE)	32	R/W	744A_C39Bh	52.1.139/ 1224
4005_C3CC	Sync Control (XCVR_SYNC_CTRL)	32	R/W	0000_0008h	52.1.140/ 1224
4005_C3D0	SNF_THR (XCVR_SNF_THR)	32	R/W	0000_0000h	52.1.141/ 1225
4005_C3D4	FAD_THR (XCVR_FAD_THR)	32	R/W	0000_0082h	52.1.142/ 1225
4005_C3D8	ZBDEM_AFC (XCVR_ZBDEM_AFC)	32	R/W	See section	52.1.143/ 1226
4005_C3DC	LPPS Control Register (XCVR_LPPS_CTRL)	32	R/W	0000_0000h	52.1.144/ 1227
4005_C400	ADC Control (XCVR_ADC_CTRL)	32	R/W	FFFF_0001h	52.1.145/ 1228
4005_C404	ADC Tuning (XCVR_ADC_TUNE)	32	R/W	0088_0033h	52.1.146/ 1229

Table continues on the next page...

XCVR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_C408	ADC Adjustment (XCVR_ADC_ADJ)	32	R/W	4303_3033h	52.1.147/ 1230
4005_C40C	ADC Regulators (XCVR_ADC_REGS)	32	R/W	0000_0000h	52.1.148/ 1231
4005_C410	ADC Regulator Trims (XCVR_ADC_TRIMS)	32	R/W	0000_0444h	52.1.149/ 1233
4005_C414	ADC Test Control (XCVR_ADC_TEST_CTRL)	32	R/W	0000_0000h	52.1.150/ 1234
4005_C420	Baseband Filter Control (XCVR_BBF_CTRL)	32	R/W	0000_0173h	52.1.151/ 1236
4005_C42C	RX Analog Control (XCVR_RX_ANA_CTRL)	32	R/W	0000_0000h	52.1.152/ 1239
4005_C434	Crystal Oscillator Control Register 1 (XCVR_XTAL_CTRL)	32	R/W	See section	52.1.153/ 1241
4005_C438	Crystal Oscillator Control Register 2 (XCVR_XTAL_CTRL2)	32	R/W	0000_1000h	52.1.154/ 1243
4005_C43C	Bandgap Control (XCVR_BGAP_CTRL)	32	R/W	0000_0087h	52.1.155/ 1245
4005_C444	PLL Control Register (XCVR_PLL_CTRL)	32	R/W	0000_0023h	52.1.156/ 1246
4005_C448	PLL Control Register 2 (XCVR_PLL_CTRL2)	32	R/W	0000_0004h	52.1.157/ 1248
4005_C44C	PLL Test Control (XCVR_PLL_TEST_CTRL)	32	R/W	0000_0000h	52.1.158/ 1249
4005_C458	QGEN Control (XCVR_QGEN_CTRL)	32	R/W	0000_0000h	52.1.159/ 1251
4005_C464	TCA Control (XCVR_TCA_CTRL)	32	R/W	0000_0000h	52.1.160/ 1252
4005_C468	TZA Control (XCVR_TZA_CTRL)	32	R/W	0000_0044h	52.1.161/ 1254
4005_C474	TX Analog Control (XCVR_TX_ANA_CTRL)	32	R/W	0000_0000h	52.1.162/ 1255
4005_C47C	Analog Spare (XCVR_ANA_SPARE)	32	R/W	0000_0000h	52.1.163/ 1255

52.1.1 RX Digital Control (XCVR_RX_DIG_CTRL)

Address: 4005_C000h base + 0h offset = 4005_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RX_IQ_SWAP	RX_DCOC_CAL_EN	RX_DCOC_EN	RX_AGC_EN	RX_RSSI_EN	RX_NORM_EN	RX_INTERP_EN	0	RX_DEC_FILT_OSR				0	RX_ADC_RAW_EN	RX_CH_FILT_BYPASS	RX_ADC_NEGEDGE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_RX_DIG_CTRL field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 RX_IQ_SWAP	RX IQ Swap Enable swap of I/Q channels (does not affect ADC raw mode). 0 IQ swap is disabled. 1 IQ swap is enabled.
13 RX_DCOC_CAL_EN	DCOC Calibration Enable Enable DCOC warm-up calibration in receiver. 0 DCOC calibration is disabled. 1 DCOC calibration is enabled.
12 RX_DCOC_EN	DCOC Enable Enables DCO calculation and application of corrections. 0 DCOC is disabled. 1 DCOC is enabled.
11 RX_AGC_EN	AGC Global Enable Does NOT affect user gains (user gain programming has priority). 0 AGC is disabled. 1 AGC is enabled.

Table continues on the next page...

XCVR_RX_DIG_CTRL field descriptions (continued)

Field	Description
10 RX_RSSI_EN	RSSI Measurement Enable 0 RSSI measurement is disabled. 1 RSSI measurement is enabled.
9 RX_NORM_EN	Normalizer Enable 0 Normalizer is disabled. 1 Normalizer is enabled.
8 RX_INTERP_EN	Interpolator Enable 0 Interpolator is disabled. 1 Interpolator is enabled.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 RX_DEC_FILT_OSR	Decimation Filter Oversampling **NOTE:** All undocumented values are Reserved. 0 OSR 2 1 OSR 4 2 OSR 8 3 OSR 9 4 OSR 16 5 OSR 18
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RX_ADC_RAW_EN	ADC Raw Mode selection 0 Normal operation. 1 The decimation filter's 12bit output consists of two unfiltered 5-bit ADC samples. This is for test purposes only to observe ADC output via XCVR DMA or DTEST.
1 RX_CH_FILT_BYPASS	Receive Channel Filter Bypass Selects whether to disable and bypass channel filter. 0 Channel filter is enabled. 1 Disable and bypass channel filter.
0 RX_ADC_NEGEDGE	Receive ADC Negative Edge Selection Selects which edge of the clock the ADC data is registered. 0 Register ADC data on positive edge of clock 1 Register ADC data on negative edge of clock

52.1.2 AGC Control 0 (XCVR_AGC_CTRL_0)

Address: 4005_C000h base + 4h offset = 4005_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AGC_DOWN_RSSI_THRESH								AGC_UP_RSSI_THRESH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AGC_DOWN_TZA_STEP_SZ				AGC_DOWN_BBF_STEP_SZ				AGC_UP_SRC	AGC_UP_EN	FREEZE_AGC_SRC		AGC_FREEZE_EN	SLOW_AGC_SRC		SLOW_AGC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_CTRL_0 field descriptions

Field	Description
31–24 AGC_DOWN_RSSI_THRESH	AGC DOWN RSSI Threshold ADC RSSI threshold to take downward step (AGC slow).
23–16 AGC_UP_RSSI_THRESH	AGC UP RSSI Threshold ADC RSSI threshold to take upward step (AGC slow).
15–12 AGC_DOWN_TZA_STEP_SZ	AGC_DOWN_TZA_STEP_SZ Number of table steps for downward step (TZA) in AGC fast.
11–8 AGC_DOWN_BBF_STEP_SZ	AGC_DOWN_BBF_STEP_SZ Number of table steps for downward step (BBF) in AGC fast.
7 AGC_UP_SRC	AGC Up Source Criterion to use for upward AGC steps. 0 PDET LO 1 RSSI
6 AGC_UP_EN	AGC Up Enable Allow AGC to take upward steps in slow mode.
5–4 FREEZE_AGC_SRC	Freeze AGC Source Selection Select trigger source for entering freeze AGC. 0 BTLE Preamble Detect

Table continues on the next page...

XCVR_AGC_CTRL_0 field descriptions (continued)

Field	Description
	1 Zigbee Preamble Detect 2 BTLE access match (orf_access_match freeze) 3 Zigbee LQI done (1=freeze, 0=run AGC)
3 AGC_FREEZE_EN	AGC Freeze Enable Allow AGC to freeze. AGC can still go to hold mode if timer expires (same as fast expire) from slow mode.
2–1 SLOW_AGC_SRC	Slow AGC Source Selection Select trigger source for entering slow AGC. 0 BTLE Preamble Detect 1 Zigbee Preamble Detect 2 Fast AGC expire timer 3 Reserved
0 SLOW_AGC_EN	Slow AGC Enable Allow AGC to enter into slow mode.

52.1.3 AGC Control 1 (XCVR_AGC_CTRL_1)

Address: 4005_C000h base + 8h offset = 4005_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TZA_GAIN_SETTLE_TIME								0	PRESLOW_EN	USER_BBF_GAIN_EN	USER_LNM_GAIN_EN	BBF_USER_GAIN			
W									▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LNM_USER_GAIN				LNM_ALT_CODE								BBF_ALT_CODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_CTRL_1 field descriptions

Field	Description
31–24 TZA_GAIN_SETTLE_TIME	TZA_GAIN_SETTLE_TIME Number of clocks to assert TZA peak detector reset (for automatic control).

Table continues on the next page...

XCVR_AGC_CTRL_1 field descriptions (continued)

Field	Description
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 PRESLOW_EN	Pre-slow Enable Enablepre-slow state. 0 Pre-slow is disabled. 1 Pre-slow is enabled.
21 USER_BBF_GAIN_EN	User BBF Gain Enable Enable user defined BBF gain (no AGC).
20 USER_LNM_GAIN_EN	User LNM Gain Enable Enable user defined LNM gain (no AGC).
19–16 BBF_USER_GAIN	BBF_USER_GAIN User defined BBF gain index if user_bbf_gain_en =1
15–12 LNM_USER_GAIN	LNM_USER_GAIN user defined lnm gain index if user_lnm_gain_en =1
11–4 LNM_ALT_CODE	LNM_ALT_CODE Alternate LNM gain code selected when lnm_gain_xx=F. Also used as initial gain value before DCOC cal is performed.
BBF_ALT_CODE	BBF_ALT_CODE Alternate BBF gain code selected when bbf_gain_xx=0xF. Also used as initial gain value before DCOC cal is performed.

52.1.4 AGC Control 2 (XCVR_AGC_CTRL_2)

Address: 4005_C000h base + Ch offset = 4005_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		AGC_FAST_EXPIRE						TZA_PDET_THRESH_HI			TZA_PDET_THRESH_LO			BBF_PDET_THRESH_HI	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BBF_PDET_THRESH_HI	BBF_PDET_THRESH_LO			BBF_GAIN_SETTLE_TIME								0		TZA_PDET_RST	BBF_PDET_RST
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_CTRL_2 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–24 AGC_FAST_ EXPIRE	AGC Fast Expire Expire time (uS) for fast AGC (1-63uS).
23–21 TZA_PDET_ THRESH_HI	TZA PDET Threshold High TZA peak detect HI threshold. 000 0.6V 001 0.675V 010 0.75V 011 0.825V 100 0.9V 101 0.975V 110 1.05V 111 1.125V
20–18 TZA_PDET_ THRESH_LO	TZA PDET Threshold Low TZA peak detect LO threshold. 000 0.6V 001 0.675V 010 0.75V 011 0.825V 100 0.9V 101 0.975V 110 1.05V 111 1.125V
17–15 BBF_PDET_ THRESH_HI	BBF PDET Threshold High BBF peak detect HI threshold. 000 0.6V 001 0.675V 010 0.75V 011 0.825V 100 0.9V 101 0.975V 110 1.05V 111 1.125V
14–12 BBF_PDET_ THRESH_LO	BBF PDET Threshold Low BBF peak detect LO threshold. 000 0.6V 001 0.675V 010 0.75V 011 0.825V

Table continues on the next page...

XCVR_AGC_CTRL_2 field descriptions (continued)

Field	Description
	100 0.9V 101 0.975V 110 1.05V 111 1.125V
11–4 BBF_GAIN_ SETTLE_TIME	BBF Gain Settle Time Number of clocks to assert BBF peak detector reset (for automatic control).
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 TZA_PDET_RST	TZA PDET Reset TZA peak detector reset, manual control.
0 BBF_PDET_RST	BBF PDET Reset BBF peak detector reset, manual control.

52.1.5 AGC Control 3 (XCVR_AGC_CTRL_3)

Address: 4005_C000h base + 10h offset = 4005_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AGC_UP_STEP_SZ				AGC_H2S_STEP_SZ					AGC_RSSI_DELT_H2S							AGC_PDET_LO_DLY			AGC_UNFREEZE_TIME												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_CTRL_3 field descriptions

Field	Description
31–28 AGC_UP_STEP_ SZ	AGC Up Step Size Number of table steps for upward step
27–23 AGC_H2S_ STEP_SZ	AGC_H2S_STEP_SZ Step size for hold to slow jump.
22–16 AGC_RSSI_ DELT_H2S	AGC_RSSI_DELT_H2S RSSI delta that causes hold to slow transition.
15–13 AGC_PDET_LO_ DLY	AGC Peak Detect Low Delay Time (uS) to wait for pdet low to assert (1-7uS).
AGC_ UNFREEZE_ TIME	AGC Unfreeze Time Time (uS) for AGC to unfreeze (1-8191uS).

52.1.6 AGC Status (XCVR_AGC_STAT)

Address: 4005_C000h base + 14h offset = 4005_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RSSI_ADC_RAW							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						AGC_FROZEN	CURR_AGC_IDX					TZA_PDET_HI_STAT	TZA_PDET_LO_STAT	BBF_PDET_HI_STAT	BBF_PDET_LO_STAT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_STAT field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 RSSI_ADC_RAW	ADC RAW RSSI Reading Reading of ADC rssi (before adjustments)
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 AGC_FROZEN	AGC Frozen Status Status of AGC freeze. 0 AGC is not frozen. 1 AGC is frozen.
8–4 CURR_AGC_IDX	Current AGC Gain Index Current AGC gain table index
3 TZA_PDET_HI_ STAT	TZA Peak Detector High Status Status of TZA peak detector HI flag (1=set)
2 TZA_PDET_LO_ STAT	TZA Peak Detector Low Status Status of TZA peak detector LO flag (1=set)
1 BBF_PDET_HI_ STAT	BBF Peak Detector High Status Status of BBF peak detector HI flag (1=set)
0 BBF_PDET_LO_ STAT	BBF Peak Detector Low Status Status of BBF peak detector LO flag (1=set)

52.1.7 RSSI Control 0 (XCVR_RSSI_CTRL_0)

Address: 4005_C000h base + 18h offset = 4005_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSSI_ADJ								0				RSSI_IIR_WEIGHT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									RSSI_IIR_ CW_ WEIGHT		RSSI_DEC_EN	RSSI_HOLD_ EN	RSSI_ HOLD_SRC		RSSI_USE_ VALS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_RSSI_CTRL_0 field descriptions

Field	Description
31–24 RSSI_ADJ	RSSI Adjustment RSSI calculation adjustment (8-bit signed 1/4 dB).
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 RSSI_IIR_ WEIGHT	RSSI IIR Weighting IIR filter weight for RSSI filtering. 0 Bypass 1 1/2 2 1/4 3 1/8 4 1/16 5 1/32 6 Reserved 7 Reserved
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–5 RSSI_IIR_CW_ WEIGHT	RSSI IIR CW Weighting IIR filter weight for RSSI filtering of a CW input. 0 Bypass 1 1/8 2 1/16 3 1/32
4 RSSI_DEC_EN	RSSI Decimation Enable Enable RSSI 4x decimation stage.
3 RSSI_HOLD_EN	RSSI Hold Enable Enable RSSI to freeze after hold criterion met. RSSI will still be briefly held when a gain change occurs.
2–1 RSSI_HOLD_ SRC	Hold RSSI Source Selection Select trigger source for entering freezing RSSI measurement. 0 BTLE Preamble Detect 1 Zigbee Preamble Detect 2 BTLE access match (orf_access_match freeze) 3 Zigbee LQI done (1=freeze, 0=run AGC)
0 RSSI_USE_ VALS	RSSI Values Selection Enable use of TCA and BBF gain values programmed in registers for calculation.

52.1.8 RSSI Control 1 (XCVR_RSSI_CTRL_1)

Address: 4005_C000h base + 1Ch offset = 4005_C01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSSI_OUT								RSSI_ED_THRESH1_H				RSSI_ED_THRESH0_H				RSSI_ED_THRESH1								RSSI_ED_THRESH0							
W	▨																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_RSSI_CTRL_1 field descriptions

Field	Description
31–24 RSSI_OUT	RSSI Reading RSSI output (8-bit signed).
23–20 RSSI_ED_THRESH1_H	RSSI Energy Detect 1 Hysteresis ED hysteresis window size.
19–16 RSSI_ED_THRESH0_H	RSSI Energy Detect 0 Hysteresis ED hysteresis window size.
15–8 RSSI_ED_THRESH1	RSSI Energy Detect 1 Threshold Threshold for setting energy detect 1 to scanner.
RSSI_ED_THRESH0	RSSI Energy Detect 0 Threshold Threshold for setting energy detect 0 to scanner.

52.1.9 DCOC Control 0 (XCVR_DCOC_CTRL_0)

Address: 4005_C000h base + 20h offset = 4005_C020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCOC_CORR_HOLD_TIME								DCOC_CORR_DLY				DCOC_CAL_DURATION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_CAL_DURATION	DCOC_ALPHA_RADIUS_IDX			0		DCOC_ALPHAC_SCALE_IDX		0	DCOC_SIGN_SCALE_IDX		DCOC_CORRECT_EN	DCOC_TRACK_EN	0	DCOC_MAN	0
W					▨				▨					▨		▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CTRL_0 field descriptions

Field	Description
31–25 DCOC_CORR_ HOLD_TIME	DCOC Correction Hold Time Delay (in uS) from last gain change to freezing DC correction (1-127uS, 127=never freeze). A value of 0 should never be used.
24–20 DCOC_CORR_ DLY	DCOC Correction Delay Wait time (in uS) between corrections (1-31uS). A value of 0 should never be used.
19–15 DCOC_CAL_ DURATION	DCOC Calibration Duration Duration (in uS) of a calibration (1-31uS). A value of 0 should never be used.
14–12 DCOC_ALPHA_ RADIUS_IDX	Alpha-R Scaling DCOC Alpha-R Scaling. Radius stepsize used in the DCOC tracking estimator. Used when DCOC_TRACK_EN=1. 000 1 001 1/2 010 1/4 011 1/8 100 1/16 101 1/32 110 1/64 111 Reserved
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 DCOC_ ALPHAC_ SCALE_IDX	DCOC Alpha-C Scaling DCOC Alpha-C Scaling. I/Q center stepsize used in the DCOC tracking estimator. Used when DCOC_TRACK_EN=1. 00 1/2 01 1/4 10 1/8 11 1/16
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–5 DCOC_SIGN_ SCALE_IDX	DCOC Sign Scaling DCOC Sign Scaling. Sign()-based scaling factor used in the DCOC tracking estimator. Used when DCOC_TRACK_EN=1. 00 1/4 01 1/8 10 1/16 11 1/32
4 DCOC_ CORRECT_EN	DCOC Correction Enable Enables the DCOC to use the TZA and BBA DACs to correct the DC offset.

Table continues on the next page...

XCVR_DCOC_CTRL_0 field descriptions (continued)

Field	Description
3 DCOC_TRACK_EN	DCOC Tracking Enable Enables the DCOC tracking estimator to correct the DC offset. Can be used with or without DCOC calibration (RX_DCOC_CAL_EN).
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DCOC_MAN	DCOC Manual Override If the manual override bit is set, it forces the DCOC to use the DAC and digital correction values from registers XCVR_DCOC_CTRL_3 and XCVR_DCOC_CTRL_4, respectively.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.1.10 DCOC Control 1 (XCVR_DCOC_CTRL_1)

Address: 4005_C000h base + 24h offset = 4005_C024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					TZA_CORR_POL	BBA_CORR_POL	TRACK_FROM_ZERO	0						0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BBF_DCOC_STEP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CTRL_1 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 TZA_CORR_POL	TZA Correction Polarity Selects polarity of TZA corrections. 0 Normal polarity. 1 Negative polarity. This should be set if the ADC output is inverted, or if the TZA DACs were implemented with negative polarity.

Table continues on the next page...

XCVR_DCOC_CTRL_1 field descriptions (continued)

Field	Description
25 BBA_CORR_ POL	BBA Correction Polarity Selects polarity of BBA corrections. 0 Normal polarity. 1 Negative polarity. This should be set if the ADC output is inverted, or if the BBA DACs were implemented with negative polarity.
24 TRACK_FROM_ ZERO	Track from Zero Selects whether the tracking estimator resets its DC estimate on every AGC gain change to zero or uses the current I/Q sample. 0 Track from current I/Q sample. 1 Track from zero.
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BBF_DCOC_ STEP	DCOC BBF Step Size DCOC BBF Step Size (format: 6.3). The nominal value for this is the DAC resolution (1.2/2^6= 18.7mV) times AGC gain of -1.7dB (10^(-1.7/20)=0.822) times an AGC mV to quantization scaling factor (2^11/1000 = 2.048), which is 31.57. This value is stored in the register with 3 fractional bits, so use round(31.57*2^3) = 253 decimal.

52.1.11 DCOC Control 2 (XCVR_DCOC_CTRL_2)

Address: 4005_C000h base + 28h offset = 4005_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																			BBF_DCOC_STEP_RECIP												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CTRL_2 field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BBF_DCOC_ STEP_RECIP	DCOC BBF Reciprocal of Step Size DCOC BBF Reciprocal of Step Size (format: .[00]13). This the reciprocal of the BBF DCOC STEP value programmed in the XCVR_DCOC_CTRL_1 register. It's nominal value is 1.0/31.57. This value is stored as a 15 bit fraction (though only 13 bits are programmed), so use round(1.0/31.57*2^15) = 1038decimal.

52.1.12 DCOC Control 3 (XCVR_DCOC_CTRL_3)

Address: 4005_C000h base + 2Ch offset = 4005_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TZA_DCOC_INIT_Q								TZA_DCOC_INIT_I								0		BBF_DCOC_INIT_Q						0		BBF_DCOC_INIT_I					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CTRL_3 field descriptions

Field	Description
31–24 TZA_DCOC_ INIT_Q	DCOC TZA Init Q Manual override value for DCOC TZA Q channel DAC. Used when DCOC_MAN=1.
23–16 TZA_DCOC_ INIT_I	DCOC TZA Init I Manual override value for DCOC TZA I channel DAC. Used when DCOC_MAN=1.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 BBF_DCOC_ INIT_Q	DCOC BBF Init Q Manual override value for DCOC BBF Q channel DAC. Used when DCOC_MAN=1.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BBF_DCOC_ INIT_I	DCOC BBF Init I Manual override value for DCOC BBF I channel DAC. Used when DCOC_MAN=1.

52.1.13 DCOC Control 4 (XCVR_DCOC_CTRL_4)

Address: 4005_C000h base + 30h offset = 4005_C030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DIG_DCOC_INIT_Q												0				DIG_DCOC_INIT_I											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CTRL_4 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DIG_DCOC_ INIT_Q	DCOC DIG Init Q Manual override for DCOC DIG Q channel correction. Value to be subtracted from downsampled Q channel. Used when DCOC_MAN=1.

Table continues on the next page...

XCVR_DCOC_CTRL_4 field descriptions (continued)

Field	Description
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DIG_DCOC_ INIT_I	DCOC DIG Init I Manual override for DCOC DIG I channel correction. Value to be subtracted from downsampled I channel. Used when DCOC_MAN=1.

52.1.14 DCOC Calibration Gain (XCVR_DCOC_CAL_GAIN)

Address: 4005_C000h base + 34h offset = 4005_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_TZA_ CAL_GAIN3				DCOC_BBF_ CAL_GAIN3				DCOC_TZA_ CAL_GAIN2				DCOC_BBF_ CAL_GAIN2				DCOC_TZA_ CAL_GAIN1				DCOC_BBF_ CAL_GAIN1				0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_GAIN field descriptions

Field	Description
31–28 DCOC_TZA_ CAL_GAIN3	DCOC TZA Calibration Gain 3 The LNM gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
27–24 DCOC_BBF_ CAL_GAIN3	DCOC BBF Calibration Gain 3 The BBF gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
23–20 DCOC_TZA_ CAL_GAIN2	DCOC TZA Calibration Gain 2 The LNM gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
19–16 DCOC_BBF_ CAL_GAIN2	DCOC BBF Calibration Gain 2 The BBF gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
15–12 DCOC_TZA_ CAL_GAIN1	DCOC TZA Calibration Gain 1 The LNM gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
11–8 DCOC_BBF_ CAL_GAIN1	DCOC BBF Calibration Gain 1 The BBF gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.1.15 DCOC Status (XCVR_DCOC_STAT)

Address: 4005_C000h base + 38h offset = 4005_C038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TZA_DCOC_Q								TZA_DCOC_I								0		BBF_DCOC_Q						0		BBF_DCOC_I					
W																																
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

XCVR_DCOC_STAT field descriptions

Field	Description
31–24 TZA_DCOC_Q	DCOC TZA DAC Q Current TZA DAC setting for Q channel. Note that the TZA DACs have a bias of 0x80; 0x0 represents the most negative DC offset, 0x80 represents a DC offset of 0, and 0xFF represents the most postiive DC offset. This is provided for debug and characterization purposes only.
23–16 TZA_DCOC_I	DCOC TZA DAC I Current TZA DAC setting for I channel. Note that the TZA DACs have a bias of 0x80; 0x0 represents the most negative DC offset, 0x80 represents a DC offset of 0, and 0xFF represents the most postiive DC offset. This is provided for debug and characterization purposes only.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 BBF_DCOC_Q	DCOC BBF DAC Q Current BBF DAC setting for Q channel. Note that the BBF DACs have a bias of 0x20; 0x0 represents the most negative DC offset, 0x20 represents a DC offset of 0, and 0x3F represents the most postiive DC offset. This is provided for debug and characterization purposes only.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BBF_DCOC_I	DCOC BBF DAC I Current BBF DAC setting for I channel. Note that the BBF DACs have a bias of 0x20; 0x0 represents the most negative DC offset, 0x20 represents a DC offset of 0, and 0x3F represents the most postiive DC offset. This is provided for debug and characterization purposes only.

52.1.16 DCOC DC Estimate (XCVR_DCOC_DC_EST)

Address: 4005_C000h base + 3Ch offset = 4005_C03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DC_EST_Q												0				DC_EST_I											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_DC_EST field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DC_EST_Q	DCOC DC Estimate Q Reflects the current DCOC DC tracking estimate for Q channel. Used when DCOC_TRACK_EN=1. This is provided for debug and characterization purposes only.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DC_EST_I	DCOC DC Estimate I Reflects the current DCOC DC tracking estimate for I channel. Used when DCOC_TRACK_EN=1. This is provided for debug and characterization purposes only.

52.1.17 DCOC Calibration Reciprocals (XCVR_DCOC_CAL_RCP)

Address: 4005_C000h base + 40h offset = 4005_C040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											ALPHA_CALC_RECIP											DCOC_TMP_CALC_RECIP									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_RCP field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–10 ALPHA_CALC_ RECIP	Alpha Calculation Reciprocal DCOC Alpha calculation reciprocal (format: .11). This is used in DCOC calibration calculation of the alpha DC component. It is defined as: $1.0/((G_{L\_HI} - G_{L\_LO})*G_{B\_LO})$ This is stored as with 11 fractional bits, so program the value $round([1.0/((G_{L\_HI} - G_{L\_LO})*G_{B\_LO})]*2^{\wedge}11)$.
DCOC_TMP_ CALC_RECIP	DCOC Calculation Reciprocal DCOC_tmp calculation reciprocal (format: .10). This is used in DCDC calibration calculation. It is defined as $1.0/(G_{B\_HI} - G_{B\_LO})$ This is stored with 10 fractional bits, so program the value $round([1.0/(G_{B\_HI} - G_{B\_LO})]*2^{\wedge}10)$.

52.1.18 IQMC Control (XCVR_IQMC_CTRL)

Address: 4005_C000h base + 4Ch offset = 4005_C04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IQMC_NUM_ITER								0							IQMC_CAL_EN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_IQMC_CTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 IQMC_NUM_ITER	IQ Mismatch Cal Num Iter Number of iterations for IQ Mismatch Calibration.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 IQMC_CAL_EN	IQ Mismatch Cal Enable Enables IQ mismatch calibration.

52.1.19 IQMC Calibration (XCVR_IQMC_CAL)

Address: 4005_C000h base + 50h offset = 4005_C050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				IQMC_PHASE_ADJ												0					IQMC_GAIN_ADJ										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

XCVR_IQMC_CAL field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_IQMC_CAL field descriptions (continued)

Field	Description
27–16 IQMC_PHASE_ADJ	IQ Mismatch Correction Phase Coeff I/Q mismatch correction phase coefficient.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
IQMC_GAIN_ADJ	IQ Mismatch Correction Gain Coeff I/Q mismatch correction gain coefficient.

52.1.20 TCA AGC Step Values 3..0 (XCVR_TCA_AGC_VAL_3_0)

Address: 4005_C000h base + 54h offset = 4005_C054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCA_AGC_VAL_3								TCA_AGC_VAL_2								TCA_AGC_VAL_1								TCA_AGC_VAL_0							
Reset	0	0	1	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	1	0	0

XCVR_TCA_AGC_VAL_3_0 field descriptions

Field	Description
31–24 TCA_AGC_VAL_3	TCA_AGC step 3 4x value of TCA_AGC step 3, for RSSI calculation.
23–16 TCA_AGC_VAL_2	TCA_AGC step 2 4x value of TCA_AGC step 2, for RSSI calculation.
15–8 TCA_AGC_VAL_1	TCA_AGC step 1 4(x+8) value of TCA_AGC step 1, for RSSI calculation.
TCA_AGC_VAL_0	TCA_AGC step 0 4(x+8) value of TCA_AGC step 0, for RSSI calculation.

52.1.21 TCA AGC Step Values 7..4 (XCVR_TCA_AGC_VAL_7_4)

Address: 4005_C000h base + 58h offset = 4005_C058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCA_AGC_VAL_7								TCA_AGC_VAL_6								TCA_AGC_VAL_5								TCA_AGC_VAL_4							
Reset	1	0	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	0	1	0	1	0	0

XCVR_TCA_AGC_VAL_7_4 field descriptions

Field	Description
31–24 TCA_AGC_VAL_ 7	TCA_AGC step 7 4x value of TCA_AGC step 7, for RSSI calculation.
23–16 TCA_AGC_VAL_ 6	TCA_AGC step 6 4x value of TCA_AGC step 6, for RSSI calculation.
15–8 TCA_AGC_VAL_ 5	TCA_AGC step 5 4x value of TCA_AGC step 5, for RSSI calculation.
TCA_AGC_VAL_ 4	TCA_AGC step 4 4x value of TCA_AGC step 4, for RSSI calculation.

52.1.22 TCA AGC Step Values 8 (XCVR_TCA_AGC_VAL_8)

Address: 4005_C000h base + 5Ch offset = 4005_C05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								TCA_AGC_VAL_8							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0

XCVR_TCA_AGC_VAL_8 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TCA_AGC_VAL_ 8	TCA_AGC step 8 4x value of TCA_AGC step 8, for RSSI calculation.

52.1.23 BBF Resistor Tune Values 7..0 (XCVR_BBF_RES_TUNE_VAL_7_0)

Address: 4005_C000h base + 60h offset = 4005_C060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BBF_RES_ TUNE_VAL_7				BBF_RES_ TUNE_VAL_6				BBF_RES_ TUNE_VAL_5				BBF_RES_ TUNE_VAL_4				BBF_RES_ TUNE_VAL_3				BBF_RES_ TUNE_VAL_2				BBF_RES_ TUNE_VAL_1				BBF_RES_ TUNE_VAL_0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_BBF_RES_TUNE_VAL_7_0 field descriptions

Field	Description
31–28 BBF_RES_ TUNE_VAL_7	BBF Resistor Tune Step 7 Signed, 2x offset value of bbf_res_tune step 7 agc delta.
27–24 BBF_RES_ TUNE_VAL_6	BBF Resistor Tune Step 6 Signed, 2x offset value of bbf_res_tune step 6 agc delta.
23–20 BBF_RES_ TUNE_VAL_5	BBF Resistor Tune Step 5 Signed, 2x offset value of bbf_res_tune step 5 agc delta.
19–16 BBF_RES_ TUNE_VAL_4	BBF Resistor Tune Step 4 Signed, 2x offset value of bbf_res_tune step 4 agc delta.
15–12 BBF_RES_ TUNE_VAL_3	BBF Resistor Tune Step 3 Signed, 2x offset value of bbf_res_tune step 3 agc delta.
11–8 BBF_RES_ TUNE_VAL_2	BBF Resistor Tune Step 2 Signed, 2x offset value of bbf_res_tune step 2 agc delta.
7–4 BBF_RES_ TUNE_VAL_1	BBF Resistor Tune Step 1 Signed, 2x offset value of bbf_res_tune step 1 agc delta.
BBF_RES_ TUNE_VAL_0	BBF Resistor Tune Step 0 Signed, 2x offset value of bbf_res_tune step 0 agc delta.

52.1.24 BBF Resistor Tune Values 10..8 (XCVR_BBF_RES_TUNE_VAL_10_8)

Address: 4005_C000h base + 64h offset = 4005_C064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				BBF_RES_ TUNE_VAL_ 10				BBF_RES_ TUNE_VAL_9				BBF_RES_ TUNE_VAL_8			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_BBF_RES_TUNE_VAL_10_8 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 BBF_RES_ TUNE_VAL_10	BBF Resistor Tune Step 10 Signed, 2x offset value of bbf_res_tune step 10 agc delta.

Table continues on the next page...

XCVR_BBF_RES_TUNE_VAL_10_8 field descriptions (continued)

Field	Description
7–4 BBF_RES_ TUNE_VAL_9	BBF Resistor Tune Step 9 Signed, 2x offset value of bbf_res_tune step 9 agc delta.
BBF_RES_ TUNE_VAL_8	BBF Resistor Tune Step 8 Signed, 2x offset value of bbf_res_tune step 8 agc delta.

52.1.25 TCA AGC Linear Gain Values 2..0 (XCVR_TCA_AGC_LIN_VAL_2_0)

Address: 4005_C000h base + 68h offset = 4005_C068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TCA_AGC_LIN_VAL_2										TCA_AGC_LIN_VAL_1										TCA_AGC_LIN_VAL_0									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TCA_AGC_LIN_VAL_2_0 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–20 TCA_AGC_LIN_ VAL_2	TCA AGC Linear Gain Step 2 LNM linear gain value for index 2, e.g. nominal value is 10^(9/20). Stored with 2 fractional bits, e.g. round([10^(9/20)]*2^2) = 11decimal
19–10 TCA_AGC_LIN_ VAL_1	TCA AGC Linear Gain Step 1 LNM linear gain value for index 1, e.g. nominal value is 10^(3/20). Stored with 2 fractional bits, e.g. round([10^(3/20)]*2^2) = 6decimal
TCA_AGC_LIN_ VAL_0	LNM linear gain value for index 0, e.g. nominal value is 10^(-3/20). Stored with 2 fractional bits, e.g. round([10^(-3/20)]*2^2) = 3decimal Linear gain (8.2).

52.1.26 TCA AGC Linear Gain Values 5..3 (XCVR_TCA_AGC_LIN_VAL_5_3)

Address: 4005_C000h base + 6Ch offset = 4005_C06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TCA_AGC_LIN_VAL_5										TCA_AGC_LIN_VAL_4										TCA_AGC_LIN_VAL_3									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TCA_AGC_LIN_VAL_5_3 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–20 TCA_AGC_LIN_VAL_5	TCA AGC Linear Gain Step 5 LNM linear gain value for index 5, e.g. nominal value is 10^(27/20). Stored with 2 fractional bits, e.g. round([10^(27/20)]*2^2) = 90decimal
19–10 TCA_AGC_LIN_VAL_4	TCA AGC Linear Gain Step 4 LNM linear gain value for index 4, e.g. nominal value is 10^(21/20). Stored with 2 fractional bits, e.g. round([10^(21/20)]*2^2) = 45decimal
TCA_AGC_LIN_VAL_3	TCA AGC Linear Gain Step 3 LNM linear gain value for index 3, e.g. nominal value is 10^(15/20). Stored with 2 fractional bits, e.g. round([10^(15/20)]*2^2) = 22decimal

52.1.27 TCA AGC Linear Gain Values 8..6 (XCVR_TCA_AGC_LIN_VAL_8_6)

Address: 4005_C000h base + 70h offset = 4005_C070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TCA_AGC_LIN_VAL_8										TCA_AGC_LIN_VAL_7										TCA_AGC_LIN_VAL_6									
W	▨	▨																														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TCA_AGC_LIN_VAL_8_6 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–20 TCA_AGC_LIN_VAL_8	TCA AGC Linear Gain Step 8 LNM linear gain value for index 8, e.g. nominal value is 10^(45/20). Stored with 2 fractional bits, e.g. round([10^(45/20)]*2^2) = 711decimal
19–10 TCA_AGC_LIN_VAL_7	TCA AGC Linear Gain Step 7 LNM linear gain value for index 7, e.g. nominal value is 10^(39/20). Stored with 2 fractional bits, e.g. round([10^(39/20)]*2^2) = 357decimal
TCA_AGC_LIN_VAL_6	TCA AGC Linear Gain Step 6 LNM linear gain value for index 6, e.g. nominal value is 10^(33/20). Stored with 2 fractional bits, e.g. round([10^(33/20)]*2^2) = 179decimal

52.1.28 BBF Resistor Tune Values 3..0 (XCVR_BBF_RES_TUNE_LIN_VAL_3_0)

Address: 4005_C000h base + 74h offset = 4005_C074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BBF_RES_TUNE_LIN_VAL_3								BBF_RES_TUNE_LIN_VAL_2								BBF_RES_TUNE_LIN_VAL_1								BBF_RES_TUNE_LIN_VAL_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_BBF_RES_TUNE_LIN_VAL_3_0 field descriptions

Field	Description
31–24 BBF_RES_TUNE_LIN_VAL_3	BBF Resistor Tune Linear Gain Step 3 BBF linear gain value for index 3 (format: 5.3). Nominal value is 10^(9/20). Stored with 3 fractional bits, e.g. round([10^(9/20)]*2^3) = 23decimal
23–16 BBF_RES_TUNE_LIN_VAL_2	BBF Resistor Tune Linear Gain Step 2 BBF linear gain value for index 2 (format: 5.3). Nominal value is 10^(6/20). Stored with 3 fractional bits, e.g. round([10^(6/20)]*2^3) = 16decimal
15–8 BBF_RES_TUNE_LIN_VAL_1	BBF Resistor Tune Linear Gain Step 1 BBF linear gain value for index 1 (format: 5.3). Nominal value is 10^(3/20). Stored with 3 fractional bits, e.g. round([10^(3/20)]*2^3) = 11decimal
BBF_RES_TUNE_LIN_VAL_0	BBF Resistor Tune Linear Gain Step 0 BBF linear gain value for index 0 (format: 5.3). Nominal value is 10^(0/20). Stored with 3 fractional bits, e.g. round([10^(0/20)]*2^3) = 8decimal

52.1.29 BBF Resistor Tune Values 7..4 (XCVR_BBF_RES_TUNE_LIN_VAL_7_4)

Address: 4005_C000h base + 78h offset = 4005_C078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BBF_RES_TUNE_LIN_VAL_7								BBF_RES_TUNE_LIN_VAL_6								BBF_RES_TUNE_LIN_VAL_5								BBF_RES_TUNE_LIN_VAL_4							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_BBF_RES_TUNE_LIN_VAL_7_4 field descriptions

Field	Description
31–24 BBF_RES_TUNE_LIN_VAL_7	BBF Resistor Tune Linear Gain Step 7 BBF linear gain value for index 7 (format: 5.3). Nominal value is 10^(21/20). Stored with 3 fractional bits, e.g. round([10^(21/20)]*2^3) = 90decimal

Table continues on the next page...

XCVR_BBF_RES_TUNE_LIN_VAL_7_4 field descriptions (continued)

Field	Description
23–16 BBF_RES_ TUNE_LIN_VAL_ 6	BBF Resistor Tune Linear Gain Step 6 BBF linear gain value for index 6 (format: 5.3). Nominal value is 10^(18/20). Stored with 3 fractional bits, e.g. round([10^(18/20)]*2^3) = 64decimal
15–8 BBF_RES_ TUNE_LIN_VAL_ 5	BBF Resistor Tune Linear Gain Step 5 BBF linear gain value for index 5 (format: 5.3). Nominal value is 10^(15/20). Stored with 3 fractional bits, e.g. round([10^(15/20)]*2^3) = 45decimal
BBF_RES_ TUNE_LIN_VAL_ 4	BBF Resistor Tune Linear Gain Step 4 BBF linear gain value for index 4 (format: 5.3). Nominal value is 10^(12/20). Stored with 3 fractional bits, e.g. round([10^(12/20)]*2^3) = 32decimal

52.1.30 BBF Resistor Tune Values 10..8 (XCVR_BBF_RES_TUNE_LIN_VAL_10_8)

Address: 4005_C000h base + 7Ch offset = 4005_C07Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BBF_RES_TUNE_LIN_VAL_10								BBF_RES_TUNE_LIN_VAL_9								BBF_RES_TUNE_LIN_VAL_8							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_BBF_RES_TUNE_LIN_VAL_10_8 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 BBF_RES_ TUNE_LIN_VAL_ 10	BBF Resistor Tune Linear Gain Step 10 BBF linear gain value for index 10 (format: 5.3). Nominal value is 10^(30/20). Stored with 3 fractional bits, e.g. round([10^(30/20)]*2^3) = 253decimal
15–8 BBF_RES_ TUNE_LIN_VAL_ 9	BBF Resistor Tune Linear Gain Step 9 BBF linear gain value for index 9 (format: 5.3). Nominal value is 10^(27/20). Stored with 3 fractional bits, e.g. round([10^(27/20)]*2^3) = 179decimal
BBF_RES_ TUNE_LIN_VAL_ 8	BBF Resistor Tune Linear Gain Step 8 BBF linear gain value for index 8 (format: 5.3). Nominal value is 10^(24/20). Stored with 3 fractional bits, e.g. round([10^(24/20)]*2^3) = 127decimal

52.1.31 AGC Gain Tables Step 03..00 (XCVR_AGC_GAIN_TBL_03_00)

Address: 4005_C000h base + 80h offset = 4005_C080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNM_GAIN_03				BBF_GAIN_03				LNM_GAIN_02				BBF_GAIN_02				LNM_GAIN_01				BBF_GAIN_01				LNM_GAIN_00				BBF_GAIN_00			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_03_00 field descriptions

Field	Description
31–28 LNM_GAIN_03	LNM Gain 03 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_03	BBF Gain 03 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_02	LNM Gain 02 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_02	BBF Gain 02 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_01	LNM Gain 01 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_01	BBF Gain 01 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_00	LNM Gain 00 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_00	BBF Gain 00 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.32 AGC Gain Tables Step 07..04 (XCVR_AGC_GAIN_TBL_07_04)

Address: 4005_C000h base + 84h offset = 4005_C084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNM_GAIN_07				BBF_GAIN_07				LNM_GAIN_06				BBF_GAIN_06				LNM_GAIN_05				BBF_GAIN_05				LNM_GAIN_04				BBF_GAIN_04			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_07_04 field descriptions

Field	Description
31–28 LNM_GAIN_07	LNM Gain 07 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_07	BBF Gain 07 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_06	LNM Gain 06 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_06	BBF Gain 06 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_05	LNM Gain 05 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_05	BBF Gain 05 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_04	LNM Gain 04 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_04	BBF Gain 04 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.33 AGC Gain Tables Step 11..08 (XCVR_AGC_GAIN_TBL_11_08)

Address: 4005_C000h base + 88h offset = 4005_C088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LNM_GAIN_ 11				BBF_GAIN_ 11				LNM_GAIN_ 10				BBF_GAIN_ 10				LNM_GAIN_ 09				BBF_GAIN_ 09				LNM_GAIN_ 08				BBF_GAIN_ 08			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_11_08 field descriptions

Field	Description
31–28 LNM_GAIN_11	LNM Gain 11 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_11	BBF Gain 11 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_10	LNM Gain 10 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.

Table continues on the next page...

XCVR_AGC_GAIN_TBL_11_08 field descriptions (continued)

Field	Description
19–16 BBF_GAIN_10	BBF Gain 10 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_09	LNM Gain 09 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_09	BBF Gain 09 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_08	LNM Gain 08 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_08	BBF Gain 08 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.34 AGC Gain Tables Step 15..12 (XCVR_AGC_GAIN_TBL_15_12)

Address: 4005_C000h base + 8Ch offset = 4005_C08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNM_GAIN_ 15				BBF_GAIN_ 15				LNM_GAIN_ 14				BBF_GAIN_ 14				LNM_GAIN_ 13				BBF_GAIN_ 13				LNM_GAIN_ 12				BBF_GAIN_ 12			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_15_12 field descriptions

Field	Description
31–28 LNM_GAIN_15	LNM Gain 15 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_15	BBF Gain 15 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_14	LNM Gain 14 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_14	BBF Gain 14 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_13	LNM Gain 13 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_13	BBF Gain 13 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

Table continues on the next page...

XCVR_AGC_GAIN_TBL_15_12 field descriptions (continued)

Field	Description
7–4 LNM_GAIN_12	LNM Gain 12 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_12	BBF Gain 12 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.35 AGC Gain Tables Step 19..16 (XCVR_AGC_GAIN_TBL_19_16)

Address: 4005_C000h base + 90h offset = 4005_C090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNM_GAIN_19				BBF_GAIN_19				LNM_GAIN_18				BBF_GAIN_18				LNM_GAIN_17				BBF_GAIN_17				LNM_GAIN_16				BBF_GAIN_16			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_19_16 field descriptions

Field	Description
31–28 LNM_GAIN_19	LNM Gain 19 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_19	BBF Gain 193 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_18	LNM Gain 18 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_18	BBF Gain 18 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_17	LNM Gain 17 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_17	BBF Gain 17 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_16	LNM Gain 16 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_16	BBF Gain 16 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.36 AGC Gain Tables Step 23..20 (XCVR_AGC_GAIN_TBL_23_20)

Address: 4005_C000h base + 94h offset = 4005_C094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LNM_GAIN_23				BBF_GAIN_23				LNM_GAIN_22				BBF_GAIN_22				LNM_GAIN_21				BBF_GAIN_21				LNM_GAIN_20				BBF_GAIN_20			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_23_20 field descriptions

Field	Description
31–28 LNM_GAIN_23	LNM Gain 23 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
27–24 BBF_GAIN_23	BBF Gain 23 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23–20 LNM_GAIN_22	LNM Gain 22 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_22	BBF Gain 22 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_21	LNM Gain 21 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_21	BBF Gain 21 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_20	LNM Gain 20 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_20	BBF Gain 20 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.37 AGC Gain Tables Step 26..24 (XCVR_AGC_GAIN_TBL_26_24)

Address: 4005_C000h base + 98h offset = 4005_C098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								LNM_GAIN_26				BBF_GAIN_26				LNM_GAIN_25				BBF_GAIN_25				LNM_GAIN_24				BBF_GAIN_24			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_AGC_GAIN_TBL_26_24 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–20 LNM_GAIN_26	LNM Gain 26 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
19–16 BBF_GAIN_26	BBF Gain 26 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15–12 LNM_GAIN_25	LNM Gain 25 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
11–8 BBF_GAIN_25	BBF Gain 25 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7–4 LNM_GAIN_24	LNM Gain 24 LNM GAIN 0=-3 dB, 1=3 dB, 2=9 dB … 8=45 dB.
BBF_GAIN_24	BBF Gain 24 BBF GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

52.1.38 DCOC Offset (XCVR_DCOC_OFFSET_*n*)

Address: 4005_C000h base + A0h offset + (4d × i), where i=0d to 26d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_TZA_OFFSET_Q								DCOC_TZA_OFFSET_I								0		DCOC_BBF_OFFSET_Q						0		DCOC_BBF_OFFSET_I					
W																	▨								▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_OFFSET_*n* field descriptions

Field	Description
31–24 DCOC_TZA_ OFFSET_Q	DCOC TZA Q-channel offset DCOC TZA Q-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.
23–16 DCOC_TZA_ OFFSET_I	DCOC TZA I-channel offset DCOC TZA I-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 DCOC_BBF_ OFFSET_Q	DCOC BBF Q-channel offset DCOC BBF Q-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.

Table continues on the next page...

XCVR_DCOC_OFFSET_*n* field descriptions (continued)

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DCOC_BBF_ OFFSET_I	DCOC BBF I-channel offset DCOC BBF I-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.

52.1.39 DCOC TZA DC step (XCVR_DCOC_TZA_STEP_*n*)

Address: 4005_C000h base + 110h offset + (4d × i), where i=0d to 10d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DCOC_TZA_STEP_GAIN												0			DCOC_TZA_STEP_RCP												
W	▨																▨															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_TZA_STEP_*n* field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DCOC_TZA_ STEP_GAIN	DCOC_TZA_STEP_GAIN DCOC TZA Step Size with gain (format 9.3). The nominal value for this is the TZA DAC resolution (1.4/2^8= 5.47mV) times AGC gain of -1.7dB (10^(-1.7/20)=0.822) times an AGC mV to quantization scaling factor (2^11/1000 = 2.048), times the nth BBF gain. E.g., for XCVR_DCOC_TZA_STEP_0 the BBF gain index 0 is 0dB=1.0, so DCOC_TZA_STEP_GAIN = 9.209. This value is stored in the register with 3 fractional bits, so use round(9.209*2^3) = 74 decimal.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DCOC_TZA_ STEP_RCP	DCOC_TZA_STEP_RCP DCOC TZA Reciprocal of Step Size, (format: [00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN. E.g., for DCOC_TZA_STEP_0, it's nominal value is 1.0/9.209 or 0.10859. The value is stored as a 15bit fractional value (though only 13 bits are programmed), so use round(0.10859*2^15) = 3588decimal.

52.1.40 DCOC Calibration Alpha (XCVR_DCOC_CAL_ALPHA)

Address: 4005_C000h base + 16Ch offset = 4005_C16Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_CAL_ALPHA_Q																DCOC_CAL_ALPHA_I															
W	▨																▨															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_ALPHA field descriptions

Field	Description
31–16 DCOC_CAL_ ALPHA_Q	DCOC_CAL_ALPHA_Q DCOC Calibration Q-channel ALPHA. This read-only, signed 16bit value represents the Q channel estimate of the ALPHA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.
DCOC_CAL_ ALPHA_I	DCOC Calibration I-channel ALPHA constant DCOC Calibration I-channel ALPHA. This read-only, signed 16bit value represents the I channel estimate of the ALPHA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.

52.1.41 DCOC Calibration Beta (XCVR_DCOC_CAL_BETA)

Address: 4005_C000h base + 170h offset = 4005_C170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_CAL_BETA_Q																DCOC_CAL_BETA_I															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_BETA field descriptions

Field	Description
31–16 DCOC_CAL_ BETA_Q	DCOC_CAL_BETA_Q DCOC Calibration Q-channel BETA. This read-only, signed 16bit value represents the Q channel estimate of the BETA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.
DCOC_CAL_ BETA_I	DCOC_CAL_BETA_I DCOC Calibration I-channel BETA. This read-only, signed 16bit value represents the I channel estimate of the BETA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.

52.1.42 DCOC Calibration Gamma (XCVR_DCOC_CAL_GAMMA)

Address: 4005_C000h base + 174h offset = 4005_C174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_CAL_GAMMA_Q																DCOC_CAL_GAMMA_I															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_GAMMA field descriptions

Field	Description
31–16 DCOC_CAL_ GAMMA_Q	DCOC_CAL_GAMMA_Q DCOC Calibration Q-channel GAMMA. This read-only, signed 16bit value represents the Q channel estimate of the GAMMA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.
DCOC_CAL_ GAMMA_I	DCOC_CAL_GAMMA_I DCOC Calibration I-channel GAMMA. This read-only, signed 16bit value represents the I channel estimate of the GAMMA DC component calculated in DCOC calibration. This is provided for debug/characterization purposes.

52.1.43 DCOC Calibration IIR (XCVR_DCOC_CAL_IIR)

Address: 4005_C000h base + 178h offset = 4005_C178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DCOC_CAL_ IIR3A_IDX		DCOC_CAL_ IIR2A_IDX		DCOC_CAL_ IIR1A_IDX	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL_IIR field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 DCOC_CAL_ IIR3A_IDX	DCOC Calibration IIR 3A Index DCOC Calibration IIR 3A Index. Defines the filter coefficient use for the 3rd IIR filter in the DCOC calibration DC estimator. 0 1/4 1 1/8 2 1/16 3 1/32
3–2 DCOC_CAL_ IIR2A_IDX	DCOC Calibration IIR 2A Index DCOC Calibration IIR 2A Index. Defines the filter coefficient use for the 2nd IIR filter in the DCOC calibration DC estimator. 0 1/1 1 1/4 2 1/8 3 1/16

Table continues on the next page...

XCVR_DCOC_CAL_IIR field descriptions (continued)

Field	Description
DCOC_CAL_ IIR1A_IDX	DCOC Calibration IIR 1A Index DCOC Calibration IIR 1A Index. Defines the filter coefficient use for the 1st IIR filter in the DCOC calibration DC estimator. 0 1/1 1 1/4 2 1/8 3 1/16

52.1.44 DCOC Calibration Result (XCVR_DCOC_CAL*n*)

Result of one of the calibration iterations.

Address: 4005_C000h base + 180h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DCOC_CAL_RES_Q												0				DCOC_CAL_RES_I											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DCOC_CAL*n* field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DCOC_CAL_ RES_Q	DCOC Calibration Result - Q Channel Q channel DCOC calibration result. This 12bit signed value represents the DCOC's Q channel DC estimate for the nth calibration gain setting.This is provided for debug/chacterization purposes.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DCOC_CAL_ RES_I	DCOC Calibration Result - I Channel I channel DCOC calibration result. This 12bit signed value represents the DCOC's I channel DC estimate for the nth calibration gain setting.This is provided for debug/chacterization purposes.

52.1.45 Receive Channel Filter Coefficient (XCVR_RX_CHF_COEF*n*)

Receive channel filter coefficient (8-bit signed fractional)

Address: 4005_C000h base + 1A0h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RX_CH_FILT_HX							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_RX_CHF_COEF*n* field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_CH_FILT_ HX	RX Channel Filter Coefficient Receive channel filter coefficient (8-bit signed fractional)

52.1.46 TX Digital Control (XCVR_TX_DIG_CTRL)

Address: 4005_C000h base + 200h offset = 4005_C200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FREQ_WORD_ADJ										0	DP_SEL	0			POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TONE_SEL		0	DFT_CLK_SEL			LFSR_EN	DFT_LFSR_LEN			DFT_EN	DFT_MODE		
W																
Reset	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

XCVR_TX_DIG_CTRL field descriptions

Field	Description
31–22 FREQ_WORD_ ADJ	GFSK Frequency Word Adjustment This register is a signed 9 bit number that is added to the GFSK modulator output. This allows the GFSK modulation to be adjusted, or skewed, by a range of -512 to +511.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_TX_DIG_CTRL field descriptions (continued)

Field	Description
20 DP_SEL	Data Padding Pattern Select In normal user GSFK modes the BLE link layer overrides this bit to match the first bit of the packet preamble. In Radio Protocol 7, 128 Channel GFSK mode, this register bit is active and selects the 8 bits of data padding. For DFT GFSK modes, this register bit is active and the unselected data padding pattern is added as an additional 8 bits of padding after the selected pattern is shifted out. For 802.15.4 Radio Protocols, this register bit selects the data padding pattern to be used as Symbols and converted to Chips for padding. In all cases the LSB is the first bit shifted out as data padding. 0 Selects DATA_PADDING_PATTERN_0 as the source for data padding 1 Selects DATA_PADDING_PATTERN_1 as the source for data padding
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 POL	Oversample Clock Capture Polarity This bit selects between Even or Odd reference clock cycles to sample the TX Modulators output values. The TX Modulator to sample is selected by the Radio Protocol and the Radio DFT Mode, and can be either the GFSK modulator, the FSK modulator, or the DFT Tone modulator. The resulting Oversampled Modulation is the Baseband Frequency Word presented to the PLL 0 Selects Even clock cycle 1 Selects Odd clock cycle, a one cycle delay
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–12 TONE_SEL	DFT Tone Selection This register selects the DFT Tone register to use as the Modulation Source in DFT Manual Tone mode. The selected Tone will be added as a constant frequency offset to the current Radio frequency. Software can change the tone selection to modulate the Radio frequency at a software derived frequency. Note that the DFT LFSR Tone mode can be used to randomly modulate with tones at the DFT Clock frequency. 00 DFT Tone 0 01 DFT Tone 1 10 DFT Tone 2 11 DFT Tone 3
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 DFT_CLK_SEL	DFT Clock Selection This register selects the frequency of the DFT clock that is used to shift out the DFT Modulation Pattern and also used to clock the LFSR in DFT LFSR modes. 000 62.5 kHz 001 125 kHz 010 250 kHz 011 500 kHz 100 1 MHz 101 2 MHz

Table continues on the next page...

XCVR_TX_DIG_CTRL field descriptions (continued)

Field	Description
	110 4 MHz 111 Clock is off
7 LFSR_EN	DFT LFSR Enable This bit enables the DFT LFSR for use as a source for Data Padding and TX Data in LFSR Data mode, for use as TX Symbols in LFSR Symbol mode, and for use as the Tone Selector in LFSR Tone mode. Effectively this is the on-off switch for modulation in these modes. Note that the LFSR is clocked at the DFT Clock frequency.
6–4 DFT_LFSR_LEN	DFT LFSR Length This register selects the length of the DFT LFSR and the associated LFSR Tap Mask. The Mask is in the form of [MSB...LSB] 000 LFSR 9, tap mask 100010000 001 LFSR 10, tap mask 1001000000 010 LFSR 11, tap mask 11101000000 011 LFSR 13, tap mask 1101100000000 100 LFSR 15, tap mask 111010000000000 101 LFSR 17, tap mask 11110000000000000 110 Reserved 111 Reserved
3 DFT_EN	Radio DFT Mode Enable If the Radio is in DFT Pattern Register mode, then this bit is used to turn on and off the modulation.
DFT_MODE	Radio DFT Modes This register selects the Radio DFT mode as described below. In addition to setting the Radio DFT mode, the DFT LFSR needs to be configured, and the Radio Protocol needs to be chosen. Note that the LFSR Symbols mode is not supported for GFSK Protocols, and the Tone modes bypass both the GFSK and the FSK modulators. 000 Normal Radio Operation. DFT not engaged. 001 Pattern Register Mode. TX DFT Modulation Pattern Register is shifted out as the transmission data stream. Note that the DFT_EN bit must be set. 010 LFSR Data Mode. TX LFSR is used as the transmission data stream. Note that the LFSR_EN bit must be set. 011 LFSR Symbol Mode. TX LFSR is used to create 802.15.4 symbols which are then converted to Chips and transmitted. Note that the LFSR_EN bit must be set. 100 Not implemented on Apache 1.0, future use will allow a package pin to be used as the source of the TX data stream. Note that the DFT_EN bit must be set. 101 Constant Frequency Mode. No data modulation is done, Radio transmits at the channel frequency selected. 110 LFSR Tone Mode. TX LFSR is used to select the DFT Tone register to transmit, LFSR_EN bit must be set. 111 Manual Tone Mode. TONE_SEL is used to select the DFT Tone register to transmit.

52.1.47 TX Data Padding Pattern (XCVR_TX_DATA_PAD_PAT)

Address: 4005_C000h base + 204h offset = 4005_C204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LRM	DFT_LFSR_OUT														
W																
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA_PADDING_PAT_1								DATA_PADDING_PAT_0							
W																
Reset	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0

XCVR_TX_DATA_PAD_PAT field descriptions

Field	Description
31 LRM	LFSR Reset Mask When this bit is set the DFT LFSR will not be reset when LFSR_EN is cleared and will instead continue to repeat its sequence as defined by the DFT_LFSR_LEN bits when LFSR_EN is next set. When this bit is cleared the DFT LFSR will reset every time LFSR_EN is cleared.
30–16 DFT_LFSR_OUT	Transmit DFT LFSR Output This register can be read to observe the current value of the DFT LFSR, only bits [14:0] are available.
15–8 DATA_ PADDING_PAT_ 1	Data Padding Pattern 1 These bits are used for Data Padding when Pattern 1 is selected, the LSB is the first bit shifted out as padding.
DATA_ PADDING_PAT_ 0	Data Padding Pattern 0 These bits are used for Data Padding when Pattern 0 is selected, the LSB is the first bit shifted out as padding.

52.1.48 TX GFSK Modulation Control (XCVR_TX_GFSK_MOD_CTRL)

Address: 4005_C000h base + 208h offset = 4005_C208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			GFSK_FLD	0	GFSK_SYMBOL_RATE			0			GFSK_MLD	0		GFSK_MI	
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GFSK_MULTIPLY_TABLE_MANUAL															
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TX_GFSK_MOD_CTRL field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 GFSK_FLD	GFSK Filter Lookup Table Disable If this bit is set, the internal GFSK filter coefficients that are normally derived from a lookup table based on the reference clock frequency, are disabled, and the coefficients are instead derived from the GFSK_FILTER_COEFF_MANUAL1 and GFSK_FILTER_COEFF_MANUAL2 registers.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 GFSK_ SYMBOL_RATE	GFSK Symbol Rate This register selects the GFSK Symbol Rate which together with the GFSK Modulation Index determines the Peak Modulation frequency.The formula used for the Peak Modulation is (Symbol Rate / (2 x 1/ Modulation Index)) 000 50 kHz 001 100 kHz 010 200 kHz 011 1 MHz 100 2 MHz 101 Reserved 110 Reserved 111 Reserved
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 GFSK_MLD	GFSK Multiply Lookup Table Disable

Table continues on the next page...

XCVR_TX_GFSK_MOD_CTRL field descriptions (continued)

Field	Description
	If this bit is set, the GFSK Multiply Lookup table is disabled and GFSK_MULTIPLY_TABLE_MANUAL is used instead.
19–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 GFSK_MI	GFSK Modulation Index This register selects the GFSK Modulation Index which together with the GFSK Symbol Rate determines the Peak Modulation frequency.The formula used for the Peak Modulation is (Symbol Rate / (2 x 1/ Modulation Index)) 00 0.32 01 0.50 10 0.80 11 1.00
GFSK_ MULTIPLY_ TABLE_MANUAL	GFSK Multiply Lookup Table Override Value The GFSK Modulator Multiplier uses a lookup table to select the multiplicand representing the (Frequency Deviation divided by the Low Port Sigma Delta LSB resolution in Hz) for the Modulation requested based on the Modulation Index, the Symbol Rate, and the Reference Clock Frequency. The lookup table value is overridden by this register if GFSK_MLD is set, and these bits should then contain a number that represents {FDev/SD_LSB integer[11:0] + FDev/SD_LSB fraction[3:0]}

52.1.49 TX GFSK Filter Coefficients 2 (XCVR_TX_GFSK_COEFF2)

The two registers TX_GFSK_COEFF1 and TX_GFSK_COEFF2 form a 64-bit little endian register that is memory mapped internally as shown to override the GFSK Filter Coefficients.

In 64-bit they combine as {TX_GFSK_COEFF2[31:0],TX_GFSK_COEFF1[31:0]}

The resulting 64-bit value is GFSK Manual Filter Coefficient [63:0]

Address: 4005_C000h base + 20Ch offset = 4005_C20Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GFSK_FILTER_COEFF_MANUAL2																															
Reset	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

XCVR_TX_GFSK_COEFF2 field descriptions

Field	Description
GFSK_FILTER_ COEFF_ MANUAL2	GFSK Manual Filter Coefficients[63:32] If GFSK_FLD is set, the GFSK Filter Coefficients are overridden using the bits as described below, and since the filter is symmetrical each coefficient is used twice.

XCVR_TX_GFSK_COEFF2 field descriptions (continued)

Field	Description

Bits	Filter Coefficients
[36:32]	Filter coeff 0 and 15
[45:40]	Filter coeff 1 and 14
[55:48]	Filter coeff 4 and 11
[63:56]	Filter coeff 5 and 10

52.1.50 TX GFSK Filter Coefficients 1 (XCVR_TX_GFSK_COEFF1)

The two registers TX_GFSK_COEFF1 and TX_GFSK_COEFF2 form a 64-bit little endian register that is memory mapped internally as shown to override the GFSK Filter Coefficients.

In 64-bit they combine as {TX_GFSK_COEFF2[31:0],TX_GFSK_COEFF1[31:0]}

The resulting 64-bit value is GFSK Manual Filter Coefficient [63:0]

Address: 4005_C000h base + 210h offset = 4005_C210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GFSK_FILTER_COEFF_MANUAL1																															
Reset	1	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	1	1	0	1

XCVR_TX_GFSK_COEFF1 field descriptions

Field	Description
GFSK_FILTER_COEFF_MANUAL1	GFSK Manual Filter Coefficient [31:0] If GFSK_FLD is set, the GFSK Filter Coefficients are overridden using the bits as described below, and since the filter is symmetrical each coefficient is used twice.

Bits	000 Filter Coefficients
[6:0]	Filter coeff 2 and 13
[15:7]	Filter coeff 6 and 9
[22:16]	Filter coeff 3 and 12
[31:23]	Filter coeff 7 and 8

52.1.51 TX FSK Modulation Scale (XCVR_TX_FSK_MOD_SCALE)

Address: 4005_C000h base + 214h offset = 4005_C214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FSK_MODULATION_SCALE_1													0			FSK_MODULATION_SCALE_0												
W																																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

XCVR_TX_FSK_MOD_SCALE field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–16 FSK_ MODULATION_ SCALE_1	FSK Modulation Scale for a data 1 This register is used to provide the modulation level for a data 1 in 802.15.4 Protocols. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Modulation Scale will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz. The reset value of this little endian register is 0x07FF which is the 13-bit value of 0_0111_1111_1111 The two's complement of this reset value is 2047 The range of signed values that this register supports is -4096 to 4095
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FSK_ MODULATION_ SCALE_0	FSK Modulation Scale for a data 0 This register is used to provide the modulation level for a data 0 in 802.15.4 Protocols. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Modulation Scale will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz. The reset value of this little endian register is 0x1800 which is the 13-bit value of 1_1000_0000_0000 The two's complement of this reset value is -2048 The range of signed values that this register supports is -4096 to 4095

52.1.52 TX DFT Modulation Pattern (XCVR_TX_DFT_MOD_PAT)

Address: 4005_C000h base + 218h offset = 4005_C218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DFT_MOD_PATTERN																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TX_DFT_MOD_PAT field descriptions

Field	Description
DFT_MOD_ PATTERN	DFT Modulation Pattern In DFT Pattern Register mode, if DFT_EN is set, the bits in this register will be shifted out as the DFT Modulation Data in a repeating loop starting with bit [0].

52.1.53 TX DFT Tones 0 and 1 (XCVR_TX_DFT_TONE_0_1)

The DFT Tone Registers are used in DFT Tone modes to bypass the GFSK and FSK Modulators and instead present the selected Tone to the PLL as a constant tone modulation.

The two's complement range of these little endian registers is -4096 to 4095

Address: 4005_C000h base + 21Ch offset = 4005_C21Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			DFT_TONE_0													0			DFT_TONE_1												
W																																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TX_DFT_TONE_0_1 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–16 DFT_TONE_0	DFT Tone 0 This register is used to provide the modulation level in DFT Tone Modes. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Tone will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DFT_TONE_1	DFT Tone 1 This register is used to provide the modulation level in DFT Tone Modes. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Tone will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz.

52.1.54 TX DFT Tones 2 and 3 (XCVR_TX_DFT_TONE_2_3)

The DFT Tone Registers are used in DFT Tone modes to bypass the GFSK and FSK Modulators and instead present the selected Tone to the PLL as a constant tone modulation.

The two's complement range of these little endian registers is -4096 to 4095

Address: 4005_C000h base + 220h offset = 4005_C220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			DFT_TONE_2													0			DFT_TONE_3												
W																																
Reset	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

XCVR_TX_DFT_TONE_2_3 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–16 DFT_TONE_2	DFT Tone 2 This register is used to provide the modulation level in DFT Tone Modes. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Tone will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DFT_TONE_3	DFT Tone 3 This register is used to provide the modulation level in DFT Tone Modes. The signed 12 bit value that is represented by this register is presented to the PLL as the Baseband Frequency Word. The frequency result for this Tone will be the value of this register times the PLL Low Port Modulator Frequency Resolution; a typical value for the Frequency Resolution is 244.14 Hz.

52.1.55 PLL Modulation Overrides (XCVR_PLL_MOD_OVRD)

Address: 4005_C000h base + 228h offset = 4005_C228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HPM_LSB_DIS	0	HPM_LSB_MANUAL		HPM_BANK_DIS	0			HPM_BANK_MANUAL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MOD_DIS	0		MODULATION_WORD_MANUAL												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_MOD_OVRD field descriptions

Field	Description
31 HPM_LSB_DIS	Disable HPM LSB If this bit is set, the High Port Sigma Delta Modulator output is disabled, and the High Port LSB value applied to the VCO comes from the HPM_LSB_MANUAL register.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–28 HPM_LSB_ MANUAL	Manual HPM LSB If HPM_LSB_DIS is set, this register is the value that is applied to the VCO High Port LSB.
27 HPM_BANK_DIS	Disable HPM Bank If this bit is set, the High Port Bank Modulation is disabled, and the High Port Bank value applied to the VCO comes from the HPM_BANK_MANUAL register.
26–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 HPM_BANK_ MANUAL	Manual HPM bank If HPM_BANK_DIS is set, this register is the value that is applied to the VCO High Port Bank.
15 MOD_DIS	Disable Modulation Word If this bit is set, any modulation from the TX Digital is disabled and the source of the Baseband Frequency Word is the MODULATION_WORD_MANUAL register.
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MODULATION_ WORD_MANUAL	Manual Modulation Word If MOD_DIS is set, the signed 12 bit value that is represented by this register is the Baseband Frequency Word.

52.1.56 PLL Channel Mapping (XCVR_PLL_CHAN_MAP)

Address: 4005_C000h base + 22Ch offset = 4005_C22Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ZOC	BMR	BOC	0	CHANNEL_NUM						
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

XCVR_PLL_CHAN_MAP field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 ZOC	Zigbee Channel Number Override This bit controls the source of the Zigbee channel selection. 0 Zigbee channel number comes from the 802.15.4 Link Layer. 1 Zigbee channel number comes from the CHANNEL_NUM register
9 BMR	BLE MBAN Channel Remap This bit controls the mapping of BLE channel 39 in Radio Protocol 2, BLE overlap MBAN mode. 0 BLE channel 39 is mapped to BLE channel 39, 2.480 GHz 1 BLE channel 39 is mapped to MBAN channel 39, 2.399 GHz
8 BOC	BLE Channel Number Override This bit controls the source of the BLE channel selection. 0 BLE channel number comes from the BLE Link Layer 1 BLE channel number comes from the CHANNEL_NUM register
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CHANNEL_NUM	Protocol specific Channel Number for PLL Frequency Mapping When this register is active, it can be used to directly select a Protocol specific Channel Number, which is mapped internally to the correct Radio Carrier Frequency for PLL tuning. The internal mapping is detailed in the table below. This register is active when BOC or ZOC are set along with their corresponding Radio Protocols, and this register is also active in the 128 Channel FSK and GFSK protocols. The Radio Carrier Frequency in the 128 Channel FSK and GFSK protocols can be calculated using the formula below: Radio Carrier Frequency = (Channel Number + 2360) x 1 MHz The Radio Channel Frequency can also be selected by setting the SDM_MAP_DIS bit in the PLL_LP_SDM_CTRL1 register along with the LPM_INTG, LPM_NUM, and LPM_DENOM registers to get a frequency that equals ((Reference Clock Frequency x 2) x (LPM_INTG + (LPM_NUM / LPM_DENOM)) This table shows the internal mapping by Protocol of the Channel Numbers to the Radio Carrier Frequency (MHz).

Channel Number	000 BLE	001 BLE in MBAN	010 BLE Overlap MBAN	100 Zigbee	101 802.15.4j
0	2402	2360	2402	2400	2363
1	2404	2361	2404	2400	2368
2	2406	2362	2406	2400	2373
3	2408	2363	2408	2400	2378
4	2410	2364	2410	2400	2383
5	2412	2365	2412	2400	2388
6	2414	2366	2414	2400	2393

Table continues on the next page...

XCVR_PLL_CHAN_MAP field descriptions (continued)

Field	Description

Channel Number	000 BLE	001 BLE in MBAN	010 BLE Overlap MBAN	100 Zigbee	101 802.15.4j
7	2416	2367	2416	2400	2367
8	2418	2368	2418	2400	2372
9	2420	2369	2420	2400	2377
10	2422	2370	2422	2400	2382
11	2424	2371	2424	2405	2387
12	2426	2372	2426	2410	2392
13	2428	2373	2428	2415	2397
14	2430	2374	2430	2420	2395
15	2432	2375	2432	2425	2399
16	2434	2376	2434	2430	2399
17	2436	2377	2436	2435	2399
18	2438	2378	2438	2440	2399
19	2440	2379	2440	2445	2399
20	2442	2380	2442	2450	2399
21	2444	2381	2444	2455	2399
22	2446	2382	2446	2460	2399
23	2448	2383	2448	2465	2399
24	2450	2384	2450	2470	2399
25	2452	2385	2452	2475	2399
26	2454	2386	2454	2480	2399
27	2456	2387	2456	2400	2399
28	2458	2388	2458	2400	2399
29	2460	2389	2460	2400	2399
30	2462	2390	2390	2400	2399
31	2464	2391	2391	2400	2399
32	2466	2392	2392	2400	2399
33	2468	2393	2393	2400	2399
34	2470	2394	2394	2400	2399
35	2472	2395	2395	2400	2399
36	2474	2396	2396	2400	2399
37	2476	2397	2397	2400	2399
38	2478	2398	2398	2400	2399
39	2480	2399	2480 or 2399 *	2400	2399
Default	2400	2399	2399	2400	2399

* The BLE MBAN Channel Remap bit, BMR, controls the frequency mapping in this case.

52.1.57 PLL Lock Detect (XCVR_PLL_LOCK_DETECT)

Address: 4005_C000h base + 230h offset = 4005_C230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FTW_TX	0	FTF_TX_THRSH						FTW_RX	0	FTF_RX_THRSH	
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FTF_RX_THRSH				CTUNE_LDF_LEV				TAFF	0	FTFF	FT_FAIL	CSFF	CS_FAIL	CTFF	CT_FAIL
W									w1c		w1c		w1c		w1c	
Reset	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0

XCVR_PLL_LOCK_DETECT field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 FTW_TX	TX Frequency Target Window time select In Radio Transmit Mode, this bit selects the length of time to count for the estimation of PLL lock frequency, which is compared with the Frequency Target Window, set by FTF_TX. 0 4 us 1 8 us
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–20 FTF_TX_THRSH	TX Frequency Target Fail Threshold In Radio Transmit Mode, the FT_FAIL and FTFF bits will be set after the Frequency Target Count completes if the absolute value of the count difference from the frequency target count is greater than this register value.
19 FTW_RX	RX Frequency Target Window time select In Radio Receive Mode, this bit selects the length of time to count for the estimation of PLL lock frequency, which is compared with the Frequency Target Window, set by FTF_RX.

Table continues on the next page...

XCVR_PLL_LOCK_DETECT field descriptions (continued)

Field	Description
	0 4 us 1 8 us
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–12 FTF_RX_THRSH	RX Frequency Target Fail Threshold In Radio Receive Mode, the FT_FAIL and FTFF bits will be set after the Frequency Target Count completes if the absolute value of the count difference from the frequency target count is greater than this register value.
11–8 CTUNE_LDF_LEV	CTUNE Lock Detect Fail Level The CT_FAIL and CTFF bits will be set after Coarse Tune Calibration completes if the absolute value of the Coarse Tune best count difference (CTUNE_BEST_DIFF in the PLL_CTUNE_RESULTS register) is greater than this register value.
7 TAFF	TSM Abort Failure Flag This bit is set if the TSM Sequence Aborts, and this bit is cleared by writing a 1 to it.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 FTFF	Frequency Target Failure Flag This bit is set when FT_FAIL is first set, and this bit is cleared by writing a 1 to it.
4 FT_FAIL	Real time status of Frequency Target Failure If the Frequency Target Count has completed and the count was out of the range selected by FTW_TX or FTW_RX, then this bit will be set.
3 CSFF	Cycle Slip Failure Flag, held until cleared This bit is set when CS_FAIL is first set, and this bit is cleared by writing a 1 to it.
2 CS_FAIL	Real time status of Cycle Slip circuit This bit shows the real-time status of the Cycle Slip State Machine Interrupt which is configured using the control bits in the PLL_HPM_CAL1 and PLL_HPM_CAL2 registers.
1 CTFF	CTUNE Failure Flag, held until cleared This bit is set when CT_FAIL is first set, and this bit is cleared by writing a 1 to it.
0 CT_FAIL	Real time status of Coarse Tune Fail signal If the Coarse Tune Calibration has completed and the best count difference is out of the range selected by CTUNE_LDF_LEV, then this bit will be set.

52.1.58 PLL High Port Modulation Control (XCVR_PLL_HP_MOD_CTRL)

Address: 4005_C000h base + 234h offset = 4005_C234h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HP_MOD_INV	0					HPM_SCALE		HPM_DTH_EN	0		HP_DTH_SCL	0	HPM_LFSR_LEN		
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HP_SDM_DIS	HP_SDM_INV	HPFF	0			HPM_SDM_MANUAL									
W			w1c													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_HP_MOD_CTRL field descriptions

Field	Description
31 HP_MOD_INV	HPM Invert If this bit is set, the Baseband Frequency Word will be Inverted before Scaling and Dithering, and then applied to the High Port Bank and the High Port Sigma Delta Modulator of the HP LSB. Note that the High Port SDM has it's own, additional, inversion bit.
30–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 HPM_SCALE	HPM Scale Factor This register controls the scaling of the High Port Modulation Integer Value applied to the VCO High Port Bank. 00 No Scaling 01 Multiply by 2 10 Divide by 2 11 Reserved

Table continues on the next page...

XCVR_PLL_HP_MOD_CTRL field descriptions (continued)

Field	Description
23 HPM_DTH_EN	Dither Enable for HPM LFSR If this bit is set, the High Port LSB Fractional Word will be Dithered by the High Port LFSR before it is applied to the HP SDM.
22–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 HP_DTH_SCL	HPM Dither Scale If this bit is set, the LFSR dithering of the High Port LSB Fractional Word will be multiplied by 2.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 HPM_LFSR_LEN	HPM LFSR Length This register selects the length of the HPM LFSR and the associated LFSR Tap Mask 000 LFSR 9, tap mask 100010000 001 LFSR 10, tap mask 1001000000 010 LFSR 11, tap mask 11101000000 011 LFSR 13, tap mask 1101100000000 100 LFSR 15, tap mask 111010000000000 101 LFSR 17, tap mask 11110000000000000 110 Reserved 111 Reserved
15 HP_SDM_DIS	Disable HPM SDM If this bit is set, the High Port Modulation to the HP SDM is disabled, and the HP SDM Numerator comes from the HPM_SDM_MANUAL register.
14 HP_SDM_INV	Invert HPM SDM If this bit is set the High Port LSB Fractional Word, including any Dithering, will be Inverted before it is applied to the High Port Sigma Delta Modulator.
13 HPFF	HPM SDM Invalid Flag This bit is set if the High Port Sigma Delta Modulator output is invalid due to an error in the Fraction applied, and this bit is cleared by writing a 1 to it.
12–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HPM_SDM_ MANUAL	PLL HPM SDM MANUAL If HP_SDM_DIS is set, this register is the value that is applied as the Numerator to the High Port SDM.

52.1.59 PLL HPM Calibration Control (XCVR_PLL_HPM_CAL_CTRL)

Address: 4005_C000h base + 238h offset = 4005_C238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HP_CAL_TIME	HP_CAL_ARY	0	HPM_CAL_FACTOR_MANUAL												
W			▨													
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HP_CAL_DIS	0		HPM_CAL_FACTOR												
W		▨		▨												
Reset	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0

XCVR_PLL_HPM_CAL_CTRL field descriptions

Field	Description
31 HP_CAL_TIME	High Port Modulation Calibration Time This bit selects the length of time to count during HPM Calibration and is used by the math implemented in the HPM Calibration Factor lookup table. 0 25 us 1 50 us
30 HP_CAL_ARY	High Port Modulation Calibration Array Size This bit selects the size of the array to be used by the math implemented in the HPM Calibration Factor lookup table. 0 128 1 256
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_PLL_HPM_CAL_CTRL field descriptions (continued)

Field	Description
28–16 HPM_CAL_ FACTOR_ MANUAL	HPM Manual Calibration Factor If HP_CAL_DIS is set, this register is the unsigned 13 bit value used by the HPM Multiplier, the maximum useable value is decimal 6400.
15 HP_CAL_DIS	If this bit is set, the lookup table value for the HPM Calibration Factor is overridden by the HPM_CAL_FACTOR_MANUAL register.
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HPM_CAL_ FACTOR	High Port Modulation Calibration Factor This is the value currently being used by High Port Modulation Multiplier.

52.1.60 PLL Cycle Slip Lock Detect Configuration and HPM Calibration 1 (XCVR_PLL_LD_HPM_CAL1)

Address: 4005_C000h base + 23Ch offset = 4005_C23Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CS_FCNT				0	CS_FW			0	CS_WT			0			CNT_1
W																
Reset	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNT_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LD_HPM_CAL1 field descriptions

Field	Description
31–28 CS_FCNT	Cycle Slip Flag Count This register value is the maximum number of Cycle Slip Flags that can be counted before the CS_FAIL is set.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 CS_FW	Cycle Slip Flag Window This register sets the window time for capturing Cycle Slip Flags before shutting down. 000 8 us 001 16 us 010 24 us 011 32 us 100 64 us

Table continues on the next page...

XCVR_PLL_LD_HPM_CAL1 field descriptions (continued)

Field	Description
	101 96 us 110 128 us 111 256 us
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–20 CS_WT	Cycle Slip Wait Time This register sets the time to wait before restarting a Cycle Slip Search if CS_RC, Cycle Slip Recycle, is set. 000 128 us 001 256 us 010 384 us 011 512 us 100 640 us 101 768 us 110 896 us 111 1024 us
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CNT_1	High Port Modulation Counter Value 1 This is the Ripple counter value used for HPM Cal 1.

52.1.61 PLL Cycle Slip Lock Detect Configuration and HPM Calibration 2 (XCVR_PLL_LD_HPM_CAL2)

Address: 4005_C000h base + 240h offset = 4005_C240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			CS_FT					0			CS_RC	0			CNT_2
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNT_2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LD_HPM_CAL2 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 CS_FT	Cycle Slip Flag Timeout This register sets the time to hold the analog cycle slip circuit in reset after a Cycle Slip Flag, the time is a number of counts of the reference clock equal to the value of this register.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 CS_RC	Cycle Slip Recycle If this is bit is set, the Cycle Slip Lock Detector will restart after the Cycle Slip Wait time.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CNT_2	High Port Modulation Counter Value 2 This is the Ripple counter value used for HPM Cal 2.

52.1.62 PLL HPM SDM Fraction (XCVR_PLL_HPM_SDM_FRACTION)

Address: 4005_C000h base + 244h offset = 4005_C244h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						HPM_DENOM										0						HPM_NUM_SELECTED									
W																																
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_HPM_SDM_FRACTION field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–16 HPM_DENOM	High Port Modulation Denominator This is the denominator that is currently being applied to the High Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6 The High Port Sigma Delta Modulator LSB in Hz can be calculated as follows: HP SDM LSB Resolution = High Port Array LSB in Hz / 2^6 = 6e3/256 = 93.75 Hz The High Port Array LSB is expected to be 6 kHz after calibration.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HPM_NUM_ SELECTED	HPM_NUM_SELECTED This is the numerator that is currently being applied to the High Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

52.1.63 PLL Low Port Modulation Control (XCVR_PLL_LP_MOD_CTRL)

Address: 4005_C000h base + 248h offset = 4005_C248h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				LPM_SCALE				LPM_D_OVRD	LPM_D_CTRL	0		LPM_DTH_SCL			
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPM_SDM_DIS	LPM_SDM_INV	LPFF	0	PLL_LD_DIS	0					PLL_LOOP_DIVIDER_MANUAL					
W			w1c													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LP_MOD_CTRL field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 LPM_SCALE	LPM Scale Factor This register controls the scaling of the Baseband Frequency Word and is used to match the Modulation Frequency Deviation required to the Low Port Sigma Delta Modulator LSB size in Hz. 0000 No Scaling 0001 Multiply by 2 0010 Multiply by 4 0011 Multiply by 8 0100 Multiply by 16 0101 Multiply by 32 0110 Multiply by 64 0111 Multiply by 128

Table continues on the next page...

XCVR_PLL_LP_MOD_CTRL field descriptions (continued)

Field	Description
	1000 Multiply by 256 1001 Multiply by 512 1010 Multiply by 1024 1011 Multiply by 2048 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
23 LPM_D_OVRD	LPM Dither Override Mode Select When this bit is set, the Scaled Baseband Frequency Word applied to the Low Port Sigma Delta Modulator will be dithered if LPM_D_CTRL is set, and not dithered if LPM_D_CTRL is cleared. If this bit is cleared, then the LPM Numerator will be dithered in Radio Receive mode, and also when the LPM Numerator approaches an Integer value in order to preserve the validity of the Sigma Delta Modulator output.
22 LPM_D_CTRL	LPM Dither Control in Override Mode If LPM_D_OVRD is set, this bit turns LPM Dithering on and off.
21–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 LPM_DTH_SCL	LPM Dither Scale This register controls the scale of the Dithering added to the Scaled Baseband Frequency Word before it is applied to the Low Port Sigma Delta Modulator as the LPM Numerator. The unit for the ranges shown below is the LP SDM LSB in Hz. 0000 Reserved 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 -128 to 96 0110 -256 to 192 0111 -512 to 384 1000 -1024 to 768 1001 -2048 to 1536 1010 -4096 to 3072 1011 -8192 to 6144 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
15 LPM_SDM_DIS	Disable LPM SDM This bit controls the Modulation of the Low Port Sigma Delta. If this bit is set, the Low Port Sigma Delta will be active and control the PLL to maintain a steady frequency based on the current Integer, Numerator, and Denominator values that are being applied. Modulation and Dithering will be disabled.
14 LPM_SDM_INV	Invert LPM SDM

Table continues on the next page...

XCVR_PLL_LP_MOD_CTRL field descriptions (continued)

Field	Description
	If this bit is set the Scaled Baseband Frequency Word, including any Dithering, will be Inverted before it is applied to the Low Port Sigma Delta Modulator.
13 LPFF	LPM SDM Invalid Flag This bit is set if the Low Port Sigma Delta Modulator output is invalid due to an error in the Fraction applied, and this bit is cleared by writing a 1 to it.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 PLL_LD_DIS	PLL Loop Divider Disable If this bit is set, the Low Port Sigma Delta Modulator output is disabled, and the PLL Loop Divider value applied to the PLL comes from the PLL_LOOP_DIVIDER_MANUAL register.
10–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PLL_LOOP_ DIVIDER_ MANUAL	PLL Loop Divider Manual If PLL_LD_DIS is set, this register is the value that is applied to the PLL Loop Divider i.

52.1.64 PLL Low Port SDM Control 1 (XCVR_PLL_LP_SDM_CTRL1)

Address: 4005_C000h base + 24Ch offset = 4005_C24Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SDM_MAP_DIS	0								LPM_INTG						
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									LPM_INTG_SELECTED						
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0

XCVR_PLL_LP_SDM_CTRL1 field descriptions

Field	Description
31 SDM_MAP_DIS	SDM Mapping Disable If this bit is set, the Low Port Sigma Delta Modulator internal frequency mapping based on Protocol specific channel numbers is disabled, and the Radio Channel Frequency is selected by setting the LPM_INTG, LPM_NUM, and LPM_DENOM registers to get a frequency that equals ((Reference Clock Frequency x 2) x (LPM_INTG + (LPM_NUM / LPM_DENOM))
30–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 LPM_INTG	Low Port Modulation Integer Manual Value If SDM_MAP_DIS is set, this register is the value that is applied to the Low Port Sigma Delta Modulator for the Integer, the nominal range is 36 to 39 in decimal.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPM_INTG_ SELECTED	Low Port Modulation Integer Value Selected This shows the integer value that is currently being applied to the low port sigma delta modulator.

52.1.65 PLL Low Port SDM Control 2 (XCVR_PLL_LP_SDM_CTRL2)

Address: 4005_C000h base + 250h offset = 4005_C250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LPM_NUM																											
W																																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LP_SDM_CTRL2 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPM_NUM	Low Port Modulation Numerator If SDM_MAP_DIS is set, this register is the signed 27 bit value that is applied to the Low Port Sigma Delta Modulator for the Numerator. For valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

52.1.66 PLL Low Port SDM Control 3 (XCVR_PLL_LP_SDM_CTRL3)

Address: 4005_C000h base + 254h offset = 4005_C254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LPM_DENOM																											
W																																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LP_SDM_CTRL3 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPM_DENOM	Low Port Modulation Denominator If SDM_MAP_DIS is set, this register is the signed 27 bit value that is applied to the Low Port Sigma Delta Modulator for the Denominator. For valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6 The Low Port Sigma Delta Modulator LSB in Hz can be calculated as follows: LP SDM LSB Resolution = Reference Clock Frequency / LPM_DENOM value The default value of LPM_DENOM for a 32 MHz reference is 2 raised to the 26th power, so the SDM LSB is 0.4768 Hz

52.1.67 PLL Low Port SDM Numerator Applied (XCVR_PLL_LP_SDM_NUM)

Address: 4005_C000h base + 258h offset = 4005_C258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LPM_NUM_SELECTED																											
W																																
Reset	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LP_SDM_NUM field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPM_NUM_ SELECTED	Low Port Modulation Numerator Applied This is the value that is currently being applied to the sigma delta modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

52.1.68 PLL Low Port SDM Denominator Applied (XCVR_PLL_LP_SDM_DENOM)

Address: 4005_C000h base + 25Ch offset = 4005_C25Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LPM_DENOM_SELECTED																											
W																																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_LP_SDM_DENOM field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPM_DENOM_ SELECTED	Low Port Modulation Denominator Selected This is the value that is currently being applied to the sigma delta modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

52.1.69 PLL Delay Matching (XCVR_PLL_DELAY_MATCH)

Address: 4005_C000h base + 260h offset = 4005_C260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												HPM_BANK_ DELAY				0				HPM_SDM_ DELAY				0				LP_SDM_ DELAY			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

XCVR_PLL_DELAY_MATCH field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 HPM_BANK_ DELAY	HPM Bank Delay This register selects the number of clock cycles of the (PLL Sigma Delta Clock divided by 2) to delay the High Port modulation of the VCO High Port Bank Array.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 HPM_SDM_ DELAY	HPM_SDM_DELAY This register selects the number of clock cycles of the (PLL Sigma Delta Clock divided by 2) to delay the High Port Sigma Delta modulation of the VCO High Port Bank LSB. Note that the High Port SDM is clocked by the (PLL Sigma Delta Clock) but the modulation is based on a divide by 2 version of this same clock.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LP_SDM_DELAY	LP_SDM_DELAY This register selects the number of clock cycles of the (PLL Sigma Delta Clock) to delay the Low Port Sigma Delta modulation of the PLL Loop Divider.

52.1.70 PLL Coarse Tune Control (XCVR_PLL_CTUNE_CTRL)

Address: 4005_C000h base + 264h offset = 4005_C264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CTUNE_DIS	CTUNE_MANUAL							0				CTUNE_ADJUST			
W									▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTUNE_TD	0			CTUNE_TARGET_MANUAL											
W		▨														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_CTUNE_CTRL field descriptions

Field	Description
31 CTUNE_DIS	CTUNE Disable If this bit is set, the Coarse Tune Band applied to the VCO comes from the CTUNE_MANUAL register.
30–24 CTUNE_ MANUAL	CTUNE Manual If CTUNE_DIS is set, this register is the value that is applied to the VCO as the Coarse Tune Band.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 CTUNE_ADJUST	CTUNE Count Adjustment The PLL 17-bit Ripple Counter count used in Coarse Tune Calibration will be increased by the value of this register in order to allow for any consistent under-counting that might present itself in the analog circuit.
15 CTUNE_TD	CTUNE Target Disable If this bit is set, the Frequency Target presented to the Coarse Tune Calibrator comes from the CTUNE_TARGET_MANUAL register.
14–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTUNE_ TARGET_ MANUAL	CTUNE Target Manual If CTUNE_TD is set, this register is the value that is presented to the Coarse Tune Calibrator as the Frequency Target in MHz. The nominal range of this target is from 2360 to 2487 in decimal.

52.1.71 PLL Coarse Tune Count 6 (XCVR_PLL_CTUNE_CNT6)

Address: 4005_C000h base + 268h offset = 4005_C268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				CTUNE_COUNT_6											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_CTUNE_CNT6 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTUNE_ COUNT_6	CTUNE Count 6 This is the Ripple counter value used to set coarse tune band bit 6.

52.1.72 PLL Coarse Tune Counts 5 and 4 (XCVR_PLL_CTUNE_CNT5_4)

Address: 4005_C000h base + 26Ch offset = 4005_C26Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CTUNE_COUNT_5												0				CTUNE_COUNT_4											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_CTUNE_CNT5_4 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 CTUNE_ COUNT_5	CTUNE Count 5 This is the Ripple counter value used to set coarse tune band bit 5.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTUNE_ COUNT_4	CTUNE Count 4 This is the Ripple counter value used to set coarse tune band bit 4.

52.1.73 PLL Coarse Tune Counts 3 and 2 (XCVR_PLL_CTUNE_CNT3_2)

Address: 4005_C000h base + 270h offset = 4005_C270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CTUNE_COUNT_3												0				CTUNE_COUNT_2											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_CTUNE_CNT3_2 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 CTUNE_ COUNT_3	CTUNE Count 3 This is the Ripple counter value used to set coarse tune band bit 3.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTUNE_ COUNT_2	CTUNE Count 2 This is the Ripple counter value used to set coarse tune band bit 2.

52.1.74 PLL Coarse Tune Counts 1 and 0 (XCVR_PLL_CTUNE_CNT1_0)

Address: 4005_C000h base + 274h offset = 4005_C274h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CTUNE_COUNT_1												0				CTUNE_COUNT_0											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_CTUNE_CNT1_0 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 CTUNE_ COUNT_1	CTUNE Count 1 This is the Ripple counter value used to set coarse tune band bit 1.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_PLL_CTUNE_CNT1_0 field descriptions (continued)

Field	Description
CTUNE_ COUNT_0	CTUNE Count 0 This is the Ripple counter value used to set coarse tune band bit 0.

52.1.75 PLL Coarse Tune Results (XCVR_PLL_CTUNE_RESULTS)

Address: 4005_C000h base + 278h offset = 4005_C278h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				CTUNE_FREQ_TARGET											
W																
Reset	0	0	0	0	1	0	0	1	0	1	1	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTUNE_BEST_DIFF								0	CTUNE_SELECTED						
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

XCVR_PLL_CTUNE_RESULTS field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 CTUNE_FREQ_ TARGET	Coarse Tune Frequency Target This is the Frequency Target in MHz that is currently being presented to the Coarse Tune Calibrator.
15–8 CTUNE_BEST_ DIFF	Coarse Tune Absolute Best Difference This is the absolute value of the best difference found during Coarse Tune between the targeted frequency count and the actual frequency count.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTUNE_ SELECTED	Coarse Tune Band to VCO This is the current VCO Coarse Tune setting, it is the result of the Coarse Tune Calibration, unless overridden using CTUNE_DIS.

52.1.76 Transceiver Control (XCVR_XCVR_CTRL)

Address: 4005_C000h base + 280h offset = 4005_C280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								REF_CLK_FREQ		TGT_PWR_SRC		0	PROTOCOL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_XCVR_CTRL field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–8
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>7–6
REF_CLK_FREQ</td><td>Radio Reference Clock Frequency
This register selects the Reference Clock Frequency for the Radio.
00 32 MHz
01 Reserved
10 Reserved
11 Reserved</td></tr>
<tr><td>5–4
TGT_PWR_SRC</td><td>Target Power Source
For determining transmit power, the TGT_PWR_SRC[1:0] bits control target power selection, according to the following table.
<table>
<tr><th>TGT_PWR_SRC[1:0]</th><th>TARGET POWER SOURCE</th></tr>
<tr><td>00</td><td>PA_POWER[3:0] register (XCVR space)</td></tr>
<tr><td>01</td><td>BTLE Link Layer</td></tr>
<tr><td>10</td><td>Zigbee Link Layer (PA_PWR[3:0] register in ZIGBEE space)</td></tr>
<tr><td>11</td><td>PROTOCOL[2:0] bits select target power source</td></tr>
</table></td></tr>
<tr><td>3
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>PROTOCOL</td><td>Radio Protocol Selection
This register selects the Radio Communication Protocol.
000 BLE
001 BLE in MBAN
010 BLE overlap MBAN
011 Reserved</td></tr>
</table>

Table continues on the next page...

XCVR_XCVR_CTRL field descriptions (continued)

Field	Description
	100 Zigbee 101 802.15.4j 110 128 Channel FSK 111 128 Channel GFSK

52.1.77 Transceiver Status (XCVR_XCVR_STATUS)

Address: 4005_C000h base + 284h offset = 4005_C284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												SOC_USING_RF_OSC_CLK	XTAL_READY	RIF_LL_ACTIVE	BTLE_SYSCLK_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TX_MODE	RX_MODE	PLL_SEQ_STATE				TSM_COUNT							
W																
Reset	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

XCVR_XCVR_STATUS field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 SOC_USING_ RF_OSC_CLK	SOC Using RF Clock Indication SoC signal from the CLKGEN that asserts high when the MCG is configured to use RF OSC clock as the SoC clock source
18 XTAL_READY	RF Osciallator Xtal Ready Oscillator warmup count complete. 0 Indicates that the RF Oscillator is disabled or has not completed its warmup. 1 Indicates that the RF Oscillator has completed its warmup count and is ready for use.
17 RIF_LL_ACTIVE	Link Layer Active Indication Reflects the state of the BTLE “Link Layer Active” status bit. RIF_LL_ACTIVE is to be used by the host as an 'early' indication to prevent host to do any operations while BTLE IP is doing transceiver operations, so as to reduce the peak power and noise.

Table continues on the next page...

XCVR_XCVR_STATUS field descriptions (continued)

Field	Description
16 BTLE_SYSCLK_REQ	BTLE System Clock Request Reflects the state of the BTLE oscillator request signal. BTLE_SYSCLK_REQ is the BTLE control for the RF Oscillator. BTLE will deassert this signal upon entering DSM (deep sleep mode) to request oscillator turn-off, and will re-assert it prior to exiting DSM. The turn-on leadtime on this signal for exiting DSM, is programmable with the BTLE block. This read-only bit can thus be queried to ascertain the power-state of BTLE.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 TX_MODE	Transmit Mode Indicates an TX transceiver operation is in progress.
12 RX_MODE	Receive Mode Indicates an RX transceiver operation is in progress.
11–8 PLL_SEQ_STATE	PLL Sequence State Reflects the state of the PLL digital state machine. 0 PLL OFF 2 CTUNE 3 CTUNE_SETTLE 6 HPMCAL1 8 HPMCAL1_SETTLE 10 HPMCAL2 12 HPMCAL2_SETTLE 15 PLLREADY
TSM_COUNT	TSM Count Reflects the instantaneous value of the TSM counter.

52.1.78 Soft Reset (XCVR_SOFT_RESET)

Transceiver Soft Reset **Note:** Register not currently implemented.

Address: 4005_C000h base + 288h offset = 4005_C288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_SOFT_RESET field descriptions

Field	Description
Reserved	This field is reserved.

XCVR_SOFT_RESET field descriptions (continued)

Field	Description
	This read-only field is reserved and always has the value 0.

52.1.78 Overwrite Version (XCVR_OVERWRITE_VER)

The Overwrite Version allows software to store a version number of trim and calibration values which are used to overwrite the chip default values in the registers. Typically, software would perform the overwrite of the defaults in transceiver registers and then write the version number from the file containing the overwrite values into this register.

NOTE

This register has no hardware connections, it is simply a designated storage location for a version number.

Address: 4005_C000h base + 290h offset = 4005_C290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								OVERWRITE_VER							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_OVERWRITE_VER field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OVERWRITE_VER	Overwrite Version Number. Points to the version number of the overwrites.h file used to initialize the device; can be used by software to identify a version-controlled set of non-default values to be written into the transceiver's register map.

52.1.79 DMA Control (XCVR_DMA_CTRL)

Transceiver DMA per-channel enable register. Transceiver DMA is intended for engineering evaluation only. If DMA_I_EN and DMA_Q_EN are both set, I channel samples appear in register field DMA_DATA[DMA_DATA_11_0] (lower halfword), and Q channel samples appears in DMA_DATA[DMA_DATA_27_16] (upper halfword).

Address: 4005_C000h base + 294h offset = 4005_C294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DMA_Q_EN	DMA_I_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DMA_CTRL field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DMA_Q_EN	DMA Q Enable DMA Q Enable 0 Transceiver Q channel DMA disabled. 1 Enable the transceiver DMA engine to store RX_DIG Q channel outputs to system memory.
0 DMA_I_EN	DMA I Enable DMA I Enable 0 Transceiver I channel DMA disabled. 1 Enable the transceiver DMA engine to store RX_DIG I channel outputs to system memory.

52.1.80 DMA Data (XCVR_DMA_DATA)

Transceiver DMA Data. Not intended to be read directly by application software, this register is merely an address slot where the SoC DMA controller can access the RX_DIG samples to be transferred to memory. After enabling transceiver DMA, the first sample pair should be discarded due to DMA controller latency; all subsequent samples are valid

Address: 4005_C000h base + 298h offset = 4005_C298h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DMA_DATA_27_16												0				DMA_DATA_11_0											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DMA_DATA field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DMA_DATA_27_16	DMA_DATA_27_16 Reflection of the RX_DIG DMA data to be stored to system memory. The transceiver DMA engine acquires 12-bit samples from RX_DIG, and deposits them in this register field. If DMA Q channel is enabled, i.e. DMA_CTRL[DMA_Q_EN]=1, then this register field contains Q channel samples; otherwise it contains I channel samples. For single-channel mode, this register field represents the second consecutive RX_DIG sample of the sample-pair; for dual-channel mode, I and Q samples are captured simultaneously.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DMA_DATA_11_0	DMA_DATA_11_0 Reflection of the RX_DIG DMA data to be stored to system memory. The transceiver DMA engine acquires 12-bit samples from RX_DIG, and deposits them in this register field. If DMA I channel is enabled, i.e. DMA_CTRL[DMA_I_EN]=1, then this register field contains I channel samples; otherwise it contains Q channel samples. For single-channel mode, this register field represents the first consecutive RX_DIG sample of the sample-pair; for dual-channel mode, I and Q samples are captured simultaneously.

52.1.81 Digital Test Control (XCVR_DTEST_CTRL)

Digital Test Control. Allows selection and enablement of a page of DTEST signals to appear on the SoC DTEST pins.

NOTE

This register configures only the transceiver for DTEST mode; since DTEST pads on the SoC are multiplexed with other functions, SoC Port Pin programming is also required for each DTEST output

Address: 4005_C000h base + 29Ch offset = 4005_C29Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		RAW_MODE_Q	RAW_MODE_I	0	DTEST_SHFT			0						TSM_GPIO_OVLAY_1	TSM_GPIO_OVLAY_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO1_OVLAY_PIN				GPIO0_OVLAY_PIN				DTEST_EN	0	DTEST_PAGE					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_DTEST_CTRL field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 RAW_MODE_Q	DTEST Raw Mode Enable for Q Channel DTEST Q Channel Raw Mode allows raw, unfiltered ADC samples to be brought out to DTEST pins on the RXDIGIQ DTEST page. In raw mode, 2 5-bit ADC samples are concatenated into a single 12-bit DMA sample. DMA transfers these samples to memory in the same way it transfers filtered samples. Raw mode is only supported when the RX_DIG is programmed to decimate-by-2. The procedure to active Raw mode on Q channel is as follows: 1. RX_DIG_CTRL[RX_ADC_RAW_EN]=1 2. RX_DIG_CTRL[RX_DEC_FILT_OSR]=0 3. DTEST_CTRL[DTEST_PAGE]=0x0E (RXDIGIQ) 4. DTEST_CTRL[DTEST_EN]=1 5. DTEST_CTRL[RAW_MODE_Q]=1
28 RAW_MODE_I	DTEST Raw Mode Enable for I Channel DTEST I Channel Raw Mode allows raw, unfiltered ADC samples to be brought out to DTEST pins on the RXDIGIQ DTEST page. In raw mode, 2 5-bit ADC samples are concatenated into a single 12-bit DMA sample. DMA transfers these samples to memory in the same way it transfers filtered samples. Raw mode is only supported when the RX_DIG is programmed to decimate-by-2. The procedure to active Raw mode on I channel is as follows: 1. RX_DIG_CTRL[RX_ADC_RAW_EN]=1 2. RX_DIG_CTRL[RX_DEC_FILT_OSR]=0 3. DTEST_CTRL[DTEST_PAGE]=0x0E (RXDIGIQ) 4. DTEST_CTRL[DTEST_EN]=1 5. DTEST_CTRL[RAW_MODE_I]=1
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 DTEST_SHFT	DTEST Shift Control This register field DTEST_SHFT[1:0] control the amount of "arithmetic shift", which can optionally be applied to DTEST output busses. DTEST_SHFT affects only 2 DTEST output busses:

Table continues on the next page...

XCVR_DTEST_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>PLL_RIPPLE_COUNTER[16:0] on DTEST page: PLLRIPPLE (0x02). Shift is to the left (magnitude increasing)
RX_DIG_IQ[11:0] on DTEST page: RXDIGIQ (0x0E). Shift is to the right (magnitude decreasing)

The bits of PLL_RIPPLE_COUNTER[16:0], an unsigned value, are shifted by DTEST_SHFT[1:0] according to the following table
<table>
<tr><th>DTEST_SHFT[1:0]</th><th>THESE BITS APPEAR ON DTEST[13:0]</th></tr>
<tr><td>00</td><td>PLL_RIPPLE_COUNTER[16:3]</td></tr>
<tr><td>01</td><td>PLL_RIPPLE_COUNTER[15:2]</td></tr>
<tr><td>10</td><td>PLL_RIPPLE_COUNTER[14:1]</td></tr>
<tr><td>11</td><td>PLL_RIPPLE_COUNTER[13:0]</td></tr>
</table>

The bits of RX_DIG_IQ[11:0], a signed value, are shifted/sign-extended by DTEST_SHFT[1:0] according to the following table
<table>
<tr><th>DTEST_SHFT[1:0]</th><th>THESE BITS APPEAR ON DTEST[13:2]</th></tr>
<tr><td>00</td><td>RX_DIG_IQ[11:0] (no shift)</td></tr>
<tr><td>01</td><td>RX_DIG_IQ[11],RX_DIG_IQ[11:1] (right shift by 1)</td></tr>
<tr><td>10</td><td>RX_DIG_IQ[11],RX_DIG_IQ[11],RX_DIG_IQ[11:2] (right shift by 2)</td></tr>
<tr><td>11</td><td>RX_DIG_IQ[11:0] (no shift)</td></tr>
</table>
</td></tr>
<tr><td>23–18
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>17
TSM_GPIO_
OVLAY_1</td><td>TSM GPIO 1 Overlay Pin

The TSM-controlled output GPIO1_TRIG_EN can be routed to any DTEST pin, regardless of page selection (DTEST_PAGE), replacing the nominal page-selected output for that pin. When TSM_GPIO_OVLAY_1 = 1, the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear, any of DTEST0 - DTEST13. When TSM_GPIO_OVLAY_1 = 0, there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin.

0 there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin.
1 the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear.</td></tr>
<tr><td>16
TSM_GPIO_
OVLAY_0</td><td>TSM GPIO 0 Overlay Pin

The TSM-controlled output GPIO0_TRIG_EN can be routed to any DTEST pin, regardless of page selection (DTEST_PAGE), replacing the nominal page-selected output for that pin. When TSM_GPIO_OVLAY_0 = 1, the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear, any of DTEST0 - DTEST13. When TSM_GPIO_OVLAY_0 = 0, there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin.

0 there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin.
1 the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear.</td></tr>
<tr><td>15–12
GPIO1_OVLAY_
PIN</td><td>GPIO 1 Overlay Pin</td></tr>
</table>

Table continues on the next page...

XCVR_DTEST_CTRL field descriptions (continued)

Field	Description
	The TSM-controlled output GPIO1_TRIG_EN can be routed to any DTEST pin, regardless of page selection (DTEST_PAGE), replacing the nominal page-selected output for that pin. When TSM_GPIO_OVLAY_1 = 1, this register selects the DTEST pin onto which GPIO1_TRIG_EN will appear, any of DTEST0 - DTEST13. When TSM_GPIO_OVLAY_1 = 0, this register is ignored, and the DTEST Page Table dictates the node that appears on each DTEST pin.
11–8 GPIO0_OVLAY_PIN	GPIO 0 Overlay Pin The TSM-controlled output GPIO0_TRIG_EN can be routed to any DTEST pin, regardless of page selection (DTEST_PAGE), replacing the nominal page-selected output for that pin. When TSM_GPIO_OVLAY_0 = 1, this register selects the DTEST pin onto which GPIO0_TRIG_EN will appear, any of DTEST0 - DTEST13. When TSM_GPIO_OVLAY_0 = 0, this register is ignored, and the DTEST Page Table dictates the node that appears on each DTEST pin.
7 DTEST_EN	DTEST Enable DTEST enable 0 Disables DTEST. The IC's DTEST pins assume their mission function. 1 Enables DTEST. The contents of the selected page (DTEST_PAGE) will appear on the IC's DTEST output pins.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DTEST_PAGE	DTEST Page Selector DTEST Page signal assignments are defined in a spreadsheet located at http://compass.freescale.net/livelink/livelink/open/231923338.

52.1.82 Packet Buffer Control Register (XCVR_PB_CTRL)

Address: 4005_C000h base + 2A0h offset = 4005_C2A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															PB_PROTECT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PB_CTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 PB_PROTECT	PB Protect Protect Packet Buffer contents against overwriting by the next received packet 0 Incoming received packets overwrite Packet Buffer contents (default) 1 Incoming received packets are blocked from overwriting Packet Buffer contents

52.1.83 Transceiver Sequence Manager Control (XCVR_TSM_CTRL)

Address: 4005_C000h base + 2C0h offset = 4005_C2C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BKPT								0			ABORT_ON_FREQ_TARG	ABORT_ON_CYCLE_SLIP	ABORT_ON_CTUNE	RX_ABORT_DIS	TX_ABORT_DIS
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									DATA_PADDING_EN	PA_RAMP_SEL		FORCE_RX_EN	FORCE_TX_EN	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TSM_CTRL field descriptions

Field	Description
31–24 BKPT	TSM Breakpoint Temporarily halt a TSM sequence during the warmup or warmdown phase. When the TSM counter matches the value of BKPT[7:0], breakpoint will take effect and the TSM counter will stop and hold its count. Breakpoint will remain in effect as long as BKPT[7:0] matches the TSM counter value. The TSM Breakpoint can be lifted by modifying the contents of this register. The default value of this register, 0xFF, is greater than the length of the longest possible sequence, so a breakpoint will never be triggered unless BKPT[7:0] is programmed to a value less than the length of sequence.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_TSM_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>20
ABORT_ON_
FREQ_TARG</td><td>Abort On Frequency Target Lock Detect Failure
0 don't allow TSM abort on Frequency Target Unlock Detect
1 allow TSM abort on Frequency Target Unlock Detect</td></tr>
<tr><td>19
ABORT_ON_
CYCLE_SLIP</td><td>Abort On Cycle Slip Lock Detect Failure
0 don't allow TSM abort on Cycle Slip Unlock Detect
1 allow TSM abort on Cycle Slip Unlock Detect</td></tr>
<tr><td>18
ABORT_ON_
CTUNE</td><td>Abort On Coarse Tune Lock Detect Failure
0 don't allow TSM abort on Coarse Tune Unlock Detect
1 allow TSM abort on Coarse Tune Unlock Detect</td></tr>
<tr><td>17
RX_ABORT_DIS</td><td>Receive Abort Disable
RX Abort disable. When set, prevents PLL unlock events during RX sequences from aborting the sequence.</td></tr>
<tr><td>16
TX_ABORT_DIS</td><td>Transmit Abort Disable
TX Abort disable. When set, prevents PLL unlock events during TX sequences from aborting the sequence.</td></tr>
<tr><td>15–7
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>6
DATA_
PADDING_EN</td><td>Data Padding Enable
Enables TX Data Padding. Data padding works in conjunction with PA ramping to minimize spectral transients during PA turn-on and turn-off. The nature of the data padding depends on the setting of XCVR_CTRL[PROTOCOL].
0 Disable TX Data Padding
1 Enable TX Data Padding</td></tr>
<tr><td>5–4
PA_RAMP_SEL</td><td>PA Ramp Selection
Selects the ramp-rate, and thus the duration, for PA ramping. Ramp-rate is the rate at which the PA ramping logic steps through the PA Bias Table.
<b>NOTE:</b> the default TSM TX sequence needs to be adjusted (re-programmed) for a 4us or 8us ramp.
<table>
<tr><th>PA_RAMP_SEL[1:0]</th><th>TOTAL RAMP DURATION</th><th>DURATION OF EACH RAMP STEP</th></tr>
<tr><td>00</td><td>No ramp</td><td>No ramp</td></tr>
<tr><td>01</td><td>2us</td><td>0.25us</td></tr>
<tr><td>10</td><td>4us</td><td>0.5us</td></tr>
<tr><td>11</td><td>8us</td><td>1us</td></tr>
</table></td></tr>
<tr><td>3
FORCE_RX_EN</td><td>Force Receive Enable
Direct software control to launch a RX TSM sequence. Initiates RX Warmup on a 0 to 1 transition and RX Warmdown on a 1 to 0 transition.</td></tr>
</table>

Table continues on the next page...

XCVR_TSM_CTRL field descriptions (continued)

Field	Description
	0 TSM Idle 1 TSM executes a RX sequence
2 FORCE_TX_EN	Force Transmit Enable Direct software control to launch a TX TSM sequence. Initiates a TX Warmup sequence on a 0 to 1 transition and a TX Warmdown sequence on a 1 to 0 transition. 0 TSM Idle 1 TSM executes a TX sequence
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.1.84 End of Sequence Control (XCVR_END_OF_SEQ)

Address: 4005_C000h base + 2C4h offset = 4005_C2C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	END_OF_RX_WD								END_OF_RX_WU								END_OF_TX_WD								END_OF_TX_WU							
Reset	0	1	1	0	0	1	0	1	0	1	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	1	1	1

XCVR_END_OF_SEQ field descriptions

Field	Description
31–24 END_OF_RX_ WD	End of RX Warmdown This register defines the point at which the TSM RX sequence warmdown completes, and the TSM returns to idle. The duration of the TSM warmdown phase is determined by: END_OF_RX_WD – END_OF_RX_WU. For example: the sequence register defaults render the END_OF_RX_WD=0x33 and END_OF_RX_WU=0x32, so the duration of the warmdown phase is 0x33 – 0x32 = 1 microseconds.
23–16 END_OF_RX_ WU	End of RX Warmup This register defines the length of the TSM RX warmup sequence. After the assertion of a RX sequence-initiating event, when the TSM counter reaches the count matching this register, it will stop and hold its count, and the TSM will transition from the WARMUP to the ON phase.
15–8 END_OF_TX_ WD	End of TX Warmdown This register defines the point at which the TSM TX sequence warmdown completes, and the TSM returns to idle. The duration of the TSM warmdown phase is determined by: END_OF_TX_WD – END_OF_TX_WU. For example: the sequence register defaults render the END_OF_TX_WD=0x69 and END_OF_TX_WU=0x66, so the duration of the warmdown phase is 0x69 – 0x66 = 3 microseconds.
END_OF_TX_ WU	End of TX Warmup This register defines the length of the TSM TX warmup sequence. After the assertion of a TX sequence-initiating event, when the TSM counter reaches the count matching this register, it will stop and hold its count, and the TSM will transition from the WARMUP to the ON phase.

52.1.85 TSM Override 0 (XCVR_TSM_OVRD0)

Address: 4005_C000h base + 2C8h offset = 4005_C2C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL_RX_LDV_RIPPLE_MUX_EN_OVRD	PLL_RX_LDV_RIPPLE_MUX_EN_OVRD_EN	PLL_LDV_EN_OVRD	PLL_LDV_EN_OVRD_EN	PLL_PA_BUF_EN_OVRD	PLL_PA_BUF_EN_OVRD_EN	PLL_VCO_BUF_TX_EN_OVRD	PLL_VCO_BUF_TX_EN_OVRD_EN	PLL_VCO_BUF_RX_EN_OVRD	PLL_VCO_BUF_RX_EN_OVRD_EN	PLL_VCO_EN_OVRD	PLL_VCO_EN_OVRD_EN	PLL_CYCLE_SLIP_LD_EN_OVRD	PLL_CYCLE_SLIP_LD_EN_OVRD_EN	PLL_VCO_AUTOTUNE_EN_OVRD	PLL_VCO_AUTOTUNE_EN_OVRD_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XTAL_ADC_REF_CLK_EN_OVRD	XTAL_ADC_REF_CLK_EN_OVRD_EN	XTAL_PLL_REF_CLK_EN_OVRD	XTAL_PLL_REF_CLK_EN_OVRD_EN	ADC_DIG_REG_EN_OVRD	ADC_DIG_REG_EN_OVRD_EN	ADC_ANA_REG_EN_OVRD	ADC_ANA_REG_EN_OVRD_EN	TCA_TX_REG_EN_OVRD	TCA_TX_REG_EN_OVRD_EN	QGEN_REG_EN_OVRD	QGEN_REG_EN_OVRD_EN	PLL_VCO_REG_EN_OVRD	PLL_VCO_REG_EN_OVRD_EN	PLL_REG_EN_OVRD	PLL_REG_EN_OVRD_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TSM_OVRD0 field descriptions

Field	Description
31 PLL_RX_LDV_RIPPLE_MUX_EN_OVRD	Override value for PLL_RX_LDV_RIPPLE_MUX_EN When PLL_RX_LDV_RIPPLE_MUX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_rx_ldv_ripple_mux_en". This bit is ignored when PLL_RX_LDV_RIPPLE_MUX_EN_OVRD_EN==0.
30 PLL_RX_LDV_RIPPLE_MUX_EN_OVRD_EN	Override control for PLL_RX_LDV_RIPPLE_MUX_EN 0 Normal operation. 1 Use the state of PLL_RX_LDV_RIPPLE_MUX_EN_OVRD to override the signal "pll_rx_ldv_ripple_mux_en".
29 PLL_LDV_EN_OVRD	Override value for PLL_LDV_EN When PLL_LDV_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_ldv_en". This bit is ignored when PLL_LDV_EN_OVRD_EN==0.
28 PLL_LDV_EN_OVRD_EN	Override control for PLL_LDV_EN 0 Normal operation. 1 Use the state of PLL_LDV_EN_OVRD to override the signal "pll_ldv_en".

Table continues on the next page...

XCVR_TSM_OVRD0 field descriptions (continued)

Field	Description
27 PLL_PA_BUF_ EN_OVRD	Override value for PLL_PA_BUF_EN When PLL_PA_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_pa_buf_en". This bit is ignored when PLL_PA_BUF_EN_OVRD_EN==0.
26 PLL_PA_BUF_ EN_OVRD_EN	Override control for PLL_PA_BUF_EN 0 Normal operation. 1 Use the state of PLL_PA_BUF_EN_OVRD to override the signal "pll_pa_buf_en".
25 PLL_VCO_BUF_ TX_EN_OVRD	Override value for PLL_VCO_BUF_TX_EN When PLL_VCO_BUF_TX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_vco_buf_tx_en". This bit is ignored when PLL_VCO_BUF_TX_EN_OVRD_EN==0.
24 PLL_VCO_BUF_ TX_EN_OVRD_ EN	Override control for PLL_VCO_BUF_TX_EN 0 Normal operation. 1 Use the state of PLL_VCO_BUF_TX_EN_OVRD to override the signal "pll_vco_buf_tx_en".
23 PLL_VCO_BUF_ RX_EN_OVRD	Override value for PLL_VCO_BUF_RX_EN When PLL_VCO_BUF_RX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_vco_buf_rx_en". This bit is ignored when PLL_VCO_BUF_RX_EN_OVRD_EN==0.
22 PLL_VCO_BUF_ RX_EN_OVRD_ EN	Override control for PLL_VCO_BUF_RX_EN 0 Normal operation. 1 Use the state of PLL_VCO_BUF_RX_EN_OVRD to override the signal "pll_vco_buf_rx_en".
21 PLL_VCO_EN_ OVRD	Override value for PLL_VCO_EN When PLL_VCO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_vco_en". This bit is ignored when PLL_VCO_EN_OVRD_EN==0.
20 PLL_VCO_EN_ OVRD_EN	Override control for PLL_VCO_EN 0 Normal operation. 1 Use the state of PLL_VCO_EN_OVRD to override the signal "pll_vco_en".
19 PLL_CYCLE_ SLIP_LD_EN_ OVRD	Override value for PLL_CYCLE_SLIP_LD_EN When PLL_CYCLE_SLIP_LD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_cycle_slip_ld_en". This bit is ignored when PLL_CYCLE_SLIP_LD_EN_OVRD_EN==0.
18 PLL_CYCLE_ SLIP_LD_EN_ OVRD_EN	Override control for PLL_CYCLE_SLIP_LD_EN 0 Normal operation. 1 Use the state of PLL_CYCLE_SLIP_LD_EN_OVRD to override the signal "pll_cycle_slip_ld_en".
17 PLL_VCO_ AUTOTUNE_ EN_OVRD	Override value for PLL_VCO_AUTOTUNE_EN When PLL_VCO_AUTOTUNE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_vco_autotune_en". This bit is ignored when PLL_VCO_AUTOTUNE_EN_OVRD_EN==0.
16 PLL_VCO_ AUTOTUNE_ EN_OVRD_EN	Override control for PLL_VCO_AUTOTUNE_EN 0 Normal operation. 1 Use the state of PLL_VCO_AUTOTUNE_EN_OVRD to override the signal "pll_vco_autotune_en".

Table continues on the next page...

XCVR_TSM_OVRD0 field descriptions (continued)

Field	Description
15 XTAL_ADC_ REF_CLK_EN_ OVRD	Override value for XTAL_ADC_REF_CLK_EN When XTAL_ADC_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "xtal_adc_ref_clk_en". This bit is ignored when XTAL_ADC_REF_CLK_EN_OVRD_EN==0.
14 XTAL_ADC_ REF_CLK_EN_ OVRD_EN	Override control for XTAL_ADC_REF_CLK_EN 0 Normal operation. 1 Use the state of XTAL_ADC_REF_CLK_EN_OVRD to override the signal "xtal_adc_ref_clk_en".
13 XTAL_PLL_REF_ CLK_EN_OVRD	Override value for XTAL_PLL_REF_CLK_EN When XTAL_PLL_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "xtal_pll_ref_clk_en". This bit is ignored when XTAL_PLL_REF_CLK_EN_OVRD_EN==0.
12 XTAL_PLL_REF_ CLK_EN_OVRD_ EN	Override control for XTAL_PLL_REF_CLK_EN 0 Normal operation. 1 Use the state of XTAL_PLL_REF_CLK_EN_OVRD to override the signal "xtal_pll_ref_clk_en".
11 ADC_DIG_REG_ EN_OVRD	Override value for ADC_DIG_REG_EN When ADC_DIG_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_dig_reg_en". This bit is ignored when ADC_DIG_REG_EN_OVRD_EN==0.
10 ADC_DIG_REG_ EN_OVRD_EN	Override control for ADC_DIG_REG_EN 0 Normal operation. 1 Use the state of ADC_DIG_REG_EN_OVRD to override the signal "adc_dig_reg_en".
9 ADC_ANA_ REG_EN_OVRD	Override value for ADC_ANA_REG_EN When ADC_ANA_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_ana_reg_en". This bit is ignored when ADC_ANA_REG_EN_OVRD_EN==0.
8 ADC_ANA_ REG_EN_ OVRD_EN	Override control for ADC_ANA_REG_EN 0 Normal operation. 1 Use the state of ADC_ANA_REG_EN_OVRD to override the signal "adc_ana_reg_en".
7 TCA_TX_REG_ EN_OVRD	Override value for TCA_TX_REG_EN When TCA_TX_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tca_tx_reg_en". This bit is ignored when TCA_TX_REG_EN_OVRD_EN==0.
6 TCA_TX_REG_ EN_OVRD_EN	Override control for TCA_TX_REG_EN 0 Normal operation. 1 Use the state of TCA_TX_REG_EN_OVRD to override the signal "tca_tx_reg_en".
5 QGEN_REG_ EN_OVRD	Override value for QGEN_REG_EN When QGEN_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "qgen_reg_en". This bit is ignored when QGEN_REG_EN_OVRD_EN==0.
4 QGEN_REG_ EN_OVRD_EN	Override control for QGEN_REG_EN 0 Normal operation. 1 Use the state of QGEN_REG_EN_OVRD to override the signal "qgen_reg_en".

Table continues on the next page...

XCVR_TSM_OVRD0 field descriptions (continued)

Field	Description
3 PLL_VCO_REG_EN_OVRD	Override value for PLL_VCO_REG_EN When PLL_VCO_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_vco_reg_en". This bit is ignored when PLL_VCO_REG_EN_OVRD_EN==0.
2 PLL_VCO_REG_EN_OVRD_EN	Override control for PLL_VCO_REG_EN 0 Normal operation. 1 Use the state of PLL_VCO_REG_EN_OVRD to override the signal "pll_vco_reg_en".
1 PLL_REG_EN_OVRD	Override value for PLL_REG_EN When PLL_REG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_reg_en". This bit is ignored when PLL_REG_EN_OVRD_EN==0.
0 PLL_REG_EN_OVRD_EN	Override control for PLL_REG_EN 0 Normal operation. 1 Use the state of PLL_REG_EN_OVRD to override the signal "pll_reg_en".

52.1.86 TSM Override 1 (XCVR_TSM_OVRD1)

Address: 4005_C000h base + 2CCh offset = 4005_C2CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BBF_PDET_EN_OVRD	BBF_PDET_EN_OVRD_EN	BBF_Q_EN_OVRD	BBF_Q_EN_OVRD_EN	BBF_I_EN_OVRD	BBF_I_EN_OVRD_EN	ADC_RST_EN_OVRD	ADC_RST_EN_OVRD_EN	ADC_DAC2_EN_OVRD	ADC_DAC2_EN_OVRD_EN	ADC_DAC1_EN_OVRD	ADC_DAC1_EN_OVRD_EN	ADC_Q_ADC_EN_OVRD	ADC_Q_ADC_EN_OVRD_EN	ADC_I_ADC_EN_OVRD	ADC_I_ADC_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADC_CLK_EN_OVRD	ADC_CLK_EN_OVRD_EN	ADC_BIAS_EN_OVRD	ADC_BIAS_EN_OVRD_EN	ADC_EN_OVRD	ADC_EN_OVRD_EN	TX_EN_OVRD	TX_EN_OVRD_EN	QGEN25_EN_OVRD	QGEN25_EN_OVRD_EN	PLL_PHDET_EN_OVRD	PLL_PHDET_EN_OVRD_EN	PLL_FILTER_CHARGE_EN_OVRD	PLL_FILTER_CHARGE_EN_OVRD_EN	PLL_TX_LDV_RIPPLE_MUX_EN_OVRD	PLL_TX_LDV_RIPPLE_MUX_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TSM_OVRD1 field descriptions

Field	Description
31 BBF_PDET_EN_ OVRD	Override value for BBF_PDET_EN When BBF_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bbf_pdet_en". This bit is ignored when BBF_PDET_EN_OVRD_EN==0.
30 BBF_PDET_EN_ OVRD_EN	Override control for BBF_PDET_EN 0 Normal operation. 1 Use the state of BBF_PDET_EN_OVRD to override the signal "bbf_pdet_en".
29 BBF_Q_EN_ OVRD	Override value for BBF_Q_EN When BBF_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bbf_q_en". This bit is ignored when BBF_Q_EN_OVRD_EN==0.
28 BBF_Q_EN_ OVRD_EN	Override control for BBF_Q_EN 0 Normal operation. 1 Use the state of BBF_Q_EN_OVRD to override the signal "bbf_q_en".
27 BBF_I_EN_ OVRD	Override value for BBF_I_EN When BBF_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bbf_i_en". This bit is ignored when BBF_I_EN_OVRD_EN==0.
26 BBF_I_EN_ OVRD_EN	Override control for BBF_I_EN 0 Normal operation. 1 Use the state of BBF_I_EN_OVRD to override the signal "bbf_i_en".
25 ADC_RST_EN_ OVRD	Override value for ADC_RST_EN When ADC_RST_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_rst_en". This bit is ignored when ADC_RST_EN_OVRD_EN==0.
24 ADC_RST_EN_ OVRD_EN	Override control for ADC_RST_EN 0 Normal operation. 1 Use the state of ADC_RST_EN_OVRD to override the signal "adc_rst_en".
23 ADC_DAC2_EN_ OVRD	Override value for ADC_DAC2_EN When ADC_DAC2_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_dac2_en". This bit is ignored when ADC_DAC2_EN_OVRD_EN==0.
22 ADC_DAC2_EN_ OVRD_EN	Override control for ADC_DAC2_EN 0 Normal operation. 1 Use the state of ADC_DAC2_EN_OVRD to override the signal "adc_dac2_en".
21 ADC_DAC1_EN_ OVRD	Override value for ADC_DAC1_EN When ADC_DAC1_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_dac1_en". This bit is ignored when ADC_DAC1_EN_OVRD_EN==0.
20 ADC_DAC1_EN_ OVRD_EN	Override control for ADC_DAC1_EN 0 Normal operation. 1 Use the state of ADC_DAC1_EN_OVRD to override the signal "adc_dac1_en".

Table continues on the next page...

XCVR_TSM_OVRD1 field descriptions (continued)

Field	Description
19 ADC_Q_ADC_ EN_OVRD	Override value for ADC_Q_ADC_EN When ADC_Q_ADC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_q_adc_en". This bit is ignored when ADC_Q_ADC_EN_OVRD_EN==0.
18 ADC_Q_ADC_ EN_OVRD_EN	Override control for ADC_Q_ADC_EN 0 Normal operation. 1 Use the state of ADC_Q_ADC_EN_OVRD to override the signal "adc_q_adc_en".
17 ADC_I_ADC_ EN_OVRD	Override value for ADC_I_ADC_EN When ADC_I_ADC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_i_adc_en". This bit is ignored when ADC_I_ADC_EN_OVRD_EN==0.
16 ADC_I_ADC_ EN_OVRD_EN	Override control for ADC_I_ADC_EN 0 Normal operation. 1 Use the state of ADC_I_ADC_EN_OVRD to override the signal "adc_i_adc_en".
15 ADC_CLK_EN_ OVRD	Override value for ADC_CLK_EN When ADC_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_clk_en". This bit is ignored when ADC_CLK_EN_OVRD_EN==0.
14 ADC_CLK_EN_ OVRD_EN	Override control for ADC_CLK_EN 0 Normal operation. 1 Use the state of ADC_CLK_EN_OVRD to override the signal "adc_clk_en".
13 ADC_BIAS_EN_ OVRD	Override value for ADC_BIAS_EN When ADC_BIAS_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_bias_en". This bit is ignored when ADC_BIAS_EN_OVRD_EN==0.
12 ADC_BIAS_EN_ OVRD_EN	Override control for ADC_BIAS_EN 0 Normal operation. 1 Use the state of ADC_BIAS_EN_OVRD to override the signal "adc_bias_en".
11 ADC_EN_OVRD	Override value for ADC_EN When ADC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "adc_en". This bit is ignored when ADC_EN_OVRD_EN==0.
10 ADC_EN_ OVRD_EN	Override control for ADC_EN 0 Normal operation. 1 Use the state of ADC_EN_OVRD to override the signal "adc_en".
9 TX_EN_OVRD	Override value for TX_EN When TX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_en". This bit is ignored when TX_EN_OVRD_EN==0.
8 TX_EN_OVRD_ EN	Override control for TX_EN 0 Normal operation. 1 Use the state of TX_EN_OVRD to override the signal "tx_en".

Table continues on the next page...

XCVR_TSM_OVRD1 field descriptions (continued)

Field	Description
7 QGEN25_EN_ OVRD	Override value for QGEN25_EN When QGEN25_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "qgen25_en". This bit is ignored when QGEN25_EN_OVRD_EN==0.
6 QGEN25_EN_ OVRD_EN	Override control for QGEN25_EN 0 Normal operation. 1 Use the state of QGEN25_EN_OVRD to override the signal "qgen25_en".
5 PLL_PHDET_ EN_OVRD	Override value for PLL_PHDET_EN When PLL_PHDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_phdet_en". This bit is ignored when PLL_PHDET_EN_OVRD_EN==0.
4 PLL_PHDET_ EN_OVRD_EN	Override control for PLL_PHDET_EN 0 Normal operation. 1 Use the state of PLL_PHDET_EN_OVRD to override the signal "pll_phdet_en".
3 PLL_FILTER_ CHARGE_EN_ OVRD	Override value for PLL_FILTER_CHARGE_EN When PLL_FILTER_CHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_filter_charge_en". This bit is ignored when PLL_FILTER_CHARGE_EN_OVRD_EN==0.
2 PLL_FILTER_ CHARGE_EN_ OVRD_EN	Override control for PLL_FILTER_CHARGE_EN 0 Normal operation. 1 Use the state of PLL_FILTER_CHARGE_EN_OVRD to override the signal "pll_filter_charge_en".
1 PLL_TX_LDV_ RIPPLE_MUX_ EN_OVRD	Override value for PLL_TX_LDV_RIPPLE_MUX_EN When PLL_TX_LDV_RIPPLE_MUX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_tx_ldv_ripple_mux_en". This bit is ignored when PLL_TX_LDV_RIPPLE_MUX_EN_OVRD_EN==0.
0 PLL_TX_LDV_ RIPPLE_MUX_ EN_OVRD_EN	Override control for PLL_TX_LDV_RIPPLE_MUX_EN 0 Normal operation. 1 Use the state of PLL_TX_LDV_RIPPLE_MUX_EN_OVRD to override the signal "pll_tx_ldv_ripple_mux_en".

52.1.87 TSM Override 2 (XCVR_TSM_OVRD2)

Address: 4005_C000h base + 2D0h offset = 4005_C2D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SAR_ADC_TRIG_EN_OVRD	SAR_ADC_TRIG_EN_OVRD_EN	FREQ_TARG_LD_EN_OVRD	FREQ_TARG_LD_EN_OVRD_EN	DCOC_INIT_OVRD	DCOC_INIT_OVRD_EN	DCOC_EN_OVRD	DCOC_EN_OVRD_EN	ZBDEM_RX_EN_OVRD	ZBDEM_RX_EN_OVRD_EN	SIGMA_DELTA_EN_OVRD	SIGMA_DELTA_EN_OVRD_EN	RX_INIT_OVRD	RX_INIT_OVRD_EN	RX_DIG_EN_OVRD	RX_DIG_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TX_DIG_EN_OVRD	TX_DIG_EN_OVRD_EN	PLL_DIG_EN_OVRD	PLL_DIG_EN_OVRD_EN	TZA_DCOC_EN_OVRD	TZA_DCOC_EN_OVRD_EN	TZA_PDET_EN_OVRD	TZA_PDET_EN_OVRD_EN	TZA_Q_EN_OVRD	TZA_Q_EN_OVRD_EN	TZA_I_EN_OVRD	TZA_I_EN_OVRD_EN	TCA_EN_OVRD	TCA_EN_OVRD_EN	BBF_DCOC_EN_OVRD	BBF_DCOC_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TSM_OVRD2 field descriptions

Field	Description
31 SAR_ADC_TRIG_EN_OVRD	Override value for SAR_ADC_TRIG_EN When SAR_ADC_TRIG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sar_adc_trig_en". This bit is ignored when SAR_ADC_TRIG_EN_OVRD_EN==0.
30 SAR_ADC_TRIG_EN_OVRD_EN	Override control for SAR_ADC_TRIG_EN 0 Normal operation. 1 Use the state of SAR_ADC_TRIG_EN_OVRD to override the signal "sar_adc_trig_en".
29 FREQ_TARG_LD_EN_OVRD	Override value for FREQ_TARG_LD_EN When FREQ_TARG_LD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "freq_targ_ld_en". This bit is ignored when FREQ_TARG_LD_EN_OVRD_EN==0.
28 FREQ_TARG_LD_EN_OVRD_EN	Override control for FREQ_TARG_LD_EN 0 Normal operation. 1 Use the state of FREQ_TARG_LD_EN_OVRD to override the signal "freq_targ_ld_en".
27 DCOC_INIT_OVRD	Override value for DCOC_INIT When DCOC_INIT_OVRD_EN=1, this value overrides the mission mode state of the signal "dcoc_init". This bit is ignored when DCOC_INIT_OVRD_EN==0.
26 DCOC_INIT_OVRD_EN	Override control for DCOC_INIT

Table continues on the next page...

XCVR_TSM_OVRD2 field descriptions (continued)

Field	Description
	0 Normal operation. 1 Use the state of DCOC_INIT_OVRD to override the signal "dcoc_init".
25 DCOC_EN_ OVRD	Override value for DCOC_EN When DCOC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "dcoc_en". This bit is ignored when DCOC_EN_OVRD_EN==0.
24 DCOC_EN_ OVRD_EN	Override control for DCOC_EN 0 Normal operation. 1 Use the state of DCOC_EN_OVRD to override the signal "dcoc_en".
23 ZBDEM_RX_EN_ OVRD	Override value for ZBDEM_RX_EN When ZBDEM_RX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "zbdem_rx_en". This bit is ignored when ZBDEM_RX_EN_OVRD_EN==0.
22 ZBDEM_RX_EN_ OVRD_EN	Override control for ZBDEM_RX_EN 0 Normal operation. 1 Use the state of ZBDEM_RX_EN_OVRD to override the signal "zbdem_rx_en".
21 SIGMA_DELTA_ EN_OVRD	Override value for SIGMA_DELTA_EN When SIGMA_DELTA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sigma_delta_en". This bit is ignored when SIGMA_DELTA_EN_OVRD_EN==0.
20 SIGMA_DELTA_ EN_OVRD_EN	Override control for SIGMA_DELTA_EN 0 Normal operation. 1 Use the state of SIGMA_DELTA_EN_OVRD to override the signal "sigma_delta_en".
19 RX_INIT_OVRD	Override value for RX_INIT When RX_INIT_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_init". This bit is ignored when RX_INIT_OVRD_EN==0.
18 RX_INIT_OVRD_ EN	Override control for RX_INIT 0 Normal operation. 1 Use the state of RX_INIT_OVRD to override the signal "rx_init".
17 RX_DIG_EN_ OVRD	Override value for RX_DIG_EN When RX_DIG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_dig_en". This bit is ignored when RX_DIG_EN_OVRD_EN==0.
16 RX_DIG_EN_ OVRD_EN	Override control for RX_DIG_EN 0 Normal operation. 1 Use the state of RX_DIG_EN_OVRD to override the signal "rx_dig_en".
15 TX_DIG_EN_ OVRD	Override value for TX_DIG_EN When TX_DIG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_dig_en". This bit is ignored when TX_DIG_EN_OVRD_EN==0.
14 TX_DIG_EN_ OVRD_EN	Override control for TX_DIG_EN 0 Normal operation. 1 Use the state of TX_DIG_EN_OVRD to override the signal "tx_dig_en".

Table continues on the next page...

XCVR_TSM_OVRD2 field descriptions (continued)

Field	Description
13 PLL_DIG_EN_ OVRD	Override value for PLL_DIG_EN When PLL_DIG_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_dig_en". This bit is ignored when PLL_DIG_EN_OVRD_EN==0.
12 PLL_DIG_EN_ OVRD_EN	Override control for PLL_DIG_EN 0 Normal operation. 1 Use the state of PLL_DIG_EN_OVRD to override the signal "pll_dig_en".
11 TZA_DCOC_EN_ OVRD	Override value for TZA_DCOC_EN When TZA_DCOC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tza_dcoc_en". This bit is ignored when TZA_DCOC_EN_OVRD_EN==0.
10 TZA_DCOC_EN_ OVRD_EN	Override control for TZA_DCOC_EN 0 Normal operation. 1 Use the state of TZA_DCOC_EN_OVRD to override the signal "tza_dcoc_en".
9 TZA_PDET_EN_ OVRD	Override value for TZA_PDET_EN When TZA_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tza_pdet_en". This bit is ignored when TZA_PDET_EN_OVRD_EN==0.
8 TZA_PDET_EN_ OVRD_EN	Override control for TZA_PDET_EN 0 Normal operation. 1 Use the state of TZA_PDET_EN_OVRD to override the signal "tza_pdet_en".
7 TZA_Q_EN_ OVRD	Override value for TZA_Q_EN When TZA_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tza_q_en". This bit is ignored when TZA_Q_EN_OVRD_EN==0.
6 TZA_Q_EN_ OVRD_EN	Override control for TZA_Q_EN 0 Normal operation. 1 Use the state of TZA_Q_EN_OVRD to override the signal "tza_q_en".
5 TZA_I_EN_ OVRD	Override value for TZA_I_EN When TZA_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tza_i_en". This bit is ignored when TZA_I_EN_OVRD_EN==0.
4 TZA_I_EN_ OVRD_EN	Override control for TZA_I_EN 0 Normal operation. 1 Use the state of TZA_I_EN_OVRD to override the signal "tza_i_en".
3 TCA_EN_OVRD	Override value for TCA_EN When TCA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tca_en". This bit is ignored when TCA_EN_OVRD_EN==0.
2 TCA_EN_OVRD_ EN	Override control for TCA_EN 0 Normal operation. 1 Use the state of TCA_EN_OVRD to override the signal "tca_en".

Table continues on the next page...

XCVR_TSM_OVRD2 field descriptions (continued)

Field	Description
1 BBF_DCOC_EN_OVRD	Override value for BBF_DCOC_EN When BBF_DCOC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bbf_dcoc_en". This bit is ignored when BBF_DCOC_EN_OVRD_EN==0.
0 BBF_DCOC_EN_OVRD_EN	Override control for BBF_DCOC_EN 0 Normal operation. 1 Use the state of BBF_DCOC_EN_OVRD to override the signal "bbf_dcoc_en".

52.1.88 TSM Override 3 (XCVR_TSM_OVRD3)

Address: 4005_C000h base + 2D4h offset = 4005_C2D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				RX_MODE_OVRD	RX_MODE_OVRD_EN	TX_MODE_OVRD	TX_MODE_OVRD_EN	TSM_SPARE3_EN_OVRD	TSM_SPARE3_EN_OVRD_EN	TSM_SPARE2_EN_OVRD	TSM_SPARE2_EN_OVRD_EN	TSM_SPARE1_EN_OVRD	TSM_SPARE1_EN_OVRD_EN	TSM_SPARE0_EN_OVRD	TSM_SPARE0_EN_OVRD_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TSM_OVRD3 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 RX_MODE_OVRD	Override value for RX_MODE When RX_MODE_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_mode". This bit is ignored when RX_MODE_OVRD_EN==0.
10 RX_MODE_OVRD_EN	Override control for RX_MODE 0 Normal operation. 1 Use the state of RX_MODE_OVRD to override the signal "rx_mode".

Table continues on the next page...

XCVR_TSM_OVRD3 field descriptions (continued)

Field	Description
9 TX_MODE_ OVRD	Override value for TX_MODE When TX_MODE_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_mode". This bit is ignored when TX_MODE_OVRD_EN==0.
8 TX_MODE_ OVRD_EN	Override control for TX_MODE 0 Normal operation. 1 Use the state of TX_MODE_OVRD to override the signal "tx_mode".
7 TSM_SPARE3_ EN_OVRD	Override value for TSM_SPARE3_EN When TSM_SPARE3_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tsm_spare3_en". This bit is ignored when TSM_SPARE3_EN_OVRD_EN==0.
6 TSM_SPARE3_ EN_OVRD_EN	Override control for TSM_SPARE3_EN 0 Normal operation. 1 Use the state of TSM_SPARE3_EN_OVRD to override the signal "tsm_spare3_en".
5 TSM_SPARE2_ EN_OVRD	Override value for TSM_SPARE2_EN When TSM_SPARE2_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tsm_spare2_en". This bit is ignored when TSM_SPARE2_EN_OVRD_EN==0.
4 TSM_SPARE2_ EN_OVRD_EN	Override control for TSM_SPARE2_EN 0 Normal operation. 1 Use the state of TSM_SPARE2_EN_OVRD to override the signal "tsm_spare2_en".
3 TSM_SPARE1_ EN_OVRD	Override value for TSM_SPARE1_EN When TSM_SPARE1_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tsm_spare1_en". This bit is ignored when TSM_SPARE1_EN_OVRD_EN==0.
2 TSM_SPARE1_ EN_OVRD_EN	Override control for TSM_SPARE1_EN 0 Normal operation. 1 Use the state of TSM_SPARE1_EN_OVRD to override the signal "tsm_spare1_en".
1 TSM_SPARE0_ EN_OVRD	Override value for TSM_SPARE0_EN When TSM_SPARE0_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tsm_spare0_en". This bit is ignored when TSM_SPARE0_EN_OVRD_EN==0.
0 TSM_SPARE0_ EN_OVRD_EN	Override control for TSM_SPARE0_EN 0 Normal operation. 1 Use the state of TSM_SPARE0_EN_OVRD to override the signal "tsm_spare0_en".

52.1.89 PA Power (XCVR_PA_POWER)

This contents of this register are used as PA target power when XCVR_CTRL[TGT_PWR_SRC] = 00.

Address: 4005_C000h base + 2D8h offset = 4005_C2D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												PA_POWER			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PA_POWER field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PA_POWER	PA Power PA Target Power

52.1.90 PA Bias Table 0 (XCVR_PA_BIAS_TBL0)

Address: 4005_C000h base + 2DCh offset = 4005_C2DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PA_BIAS3				0				PA_BIAS2				0				PA_BIAS1				0				PA_BIAS0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PA_BIAS_TBL0 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 PA_BIAS3	PA_BIAS3 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the fourth ramp step. During PA ramp down, the contents of this register are the PA power value during the fourth-to-last ramp step. In both cases, PA_BIAS3 cannot exceed target power (enforced by PA ramping logic).
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 PA_BIAS2	PA_BIAS2 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the third ramp step. During PA ramp down, the contents of

Table continues on the next page...

XCVR_PA_BIAS_TBL0 field descriptions (continued)

Field	Description
	this register are the PA power value during the third-to-last ramp step. In both cases, PA_BIAS2 cannot exceed target power (enforced by PA ramping logic).
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 PA_BIAS1	PA_BIAS1 If PA ramping is enabled (TSM_CTRL[PA_RAMP_SEL] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the second ramp step. During PA ramp down, the contents of this register are the PA power value during the second-to-last ramp step. In both cases, PA_BIAS1 cannot exceed target power (enforced by PA ramping logic).
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PA_BIAS0	PA_BIAS0 If PA ramping is enabled (TSM_CTRL[PA_RAMP_SEL] > 00), the contents of this register are presented to the PA during PA ramping, when TSM tx_en transitions low to high, and then for the duration of the first ramp step. During PA ramp down, the contents of this register are the PA power value during the final ramp step. In both cases, PA_BIAS0 cannot exceed target power (enforced by PA ramping logic). When PA ramping is enabled, the contents of PA_BIAS0 are also presented to the PA during sequence-idle conditions.

52.1.91 PA Bias Table 1 (XCVR_PA_BIAS_TBL1)

Address: 4005_C000h base + 2E0h offset = 4005_C2E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PA_BIAS7				0				PA_BIAS6				0				PA_BIAS5				0				PA_BIAS4			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PA_BIAS_TBL1 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 PA_BIAS7	PA_BIAS7 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the eighth (final) ramp step. During PA ramp down, the contents of this register are the PA power value during the eighth-to-last (first) ramp step. In both cases, PA_BIAS7 cannot exceed target power (enforced by PA ramping logic).
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 PA_BIAS6	PA_BIAS6 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the seventh ramp step. During PA ramp down, the contents

Table continues on the next page...

XCVR_PA_BIAS_TBL1 field descriptions (continued)

Field	Description
	of this register are the PA power value during the seventh-to-last ramp step. In both cases, PA_BIAS6 cannot exceed target power (enforced by PA ramping logic).
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 PA_BIAS5	PA_BIAS5 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the sixth ramp step. During PA ramp down, the contents of this register are the PA power value during the sixth-to-last ramp step. In both cases, PA_BIAS5 cannot exceed target power (enforced by PA ramping logic).
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PA_BIAS4	PA_BIAS4 If PA ramping is enabled (TSM_CTRL[PA_RAMP_EN] > 00), the contents of this register are presented to the PA during PA ramping, for the duration of the fifth ramp step. During PA ramp down, the contents of this register are the PA power value during the fifth-to-last ramp step. In both cases, PA_BIAS4 cannot exceed target power (enforced by PA ramping logic).

52.1.92 Recycle Count Register (XCVR_RECYCLE_COUNT)

Address: 4005_C000h base + 2E4h offset = 4005_C2E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RECYCLE_COUNT1								RECYCLE_COUNT0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0

XCVR_RECYCLE_COUNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 RECYCLE_COUNT1	TSM RX Recycle Count 1 The RECYCLE_COUNT1[7:0] register determines the TSM count value to which the TSM "recycles" when the Zigbee Sequence Manager (ZSM) state "RX_PAN1" is reached and the ZSM asserts tsm_recycle[1] to TSM. The intention is for this register to be programmed to a TSM count value such that the TSM de-asserts, and then re-asserts its "pll_dig_en" output, to effectuate a Dual PAN on-the-fly channel change, but there are no restrictions on programming this register. An RX recycle is a command from ZSM to TSM to jump from the TSM ON phase back to a programmable point in the WARMUP phase, and resume counting from there. A recycle will result from the expiration of the Dual PAN Dwell Timer, at which point an RF-channel change is required. This necessitates the desassertion and reassertion of pll_dig_en, hence the return to the WARMUP phase at the appropriate point.
RECYCLE_COUNT0	TSM RX Recycle Count 0 The RECYCLE_COUNT0[7:0] register determines the TSM count value to which the TSM "recycles" when the Zigbee Sequence Manager (ZSM) state "RX_CYC" is reached and the ZSM asserts tsm_recycle[0] to TSM. This register also determines the TSM count value to which the TSM recycles when the ZSM state

Table continues on the next page...

XCVR_RECYCLE_COUNT field descriptions (continued)

Field	Description
	RX_CCCA is reached because tsm_recycle[0] is also asserted in this state. The intention is for this register to be programmed to a TSM count value such that the TSM re-asserts its "rx_init" output, but there are no restrictions on programming this register. An RX recycle is a command from ZSM to TSM to jump from the TSM ON phase back to a programmable point in the WARMUP phase, and resume counting from there. A recycle will result from the reception of a packet with bad CRC or one which fails packet-filtering rules, or the end of a CCA operation in Continuous CCA mode which results in a channel indicating "busy".

52.1.93 TSM_TIMING00 (XCVR_TSM_TIMING00)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_REG_EN TSM signal or signal group.

Address: 4005_C000h base + 2E8h offset = 4005_C2E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_REG_EN_RX_LO								PLL_REG_EN_RX_HI								PLL_REG_EN_TX_LO								PLL_REG_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

XCVR_TSM_TIMING00 field descriptions

Field	Description
31–24 PLL_REG_EN_ RX_LO	Deassertion time setting for PLL_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_REG_EN signal or group will transition from HI to LO.
23–16 PLL_REG_EN_ RX_HI	Assertion time setting for PLL_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_REG_EN signal or group will transition from LO to HI.
15–8 PLL_REG_EN_ TX_LO	Deassertion time setting for PLL_REG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_REG_EN signal or group will transition from HI to LO.
PLL_REG_EN_ TX_HI	Assertion time setting for PLL_REG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_REG_EN signal or group will transition from LO to HI.

52.1.94 TSM_TIMING01 (XCVR_TSM_TIMING01)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_VCO_REG_EN TSM signal or signal group.

Address: 4005_C000h base + 2ECh offset = 4005_C2ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_VCO_REG_EN_RX_LO								PLL_VCO_REG_EN_RX_HI								PLL_VCO_REG_EN_TX_LO								PLL_VCO_REG_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

XCVR_TSM_TIMING01 field descriptions

Field	Description
31–24 PLL_VCO_REG_EN_RX_LO	Deassertion time setting for PLL_VCO_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_REG_EN signal or group will transition from HI to LO.
23–16 PLL_VCO_REG_EN_RX_HI	Assertion time setting for PLL_VCO_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_REG_EN signal or group will transition from LO to HI.
15–8 PLL_VCO_REG_EN_TX_LO	Deassertion time setting for PLL_VCO_REG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_REG_EN signal or group will transition from HI to LO.
PLL_VCO_REG_EN_TX_HI	Assertion time setting for PLL_VCO_REG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_REG_EN signal or group will transition from LO to HI.

52.1.95 TSM_TIMING02 (XCVR_TSM_TIMING02)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the QGEN_REG_EN TSM signal or signal group.

Address: 4005_C000h base + 2F0h offset = 4005_C2F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	QGEN_REG_EN_RX_LO								QGEN_REG_EN_RX_HI								QGEN_REG_EN_TX_LO								QGEN_REG_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

XCVR_TSM_TIMING02 field descriptions

Field	Description
31–24 QGEN_REG_ EN_RX_LO	Deassertion time setting for QGEN_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the QGEN_REG_EN signal or group will transition from HI to LO.
23–16 QGEN_REG_ EN_RX_HI	Assertion time setting for QGEN_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the QGEN_REG_EN signal or group will transition from LO to HI.
15–8 QGEN_REG_ EN_TX_LO	Deassertion time setting for QGEN_REG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the QGEN_REG_EN signal or group will transition from HI to LO.
QGEN_REG_ EN_TX_HI	Assertion time setting for QGEN_REG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the QGEN_REG_EN signal or group will transition from LO to HI.

52.1.96 TSM_TIMING03 (XCVR_TSM_TIMING03)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TCA_TX_REG_EN TSM signal or signal group.

Address: 4005_C000h base + 2F4h offset = 4005_C2F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCA_TX_REG_EN_RX_LO								TCA_TX_REG_EN_RX_HI								TCA_TX_REG_EN_TX_LO								TCA_TX_REG_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

XCVR_TSM_TIMING03 field descriptions

Field	Description
31–24 TCA_TX_REG_ EN_RX_LO	Deassertion time setting for TCA_TX_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TCA_TX_REG_EN signal or group will transition from HI to LO.
23–16 TCA_TX_REG_ EN_RX_HI	Assertion time setting for TCA_TX_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TCA_TX_REG_EN signal or group will transition from LO to HI.
15–8 TCA_TX_REG_ EN_TX_LO	Deassertion time setting for TCA_TX_REG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TCA_TX_REG_EN signal or group will transition from HI to LO.
TCA_TX_REG_ EN_TX_HI	Assertion time setting for TCA_TX_REG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TCA_TX_REG_EN signal or group will transition from LO to HI.

52.1.97 TSM_TIMING04 (XCVR_TSM_TIMING04)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_REG_EN TSM signal or signal group.

Address: 4005_C000h base + 2F8h offset = 4005_C2F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_REG_EN_RX_LO								ADC_REG_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING04 field descriptions

Field	Description
31–24 ADC_REG_EN_ RX_LO	Deassertion time setting for ADC_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_REG_EN signal or group will transition from HI to LO.
23–16 ADC_REG_EN_ RX_HI	Assertion time setting for ADC_REG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_REG_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.98 TSM_TIMING05 (XCVR_TSM_TIMING05)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_REF_CLK_EN TSM signal or signal group.

Address: 4005_C000h base + 2FCh offset = 4005_C2FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_REF_CLK_EN_RX_LO								PLL_REF_CLK_EN_RX_HI								PLL_REF_CLK_EN_TX_LO								PLL_REF_CLK_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	1	0	1	1	0	1	1	0	1	0	1	0	0	0	1	1	1	1	1	1

XCVR_TSM_TIMING05 field descriptions

Field	Description
31–24 PLL_REF_CLK_ EN_RX_LO	Deassertion time setting for PLL_REF_CLK_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_REF_CLK_EN signal or group will transition from HI to LO.
23–16 PLL_REF_CLK_ EN_RX_HI	Assertion time setting for PLL_REF_CLK_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_REF_CLK_EN signal or group will transition from LO to HI.

Table continues on the next page...

XCVR_TSM_TIMING05 field descriptions (continued)

Field	Description
15–8 PLL_REF_CLK_ EN_TX_LO	Deassertion time setting for PLL_REF_CLK_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_REF_CLK_EN signal or group will transition from HI to LO.
PLL_REF_CLK_ EN_TX_HI	Assertion time setting for PLL_REF_CLK_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_REF_CLK_EN signal or group will transition from LO to HI.

52.1.99 TSM_TIMING06 (XCVR_TSM_TIMING06)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_CLK_EN TSM signal or signal group.

Address: 4005_C000h base + 300h offset = 4005_C300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_CLK_EN_RX_LO								ADC_CLK_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING06 field descriptions

Field	Description
31–24 ADC_CLK_EN_ RX_LO	Deassertion time setting for ADC_CLK_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_CLK_EN signal or group will transition from HI to LO.
23–16 ADC_CLK_EN_ RX_HI	Assertion time setting for ADC_CLK_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_CLK_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.100 TSM_TIMING07 (XCVR_TSM_TIMING07)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_VCO_AUTOTUNE_EN TSM signal or signal group.

Address: 4005_C000h base + 304h offset = 4005_C304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_VCO_AUTOTUNE_EN_ RX_LO								PLL_VCO_AUTOTUNE_EN_ RX_HI								PLL_VCO_AUTOTUNE_EN_ TX_LO								PLL_VCO_AUTOTUNE_EN_ TX_HI							
Reset	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0

XCVR_TSM_TIMING07 field descriptions

Field	Description
31–24 PLL_VCO_ AUTOTUNE_ EN_RX_LO	Deassertion time setting for PLL_VCO_AUTOTUNE_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_AUTOTUNE_EN signal or group will transition from HI to LO.
23–16 PLL_VCO_ AUTOTUNE_ EN_RX_HI	Assertion time setting for PLL_VCO_AUTOTUNE_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_AUTOTUNE_EN signal or group will transition from LO to HI.
15–8 PLL_VCO_ AUTOTUNE_ EN_TX_LO	Deassertion time setting for PLL_VCO_AUTOTUNE_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_AUTOTUNE_EN signal or group will transition from HI to LO.
PLL_VCO_ AUTOTUNE_ EN_TX_HI	Assertion time setting for PLL_VCO_AUTOTUNE_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_AUTOTUNE_EN signal or group will transition from LO to HI.

52.1.101 TSM_TIMING08 (XCVR_TSM_TIMING08)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_CYCLE_SLIP_LD_EN TSM signal or signal group.

Address: 4005_C000h base + 308h offset = 4005_C308h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_CYCLE_SLIP_LD_EN_ RX_LO								PLL_CYCLE_SLIP_LD_EN_ RX_HI								PLL_CYCLE_SLIP_LD_EN_ TX_LO								PLL_CYCLE_SLIP_LD_EN_ TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1

XCVR_TSM_TIMING08 field descriptions

Field	Description
31–24 PLL_CYCLE_SLIP_LD_EN_RX_LO	Deassertion time setting for PLL_CYCLE_SLIP_LD_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_CYCLE_SLIP_LD_EN signal or group will transition from HI to LO.
23–16 PLL_CYCLE_SLIP_LD_EN_RX_HI	Assertion time setting for PLL_CYCLE_SLIP_LD_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_CYCLE_SLIP_LD_EN signal or group will transition from LO to HI.
15–8 PLL_CYCLE_SLIP_LD_EN_TX_LO	Deassertion time setting for PLL_CYCLE_SLIP_LD_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_CYCLE_SLIP_LD_EN signal or group will transition from HI to LO.
PLL_CYCLE_SLIP_LD_EN_TX_HI	Assertion time setting for PLL_CYCLE_SLIP_LD_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_CYCLE_SLIP_LD_EN signal or group will transition from LO to HI.

52.1.102 TSM_TIMING09 (XCVR_TSM_TIMING09)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_VCO_EN TSM signal or signal group.

Address: 4005_C000h base + 30Ch offset = 4005_C30Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_VCO_EN_RX_LO								PLL_VCO_EN_RX_HI								PLL_VCO_EN_TX_LO								PLL_VCO_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1

XCVR_TSM_TIMING09 field descriptions

Field	Description
31–24 PLL_VCO_EN_RX_LO	Deassertion time setting for PLL_VCO_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_EN signal or group will transition from HI to LO.
23–16 PLL_VCO_EN_RX_HI	Assertion time setting for PLL_VCO_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_EN signal or group will transition from LO to HI.
15–8 PLL_VCO_EN_TX_LO	Deassertion time setting for PLL_VCO_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_EN signal or group will transition from HI to LO.
PLL_VCO_EN_TX_HI	Assertion time setting for PLL_VCO_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_EN signal or group will transition from LO to HI.

52.1.103 TSM_TIMING10 (XCVR_TSM_TIMING10)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_VCO_BUF_RX_EN TSM signal or signal group.

Address: 4005_C000h base + 310h offset = 4005_C310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PLL_VCO_BUF_RX_EN_ RX_LO								PLL_VCO_BUF_RX_EN_ RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING10 field descriptions

Field	Description
31–24 PLL_VCO_BUF_ RX_EN_RX_LO	Deassertion time setting for PLL_VCO_BUF_RX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_BUF_RX_EN signal or group will transition from HI to LO.
23–16 PLL_VCO_BUF_ RX_EN_RX_HI	Assertion time setting for PLL_VCO_BUF_RX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_VCO_BUF_RX_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.104 TSM_TIMING11 (XCVR_TSM_TIMING11)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_VCO_BUF_TX_EN TSM signal or signal group.

Address: 4005_C000h base + 314h offset = 4005_C314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1																PLL_VCO_BUF_TX_EN_ TX_LO								PLL_VCO_BUF_TX_EN_ TX_HI							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0	1

XCVR_TSM_TIMING11 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

Table continues on the next page...

XCVR_TSM_TIMING11 field descriptions (continued)

Field	Description
15–8 PLL_VCO_BUF_ TX_EN_TX_LO	Deassertion time setting for PLL_VCO_BUF_TX_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_BUF_TX_EN signal or group will transition from HI to LO.
PLL_VCO_BUF_ TX_EN_TX_HI	Assertion time setting for PLL_VCO_BUF_TX_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_VCO_BUF_TX_EN signal or group will transition from LO to HI.

52.1.105 TSM_TIMING12 (XCVR_TSM_TIMING12)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_PA_BUF_EN TSM signal or signal group.

Address: 4005_C000h base + 318h offset = 4005_C318h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1																PLL_PA_BUF_EN_TX_LO								PLL_PA_BUF_EN_TX_HI							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0

XCVR_TSM_TIMING12 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15–8 PLL_PA_BUF_ EN_TX_LO	Deassertion time setting for PLL_PA_BUF_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_PA_BUF_EN signal or group will transition from HI to LO.
PLL_PA_BUF_ EN_TX_HI	Assertion time setting for PLL_PA_BUF_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_PA_BUF_EN signal or group will transition from LO to HI.

52.1.106 TSM_TIMING13 (XCVR_TSM_TIMING13)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_LDV_EN TSM signal or signal group.

Address: 4005_C000h base + 31Ch offset = 4005_C31Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_LDV_EN_RX_LO								PLL_LDV_EN_RX_HI								PLL_LDV_EN_TX_LO								PLL_LDV_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	1	0	0	1	0	0	1	1	1	0

XCVR_TSM_TIMING13 field descriptions

Field	Description
31–24 PLL_LDV_EN_ RX_LO	Deassertion time setting for PLL_LDV_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LDV_EN signal or group will transition from HI to LO.
23–16 PLL_LDV_EN_ RX_HI	Assertion time setting for PLL_LDV_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LDV_EN signal or group will transition from LO to HI.
15–8 PLL_LDV_EN_ TX_LO	Deassertion time setting for PLL_LDV_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LDV_EN signal or group will transition from HI to LO.
PLL_LDV_EN_ TX_HI	Assertion time setting for PLL_LDV_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LDV_EN signal or group will transition from LO to HI.

52.1.107 TSM_TIMING14 (XCVR_TSM_TIMING14)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_RX_LDV_RIPPLE_MUX_EN TSM signal or signal group.

Address: 4005_C000h base + 320h offset = 4005_C320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PLL_RX_LDV_RIPPLE_ MUX_EN_RX_LO								PLL_RX_LDV_RIPPLE_ MUX_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING14 field descriptions

Field	Description
31–24 PLL_RX_LDV_ RIPPLE_MUX_ EN_RX_LO	Deassertion time setting for PLL_RX_LDV_RIPPLE_MUX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_RX_LDV_RIPPLE_MUX_EN signal or group will transition from HI to LO.
23–16 PLL_RX_LDV_ RIPPLE_MUX_ EN_RX_HI	Assertion time setting for PLL_RX_LDV_RIPPLE_MUX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_RX_LDV_RIPPLE_MUX_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.108 TSM_TIMING15 (XCVR_TSM_TIMING15)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_TX_LDV_RIPPLE_MUX_EN TSM signal or signal group.

Address: 4005_C000h base + 324h offset = 4005_C324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1																PLL_TX_LDV_RIPPLE_ MUX_EN_TX_LO								PLL_TX_LDV_RIPPLE_ MUX_EN_TX_HI							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	0

XCVR_TSM_TIMING15 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15–8 PLL_TX_LDV_ RIPPLE_MUX_ EN_TX_LO	Deassertion time setting for PLL_TX_LDV_RIPPLE_MUX_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_TX_LDV_RIPPLE_MUX_EN signal or group will transition from HI to LO.
PLL_TX_LDV_ RIPPLE_MUX_ EN_TX_HI	Assertion time setting for PLL_TX_LDV_RIPPLE_MUX_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_TX_LDV_RIPPLE_MUX_EN signal or group will transition from LO to HI.

52.1.109 TSM_TIMING16 (XCVR_TSM_TIMING16)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_FILTER_CHARGE_EN TSM signal or signal group.

Address: 4005_C000h base + 328h offset = 4005_C328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_FILTER_CHARGE_ EN_RX_LO								PLL_FILTER_CHARGE_ EN_RX_HI								PLL_FILTER_CHARGE_ EN_TX_LO								PLL_FILTER_CHARGE_ EN_TX_HI							
Reset	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0	0	0	1	0	0

XCVR_TSM_TIMING16 field descriptions

Field	Description
31–24 PLL_FILTER_ CHARGE_EN_ RX_LO	Deassertion time setting for PLL_FILTER_CHARGE_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_FILTER_CHARGE_EN signal or group will transition from HI to LO.

Table continues on the next page...

XCVR_TSM_TIMING16 field descriptions (continued)

Field	Description
23–16 PLL_FILTER_ CHARGE_EN_ RX_HI	Assertion time setting for PLL_FILTER_CHARGE_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_FILTER_CHARGE_EN signal or group will transition from LO to HI.
15–8 PLL_FILTER_ CHARGE_EN_ TX_LO	Deassertion time setting for PLL_FILTER_CHARGE_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_FILTER_CHARGE_EN signal or group will transition from HI to LO.
PLL_FILTER_ CHARGE_EN_ TX_HI	Assertion time setting for PLL_FILTER_CHARGE_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_FILTER_CHARGE_EN signal or group will transition from LO to HI.

52.1.110 TSM_TIMING17 (XCVR_TSM_TIMING17)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_PHDET_EN TSM signal or signal group.

Address: 4005_C000h base + 32Ch offset = 4005_C32Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_PHDET_EN_RX_LO								PLL_PHDET_EN_RX_HI								PLL_PHDET_EN_TX_LO								PLL_PHDET_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0

XCVR_TSM_TIMING17 field descriptions

Field	Description
31–24 PLL_PHDET_ EN_RX_LO	Deassertion time setting for PLL_PHDET_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_PHDET_EN signal or group will transition from HI to LO.
23–16 PLL_PHDET_ EN_RX_HI	Assertion time setting for PLL_PHDET_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_PHDET_EN signal or group will transition from LO to HI.
15–8 PLL_PHDET_ EN_TX_LO	Deassertion time setting for PLL_PHDET_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_PHDET_EN signal or group will transition from HI to LO.
PLL_PHDET_ EN_TX_HI	Assertion time setting for PLL_PHDET_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_PHDET_EN signal or group will transition from LO to HI.

52.1.111 TSM_TIMING18 (XCVR_TSM_TIMING18)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the QGEN25_EN TSM signal or signal group.

Address: 4005_C000h base + 330h offset = 4005_C330h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	QGEN25_EN_RX_LO								QGEN25_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING18 field descriptions

Field	Description
31–24 QGEN25_EN_ RX_LO	Deassertion time setting for QGEN25_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the QGEN25_EN signal or group will transition from HI to LO.
23–16 QGEN25_EN_ RX_HI	Assertion time setting for QGEN25_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the QGEN25_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.112 TSM_TIMING19 (XCVR_TSM_TIMING19)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TX_EN TSM signal or signal group.

Address: 4005_C000h base + 334h offset = 4005_C334h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1																TX_EN_TX_LO								TX_EN_TX_HI							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	0	0

XCVR_TSM_TIMING19 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15–8 TX_EN_TX_LO	Deassertion time setting for TX_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_EN signal or group will transition from HI to LO.
TX_EN_TX_HI	Assertion time setting for TX_EN TX sequence.

Table continues on the next page...

XCVR_TSM_TIMING19 field descriptions (continued)

Field	Description
	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_EN signal or group will transition from LO to HI.

52.1.113 TSM_TIMING20 (XCVR_TSM_TIMING20)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_EN TSM signal or signal group.

Address: 4005_C000h base + 338h offset = 4005_C338h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_EN_RX_LO								ADC_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING20 field descriptions

Field	Description
31–24 ADC_EN_RX_LO	Deassertion time setting for ADC_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_EN signal or group will transition from HI to LO.
23–16 ADC_EN_RX_HI	Assertion time setting for ADC_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.114 TSM_TIMING21 (XCVR_TSM_TIMING21)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_I_Q_EN TSM signal or signal group.

Address: 4005_C000h base + 33Ch offset = 4005_C33Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_I_Q_EN_RX_LO								ADC_I_Q_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING21 field descriptions

Field	Description
31–24 ADC_I_Q_EN_ RX_LO	Deassertion time setting for ADC_I_Q_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_I_Q_EN signal or group will transition from HI to LO.
23–16 ADC_I_Q_EN_ RX_HI	Assertion time setting for ADC_I_Q_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_I_Q_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.115 TSM_TIMING22 (XCVR_TSM_TIMING22)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_DAC_EN TSM signal or signal group.

Address: 4005_C000h base + 340h offset = 4005_C340h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_DAC_EN_RX_LO								ADC_DAC_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING22 field descriptions

Field	Description
31–24 ADC_DAC_EN_ RX_LO	Deassertion time setting for ADC_DAC_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_DAC_EN signal or group will transition from HI to LO.
23–16 ADC_DAC_EN_ RX_HI	Assertion time setting for ADC_DAC_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_DAC_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.116 TSM_TIMING23 (XCVR_TSM_TIMING23)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ADC_RST_EN TSM signal or signal group.

Address: 4005_C000h base + 344h offset = 4005_C344h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_RST_EN_RX_LO								ADC_RST_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING23 field descriptions

Field	Description
31–24 ADC_RST_EN_ RX_LO	Deassertion time setting for ADC_RST_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_RST_EN signal or group will transition from HI to LO.
23–16 ADC_RST_EN_ RX_HI	Assertion time setting for ADC_RST_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ADC_RST_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.117 TSM_TIMING24 (XCVR_TSM_TIMING24)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the BBF_EN TSM signal or signal group.

Address: 4005_C000h base + 348h offset = 4005_C348h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BBF_EN_RX_LO								BBF_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING24 field descriptions

Field	Description
31–24 BBF_EN_RX_LO	Deassertion time setting for BBF_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BBF_EN signal or group will transition from HI to LO.
23–16 BBF_EN_RX_HI	Assertion time setting for BBF_EN signal or group RX sequence.

Table continues on the next page...

XCVR_TSM_TIMING24 field descriptions (continued)

Field	Description
	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BBF_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.118 TSM_TIMING25 (XCVR_TSM_TIMING25)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TCA_EN TSM signal or signal group.

Address: 4005_C000h base + 34Ch offset = 4005_C34Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TCA_EN_RX_LO								TCA_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING25 field descriptions

Field	Description
31–24 TCA_EN_RX_LO	Deassertion time setting for TCA_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TCA_EN signal or group will transition from HI to LO.
23–16 TCA_EN_RX_HI	Assertion time setting for TCA_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TCA_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.119 TSM_TIMING26 (XCVR_TSM_TIMING26)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the PLL_DIG_EN TSM signal or signal group.

Address: 4005_C000h base + 350h offset = 4005_C350h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_DIG_EN_RX_LO								PLL_DIG_EN_RX_HI								PLL_DIG_EN_TX_LO								PLL_DIG_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0	1

XCVR_TSM_TIMING26 field descriptions

Field	Description
31–24 PLL_DIG_EN_ RX_LO	Deassertion time setting for PLL_DIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_DIG_EN signal or group will transition from HI to LO.
23–16 PLL_DIG_EN_ RX_HI	Assertion time setting for PLL_DIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_DIG_EN signal or group will transition from LO to HI.
15–8 PLL_DIG_EN_ TX_LO	Deassertion time setting for PLL_DIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_DIG_EN signal or group will transition from HI to LO.
PLL_DIG_EN_ TX_HI	Assertion time setting for PLL_DIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_DIG_EN signal or group will transition from LO to HI.

52.1.120 TSM_TIMING27 (XCVR_TSM_TIMING27)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TX_DIG_EN TSM signal or signal group.

Address: 4005_C000h base + 354h offset = 4005_C354h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1																TX_DIG_EN_TX_LO								TX_DIG_EN_TX_HI							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	1	1

XCVR_TSM_TIMING27 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15–8 TX_DIG_EN_ TX_LO	Deassertion time setting for TX_DIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_DIG_EN signal or group will transition from HI to LO.
TX_DIG_EN_ TX_HI	Assertion time setting for TX_DIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_DIG_EN signal or group will transition from LO to HI.

52.1.121 TSM_TIMING28 (XCVR_TSM_TIMING28)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the RX_DIG_EN TSM signal or signal group.

Address: 4005_C000h base + 358h offset = 4005_C358h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX_DIG_EN_RX_LO								RX_DIG_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING28 field descriptions

Field	Description
31–24 RX_DIG_EN_ RX_LO	Deassertion time setting for RX_DIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_DIG_EN signal or group will transition from HI to LO.
23–16 RX_DIG_EN_ RX_HI	Assertion time setting for RX_DIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_DIG_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.122 TSM_TIMING29 (XCVR_TSM_TIMING29)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the RX_INIT TSM signal or signal group.

Address: 4005_C000h base + 35Ch offset = 4005_C35Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX_INIT_RX_LO								RX_INIT_RX_HI								1															
W																																
Reset	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING29 field descriptions

Field	Description
31–24 RX_INIT_RX_LO	Deassertion time setting for RX_INIT signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_INIT signal or group will transition from HI to LO.
23–16 RX_INIT_RX_HI	Assertion time setting for RX_INIT signal or group RX sequence.

Table continues on the next page...

XCVR_TSM_TIMING29 field descriptions (continued)

Field	Description
	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_INIT signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.123 TSM_TIMING30 (XCVR_TSM_TIMING30)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the SIGMA_DELTA_EN TSM signal or signal group.

Address: 4005_C000h base + 360h offset = 4005_C360h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SIGMA_DELTA_EN_RX_LO								SIGMA_DELTA_EN_RX_HI								SIGMA_DELTA_EN_TX_LO								SIGMA_DELTA_EN_TX_HI							
Reset	0	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0

XCVR_TSM_TIMING30 field descriptions

Field	Description
31–24 SIGMA_DELTA_EN_RX_LO	Deassertion time setting for SIGMA_DELTA_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SIGMA_DELTA_EN signal or group will transition from HI to LO.
23–16 SIGMA_DELTA_EN_RX_HI	Assertion time setting for SIGMA_DELTA_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SIGMA_DELTA_EN signal or group will transition from LO to HI.
15–8 SIGMA_DELTA_EN_TX_LO	Deassertion time setting for SIGMA_DELTA_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SIGMA_DELTA_EN signal or group will transition from HI to LO.
SIGMA_DELTA_EN_TX_HI	Assertion time setting for SIGMA_DELTA_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SIGMA_DELTA_EN signal or group will transition from LO to HI.

52.1.124 TSM_TIMING31 (XCVR_TSM_TIMING31)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the ZBDEM_RX_EN TSM signal or signal group.

Address: 4005_C000h base + 364h offset = 4005_C364h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ZBDEM_RX_EN_RX_LO								ZBDEM_RX_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING31 field descriptions

Field	Description
31–24 ZBDEM_RX_ EN_RX_LO	Deassertion time setting for ZBDEM_RX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ZBDEM_RX_EN signal or group will transition from HI to LO.
23–16 ZBDEM_RX_ EN_RX_HI	Assertion time setting for ZBDEM_RX_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the ZBDEM_RX_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.125 TSM_TIMING32 (XCVR_TSM_TIMING32)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the DCOC_EN TSM signal or signal group.

Address: 4005_C000h base + 368h offset = 4005_C368h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCOC_EN_RX_LO								DCOC_EN_RX_HI								1															
W																																
Reset	0	1	1	0	0	1	0	1	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING32 field descriptions

Field	Description
31–24 DCOC_EN_RX_ LO	Deassertion time setting for DCOC_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the DCOC_EN signal or group will transition from HI to LO.
23–16 DCOC_EN_RX_ HI	Assertion time setting for DCOC_EN signal or group RX sequence.

Table continues on the next page...

XCVR_TSM_TIMING32 field descriptions (continued)

Field	Description
	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the DCOC_EN signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.126 TSM_TIMING33 (XCVR_TSM_TIMING33)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the DCOC_INIT TSM signal or signal group.

Address: 4005_C000h base + 36Ch offset = 4005_C36Ch

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="8" rowspan="2">DCOC_INIT_RX_LO</td><td colspan="8" rowspan="2">DCOC_INIT_RX_HI</td><td colspan="16">1</td></tr>
<tr><td>W</td><td colspan="16"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>

XCVR_TSM_TIMING33 field descriptions

Field	Description
31–24 DCOC_INIT_RX_LO	Deassertion time setting for DCOC_INIT signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the DCOC_INIT signal or group will transition from HI to LO.
23–16 DCOC_INIT_RX_HI	Assertion time setting for DCOC_INIT signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the DCOC_INIT signal or group will transition from LO to HI.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

52.1.127 TSM_TIMING34 (XCVR_TSM_TIMING34)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the FREQ_TARG_LD_EN TSM signal or signal group.

Address: 4005_C000h base + 370h offset = 4005_C370h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="8" rowspan="2">FREQ_TARG_LD_EN_RX_LO</td><td colspan="8" rowspan="2">FREQ_TARG_LD_EN_RX_HI</td><td colspan="8" rowspan="2">FREQ_TARG_LD_EN_TX_LO</td><td colspan="8" rowspan="2">FREQ_TARG_LD_EN_TX_HI</td></tr>
<tr><td>W</td></tr>
<tr><td>Reset</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

XCVR_TSM_TIMING34 field descriptions

Field	Description
31–24 FREQ_TARG_ LD_EN_RX_LO	Deassertion time setting for FREQ_TARG_LD_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the FREQ_TARG_LD_EN signal or group will transition from HI to LO.
23–16 FREQ_TARG_ LD_EN_RX_HI	Assertion time setting for FREQ_TARG_LD_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the FREQ_TARG_LD_EN signal or group will transition from LO to HI.
15–8 FREQ_TARG_ LD_EN_TX_LO	Deassertion time setting for FREQ_TARG_LD_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the FREQ_TARG_LD_EN signal or group will transition from HI to LO.
FREQ_TARG_ LD_EN_TX_HI	Assertion time setting for FREQ_TARG_LD_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the FREQ_TARG_LD_EN signal or group will transition from LO to HI.

52.1.128 TSM_TIMING35 (XCVR_TSM_TIMING35)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the SAR_ADC_TRIG_EN TSM signal or signal group.

Address: 4005_C000h base + 374h offset = 4005_C374h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SAR_ADC_TRIG_EN_RX_ LO								SAR_ADC_TRIG_EN_RX_ HI								SAR_ADC_TRIG_EN_TX_ LO								SAR_ADC_TRIG_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING35 field descriptions

Field	Description
31–24 SAR_ADC_ TRIG_EN_RX_ LO	Deassertion time setting for SAR_ADC_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SAR_ADC_TRIG_EN signal or group will transition from HI to LO.
23–16 SAR_ADC_ TRIG_EN_RX_HI	Assertion time setting for SAR_ADC_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SAR_ADC_TRIG_EN signal or group will transition from LO to HI.
15–8 SAR_ADC_ TRIG_EN_TX_ LO	Deassertion time setting for SAR_ADC_TRIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SAR_ADC_TRIG_EN signal or group will transition from HI to LO.
SAR_ADC_ TRIG_EN_TX_HI	Assertion time setting for SAR_ADC_TRIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SAR_ADC_TRIG_EN signal or group will transition from LO to HI.

52.1.129 TSM_TIMING36 (XCVR_TSM_TIMING36)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TSM_SPARE0_EN TSM signal or signal group.

Address: 4005_C000h base + 378h offset = 4005_C378h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSM_SPARE0_EN_RX_LO								TSM_SPARE0_EN_RX_HI								TSM_SPARE0_EN_TX_LO								TSM_SPARE0_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING36 field descriptions

Field	Description
31–24 TSM_SPARE0_ EN_RX_LO	Deassertion time setting for TSM_SPARE0_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE0_EN signal or group will transition from HI to LO.
23–16 TSM_SPARE0_ EN_RX_HI	Assertion time setting for TSM_SPARE0_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE0_EN signal or group will transition from LO to HI.
15–8 TSM_SPARE0_ EN_TX_LO	Deassertion time setting for TSM_SPARE0_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE0_EN signal or group will transition from HI to LO.
TSM_SPARE0_ EN_TX_HI	Assertion time setting for TSM_SPARE0_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE0_EN signal or group will transition from LO to HI.

52.1.130 TSM_TIMING37 (XCVR_TSM_TIMING37)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TSM_SPARE1_EN TSM signal or signal group.

Address: 4005_C000h base + 37Ch offset = 4005_C37Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSM_SPARE1_EN_RX_LO								TSM_SPARE1_EN_RX_HI								TSM_SPARE1_EN_TX_LO								TSM_SPARE1_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING37 field descriptions

Field	Description
31–24 TSM_SPARE1_ EN_RX_LO	Deassertion time setting for TSM_SPARE1_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE1_EN signal or group will transition from HI to LO.
23–16 TSM_SPARE1_ EN_RX_HI	Assertion time setting for TSM_SPARE1_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE1_EN signal or group will transition from LO to HI.
15–8 TSM_SPARE1_ EN_TX_LO	Deassertion time setting for TSM_SPARE1_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE1_EN signal or group will transition from HI to LO.
TSM_SPARE1_ EN_TX_HI	Assertion time setting for TSM_SPARE1_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE1_EN signal or group will transition from LO to HI.

52.1.131 TSM_TIMING38 (XCVR_TSM_TIMING38)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TSM_SPARE2_EN TSM signal or signal group.

Address: 4005_C000h base + 380h offset = 4005_C380h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSM_SPARE2_EN_RX_LO								TSM_SPARE2_EN_RX_HI								TSM_SPARE2_EN_TX_LO								TSM_SPARE2_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING38 field descriptions

Field	Description
31–24 TSM_SPARE2_ EN_RX_LO	Deassertion time setting for TSM_SPARE2_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE2_EN signal or group will transition from HI to LO.
23–16 TSM_SPARE2_ EN_RX_HI	Assertion time setting for TSM_SPARE2_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE2_EN signal or group will transition from LO to HI.
15–8 TSM_SPARE2_ EN_TX_LO	Deassertion time setting for TSM_SPARE2_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE2_EN signal or group will transition from HI to LO.
TSM_SPARE2_ EN_TX_HI	Assertion time setting for TSM_SPARE2_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE2_EN signal or group will transition from LO to HI.

52.1.132 TSM_TIMING39 (XCVR_TSM_TIMING39)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the TSM_SPARE3_EN TSM signal or signal group.

Address: 4005_C000h base + 384h offset = 4005_C384h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSM_SPARE3_EN_RX_LO								TSM_SPARE3_EN_RX_HI								TSM_SPARE3_EN_TX_LO								TSM_SPARE3_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING39 field descriptions

Field	Description
31–24 TSM_SPARE3_ EN_RX_LO	Deassertion time setting for TSM_SPARE3_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE3_EN signal or group will transition from HI to LO.
23–16 TSM_SPARE3_ EN_RX_HI	Assertion time setting for TSM_SPARE3_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the TSM_SPARE3_EN signal or group will transition from LO to HI.
15–8 TSM_SPARE3_ EN_TX_LO	Deassertion time setting for TSM_SPARE3_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE3_EN signal or group will transition from HI to LO.
TSM_SPARE3_ EN_TX_HI	Assertion time setting for TSM_SPARE3_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TSM_SPARE3_EN signal or group will transition from LO to HI.

52.1.133 TSM_TIMING40 (XCVR_TSM_TIMING40)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the GPIO0_TRIG_EN TSM signal or signal group.

Address: 4005_C000h base + 388h offset = 4005_C388h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO0_TRIG_EN_RX_LO								GPIO0_TRIG_EN_RX_HI								GPIO0_TRIG_EN_TX_LO								GPIO0_TRIG_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING40 field descriptions

Field	Description
31–24 GPIO0_TRIG_ EN_RX_LO	Deassertion time setting for GPIO0_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO0_TRIG_EN signal or group will transition from HI to LO.
23–16 GPIO0_TRIG_ EN_RX_HI	Assertion time setting for GPIO0_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO0_TRIG_EN signal or group will transition from LO to HI.
15–8 GPIO0_TRIG_ EN_TX_LO	Deassertion time setting for GPIO0_TRIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO0_TRIG_EN signal or group will transition from HI to LO.
GPIO0_TRIG_ EN_TX_HI	Assertion time setting for GPIO0_TRIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO0_TRIG_EN signal or group will transition from LO to HI.

52.1.134 TSM_TIMING41 (XCVR_TSM_TIMING41)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the GPIO1_TRIG_EN TSM signal or signal group.

Address: 4005_C000h base + 38Ch offset = 4005_C38Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO1_TRIG_EN_RX_LO								GPIO1_TRIG_EN_RX_HI								GPIO1_TRIG_EN_TX_LO								GPIO1_TRIG_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING41 field descriptions

Field	Description
31–24 GPIO1_TRIG_ EN_RX_LO	Deassertion time setting for GPIO1_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO1_TRIG_EN signal or group will transition from HI to LO.
23–16 GPIO1_TRIG_ EN_RX_HI	Assertion time setting for GPIO1_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO1_TRIG_EN signal or group will transition from LO to HI.
15–8 GPIO1_TRIG_ EN_TX_LO	Deassertion time setting for GPIO1_TRIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO1_TRIG_EN signal or group will transition from HI to LO.
GPIO1_TRIG_ EN_TX_HI	Assertion time setting for GPIO1_TRIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO1_TRIG_EN signal or group will transition from LO to HI.

52.1.135 TSM_TIMING42 (XCVR_TSM_TIMING42)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the GPIO2_TRIG_EN TSM signal or signal group.

Address: 4005_C000h base + 390h offset = 4005_C390h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO2_TRIG_EN_RX_LO								GPIO2_TRIG_EN_RX_HI								GPIO2_TRIG_EN_TX_LO								GPIO2_TRIG_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING42 field descriptions

Field	Description
31–24 GPIO2_TRIG_ EN_RX_LO	Deassertion time setting for GPIO2_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO2_TRIG_EN signal or group will transition from HI to LO.
23–16 GPIO2_TRIG_ EN_RX_HI	Assertion time setting for GPIO2_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO2_TRIG_EN signal or group will transition from LO to HI.
15–8 GPIO2_TRIG_ EN_TX_LO	Deassertion time setting for GPIO2_TRIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO2_TRIG_EN signal or group will transition from HI to LO.
GPIO2_TRIG_ EN_TX_HI	Assertion time setting for GPIO2_TRIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO2_TRIG_EN signal or group will transition from LO to HI.

52.1.136 TSM_TIMING43 (XCVR_TSM_TIMING43)

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the GPIO3_TRIG_EN TSM signal or signal group.

Address: 4005_C000h base + 394h offset = 4005_C394h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO3_TRIG_EN_RX_LO								GPIO3_TRIG_EN_RX_HI								GPIO3_TRIG_EN_TX_LO								GPIO3_TRIG_EN_TX_HI							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XCVR_TSM_TIMING43 field descriptions

Field	Description
31–24 GPIO3_TRIG_EN_RX_LO	Deassertion time setting for GPIO3_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from HI to LO.
23–16 GPIO3_TRIG_EN_RX_HI	Assertion time setting for GPIO3_TRIG_EN signal or group RX sequence. This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from LO to HI.
15–8 GPIO3_TRIG_EN_TX_LO	Deassertion time setting for GPIO3_TRIG_EN signal or group TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from HI to LO.
GPIO3_TRIG_EN_TX_HI	Assertion time setting for GPIO3_TRIG_EN TX sequence. This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from LO to HI.

52.1.137 CORR_CTRL (XCVR_CORR_CTRL)

Address: 4005_C000h base + 3C0h offset = 4005_C3C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX_MAX_PREAMBLE								RX_MAX_CORR							
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MAX_CORR_EN	CORR_NVAL			CORR_VT							
W																
Reset	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0

* Notes:
- x = Undefined at reset.

XCVR_CORR_CTRL field descriptions

Field	Description
31–24 RX_MAX_ PREAMBLE	RX_MAX_PREAMBLE Max correlator during preamble-- max correlator value found during the preamble.
23–16 RX_MAX_CORR	RX_MAX_CORR Max correlator after preamble-- max correlator value found in packet after the preamble (refreshed every symbol rate if MAX_CORR_EN=1).
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 MAX_CORR_EN	MAX_CORR_EN Max correlator after preamble enable-- Enable the refresh of the max corr register
10–8 CORR_NVAL	CORR_NVAL Number of consecutively detected zero-symbols required to declare a preamble detected
CORR_VT	CORR_VT Correlator threshold, defines the sensitivity of demod during the preamble search state

52.1.138 PN_TYPE (XCVR_PN_TYPE)

Address: 4005_C000h base + 3C4h offset = 4005_C3C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														TX_ INV	PN_ TYPE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

XCVR_PN_TYPE field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 TX_INV	TX_INV test mode to invert the transmission
0 PN_TYPE	PN_TYPE PN Type - Pseudo Noise Chip Code Type (ZigBee=1)

52.1.139 PN_CODE (XCVR_PN_CODE)

Pseudo Noise Chip Code Seed Value

Address: 4005_C000h base + 3C8h offset = 4005_C3C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PN_MSB																PN_LSB															
Reset	0	1	1	1	0	1	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1	0	1	1

XCVR_PN_CODE field descriptions

Field	Description
31–16 PN_MSB	PN_MSB PN_CODE MS half
PN_LSB	PN_LSB PN_CODE LS half

52.1.140 Sync Control (XCVR_SYNC_CTRL)

Address: 4005_C000h base + 3CCh offset = 4005_C3CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TRACK_ENABLE	SYNC_PER		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

XCVR_SYNC_CTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TRACK_ENABLE	TRACK_ENABLE

Table continues on the next page...

XCVR_SYNC_CTRL field descriptions (continued)

Field	Description
	0 symbol timing synchronization tracking disabled in Rx frontend 1 symbol timing synchronization tracking enabled in Rx frontend (default)
SYNC_PER	Symbol Sync Tracking Period determines update rate for symbol timing, per equation. An early/late measurement is made every 2^SYNC_PER[2:0] symbols. Valid range of SYNC_PER[2:0] is 0 to 4.

52.1.141 SNF_THR (XCVR_SNF_THR)

Address: 4005_C000h base + 3D0h offset = 4005_C3D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								SNF_THR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_SNF_THR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SNF_THR	SNIFF Mode Threshold RSSI level at which the symbol demodulator will be started when SNF_CTRL[SNF_EN]=1. The control bit SNF_CTRL[SNF_EN] is in Zigbee address space. **Note:** SNIFF Mode not currently supported. SNF_THR has no effect

52.1.142 FAD_THR (XCVR_FAD_THR)

Address: 4005_C000h base + 3D4h offset = 4005_C3D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								FAD_THR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

XCVR_FAD_THR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FAD_THR	FAD_THR Correlator threshold at which the FAD will select the antenna.

52.1.143 ZBDEM_AFC (XCVR_ZBDEM_AFC)

Address: 4005_C000h base + 3D8h offset = 4005_C3D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			AFC_OUT					0						DCD_EN	AFC_EN
W																
Reset	0	0	0	x*	x*	x*	x*	x*	0	0	0	0	0	0	0	1

* Notes:

- x = Undefined at reset.

XCVR_ZBDEM_AFC field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 AFC_OUT	AFC_OUT AFC Result, Signed Two's Complement
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DCD_EN	DCD Mode Enable 0 NCD Mode (default) 1 DCD Mode
0 AFC_EN	AFC_EN Enable the AFC Function

52.1.144 LPPS Control Register (XCVR_LPPS_CTRL)

Address: 4005_C000h base + 3DCh offset = 4005_C3DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								LPPS_TCA_ALLOW	LPPS_BBF_ALLOW	LPPS_ADC_DAC_ALLOW	LPPS_ADC_I_Q_ALLOW	LPPS_ADC_CLK_ALLOW	LPPS_ADC_ALLOW	LPPS_QGEN25_ALLOW	LPPS_ENABLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_LPPS_CTRL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 LPPS_TCA_ALLOW	LPPS_TCA_ALLOW 0 Disallow TCA-related TSM outputs {**tca_en**, **tza_i_en**, **tza_q_en**, **tza_pdet_en**, **tza_dcoc_en**} to be duty-cycled during LPPS. 1 Allow TCA-related TSM outputs {**tca_en**, **tza_i_en**, **tza_q_en**, **tza_pdet_en**, **tza_dcoc_en**} to be duty-cycled during LPPS.
6 LPPS_BBF_ALLOW	LPPS_BBF_ALLOW 0 Disallow BBF-related TSM outputs {**bbf_i_en**, **bbf_q_en**, **bbf_pdet_en**, **bbf_dcoc_en**} to be duty-cycled during LPPS. 1 Allow BBF-related TSM outputs {**bbf_i_en**, **bbf_q_en**, **bbf_pdet_en**, **bbf_dcoc_en**} to be duty-cycled during LPPS.
5 LPPS_ADC_DAC_ALLOW	LPPS_ADC_DAC_ALLOW 0 Disallow ADC_DAC-related TSM outputs {**adc_dac1_en**, **adc_dac2_en**} to be duty-cycled during LPPS. 1 Allow ADC_DAC-related TSM outputs {**adc_dac1_en**, **adc_dac2_en**} to be duty-cycled during LPPS.
4 LPPS_ADC_I_Q_ALLOW	LPPS_ADC_I_Q_ALLOW 0 Disallow ADC_I/Q-related TSM outputs {**adc_I_adc_en**, **adc_Q_adc_en**} to be duty-cycled during LPPS. 1 Allow ADC_I/Q-related TSM outputs {**adc_I_adc_en**, **adc_Q_adc_en**} to be duty-cycled during LPPS.

Table continues on the next page...

XCVR_LPPS_CTRL field descriptions (continued)

Field	Description
3 LPPS_ADC_ CLK_ALLOW	LPPS_ADC_CLK_ALLOW 0 Disallow ADC-related TSM outputs {**xtal_adc_ref_clk_en**, **adc_clk_en**} to be duty-cycled during LPPS. 1 Allow ADC_CLK-related TSM outputs {**xtal_adc_ref_clk_en**, **adc_clk_en**} to be duty-cycled during LPPS.
2 LPPS_ADC_ ALLOW	LPPS_ADC_ALLOW 0 Disallow ADC-related TSM outputs {**adc_en**, **adc_bias_en**} to be duty-cycled during LPPS. 1 Allow ADC-related TSM outputs {**adc_en**, **adc_bias_en**} to be duty-cycled during LPPS.
1 LPPS_QGEN25_ ALLOW	LPPS_QGEN25_ALLOW 0 Disallow TSM output **qgen25_en** to be duty-cycled during LPPS 1 Allow TSM output **qgen25_en** to be duty-cycled during LPPS
0 LPPS_ENABLE	LPPS Mode Enable Master enable for LPPS mode. Allows Zigbee correlators to be duty-cycled during Preamble Search, and selected RF/Analog blocks to be duty-cycled simultaneously. 0 LPPS mode disabled 1 LPPS mode enabled

52.1.145 ADC Control (XCVR_ADC_CTRL)

Address: 4005_C000h base + 400h offset = 4005_C400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADC_COMP_ON															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ADC_TEST_ON	ADC_DITHER_ON	0						ADC_2X_CLK_SEL	0	ADC_32MHZ_SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

XCVR_ADC_CTRL field descriptions

Field	Description
31–16 ADC_COMP_ON	ADC Comparator Enable

Table continues on the next page...

XCVR_ADC_CTRL field descriptions (continued)

Field	Description
	These bits enable or disable the individual comparators in the quantizer. In the normal mode all these bits should be enabled. The disables are used for testing purposes only
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 ADC_TEST_ON	ADC Test On When enabled (1) this bit puts the adc in test mode where different test signals can be injected or measured. In normal mode this bit is diabled(0).
9 ADC_DITHER_ ON	ADC Dither On Enables the dither circuit inside the ADC block. By enabling this bit dithering of tones can be achieved.
8–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 ADC_2X_CLK_ SEL	ADC_2X_CLK_SEL Select 2x Clock option in the ADC. When this bit is enabled a clock of 64Mhz is assumed. This option is not used in this version and reserved for future use.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 ADC_32MHZ_ SEL	ADC 32MHZ Clock Select Select settings for a 32MHz reference clock. If this bit is not set then settings for a 36MHz clock is chosen. Only the 32Mhz option is enabled in this version.

52.1.146 ADC Tuning (XCVR_ADC_TUNE)

Address: 4005_C000h base + 404h offset = 4005_C404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								ADC_C2_TUNE				ADC_C1_TUNE			
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ADC_R2_TUNE			0	ADC_R1_TUNE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

XCVR_ADC_TUNE field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–20 ADC_C2_TUNE	ADC_C2_TUNE Allows to tune the capacitor values of the second integrator ± 15%. Default setting is 0b1000.

Table continues on the next page...

XCVR_ADC_TUNE field descriptions (continued)

Field	Description
19–16 ADC_C1_TUNE	ADC_C1_TUNE Allows to tune the capacitor values of the first integrator ± 15%. Default setting is 0b1000.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 ADC_R2_TUNE	ADC_R2_TUNE Allows to tune the resistor values of the second integrator ± 15%. Default setting is 0b011.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADC_R1_TUNE	ADC_R1_TUNE Allows to tune the resistor values of the first integrator ± 15%. Default setting is 0b011.

52.1.147 ADC Adjustment (XCVR_ADC_ADJ)

Address: 4005_C000h base + 408h offset = 4005_C408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	ADC_FLSH_RES_ADJ			0	ADC_IB_FLSH_ADJ			0					ADC_IB_DAC2_ADJ		
W																
Reset	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ADC_IB_DAC1_ADJ			0					ADC_IB_OPAMP2_ADJ			0	ADC_IB_OPAMP1_ADJ		
W																
Reset	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1

XCVR_ADC_ADJ field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–28 ADC_FLSH_RES_ADJ	ADC_FLSH_RES_ADJ Allows to adjust the value of the resistor ladder that is used to generate the reference voltages for the quantizer. The defualt value is 100
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 ADC_IB_FLSH_ADJ	ADC_IB_FLSH_ADJ Adjusts the quantizer preamplifier current plus/minus 30 percent. Default setting is 0b011.
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 ADC_IB_DAC2_ADJ	ADC_IB_DAC2_ADJ Adjusts the DAC 2 current ±30%. Default setting is 0b011.

Table continues on the next page...

XCVR_ADC_ADJ field descriptions (continued)

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–12 ADC_IB_DAC1_ADJ	ADC_IB_DAC1_ADJ Adjusts the DAC 1 current ±30%. Default setting is 0b011.
11–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 ADC_IB_OPAMP2_ADJ	ADC_IB_OPAMP2_ADJ Adjusts the 2nd integrator operational amplifier reference current ±30%. Default setting is 0b011.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADC_IB_OPAMP1_ADJ	ADC_IB_OPAMP1_ADJ Adjusts the 1st integrator operational amplifier reference current ±30%. Default setting is 0b011.

52.1.148 ADC Regulators (XCVR_ADC_REGS)

Address: 4005_C000h base + 40Ch offset = 4005_C40Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														ADC_INTERNAL_IREF_BYPASS_ON	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADC_VCMREF_BYPASS_ON	0					ADC_DIG_REG_BYPASS_ON	ADC_ANA_REG_BYPASS_ON	ADC_REG_DIG_SUPPLY				ADC_ANA_REG_SUPPLY			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_ADC_REGS field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 ADC_ INTERNAL_ IREF_BYPASS_ ON	ADC_INTERNAL_IREF_BYPASS_ON Bypass the internally generated 5uA reference current that is used for the ADC to generate all the other reference currents for the ADC and use an external reference current. This mode should be used only in combination with analog test mode for the ADC enabled and when supplying an external current reference through the test mux. This function is not used in this version of the chip
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 ADC_VCMREF_ BYPASS_ON	ADC_VCMREF_BYPASS_ON Allows to use an external 0.6V reference instead of the internally generated 0.6V. This mode should be used only in combination with analog test mode for the ADC enabled and when supplying an external 0.6V reference voltage through the test mux.This function is not used in this version of the chip
14–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 ADC_DIG_REG_ BYPASS_ON	ADC_DIG_REG_BYPASS_ON This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.
8 ADC_ANA_ REG_BYPASS_ ON	ADC_ANA_REG_BYPASS_ON This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.
7–4 ADC_REG_DIG_ SUPPLY	ADC_REG_DIG_SUPPLY Regulator trim bits to change the output voltage from 1.05 to 1.4V 0 1.2V 1 1.05V 2 1.075V 3 1.1V 4 1.125V 5 1.15V 6 1.175V 7 1.2V 8 1.225V 9 1.25V 10 1.275V 11 1.3V 12 1.325V 13 1.35V 14 1.375V 15 1.4V
ADC_ANA_ REG_SUPPLY	ADC_ANA_REG_SUPPLY Regulator trim bits to change the output voltage from 1.05 to 1.4V 0 1.2V

Table continues on the next page...

XCVR_ADC_REGS field descriptions (continued)

Field	Description
	1 1.05V 2 1.075V 3 1.1V 4 1.125V 5 1.15V 6 1.175V 7 1.2V 8 1.225V 9 1.25V 10 1.275V 11 1.3V 12 1.325V 13 1.35V 14 1.375V 15 1.4V

52.1.149 ADC Regulator Trims (XCVR_ADC_TRIMS)

Address: 4005_C000h base + 410h offset = 4005_C410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ADC_VCM_TRIM			0	ADC_IREF_FLSH_RES_TRIM			0	ADC_IREF_OPAMPS_RES_TRIM		
W																
Reset	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0

XCVR_ADC_TRIMS field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 ADC_VCM_TRIM	ADC_VCM_TRIM This allows to trim the resistor value used to generate the reference current for the DACs. These bits are mislabeled and should be called ADC_IREF_DAC_RES_TRIM. The resistor values can be trimmed by ±15%.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XCVR_ADC_TRIMS field descriptions (continued)

Field	Description
6–4 ADC_IREF_ FLSH_RES_ TRIM	ADC_IREF_FLSH_RES_TRIM This allows to trim the resistor value used to generate the reference current for the quantizer. The resistor values can be trimmed by ±15%.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADC_IREF_ OPAMPS_RES_ TRIM	ADC_IREF_OPAMPS_RES_TRIM This allows to trim the resistor value used to generate the reference current for the integrator's operational amplifiers. The resistor values can be trimmed by plus/minus 15 percent.

52.1.150 ADC Test Control (XCVR_ADC_TEST_CTRL)

Address: 4005_C000h base + 414h offset = 4005_C414h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				ADC_SPARE3	DCOC_ALPHA_ RADIUS_GS_IDX			0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ADC_ANA_ REG_ATST_ SEL		0		ADC_DIG_ REG_ATST_ SEL		0			ADC_ATST_SEL				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_ADC_TEST_CTRL field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ADC_SPARE3	ADC_SPARE3 Spare Bit for future use
26–24 DCOC_ALPHA_ RADIUS_GS_ IDX	Alpha-R Scaling DCOC Alpha-R Scaling. This has the same definition as DCOC_ALPHA_RADIUS_IDX. The tracking estimator will switch from DCOC_ALPHA_RADIUS_IDX to DCOC_ALPHA_RADIUS_GS_IDX at the time specified by the DCOC_TRK_EST_GS_CNT. 000 1 001 1/2

Table continues on the next page...

XCVR_ADC_TEST_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>010 1/4
011 1/8
100 1/16
101 1/32
110 1/64
111 Reserved</td></tr>
<tr><td>23–14
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>13–12
ADC_ANA_
REG_ATST_SEL</td><td>ADC_ANA_REG_ATST_SEL
These bits control what internal signals are connected to the ATST bus.
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table></td></tr>
<tr><td>11–10
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>9–8
ADC_DIG_REG_
ATST_SEL</td><td>ADC_DIG_REG_ATST_SEL
These bits control the what internal regualtor signals are connected to the ATST bus.
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table></td></tr>
<tr><td>7–5
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>ADC_ATST_SEL</td><td>ADC Analog Test Selection
Allows to slect between the different ADC analog test modes.All other values are reserved.
0 Inject 5uA refrence current on ATST0 ,Inject 0.6V reference voltage on ATST1
1 Monitor Flash refrence currents on ATST3
2 Monitor DAC refrence current on ATST0,Monitor mirrored reference current at ATST1,Monitor operational amplifiers reference current at ATST2, Monitor buffered 0.6V reference voltage used for opamp1 common mode at ATST3
3 Monitored buffered 0.6V reference voltrage used for opamp2 common mode at ATST0 monitor buffered 0.6V reference voltage used for opam3 comon mode at ATST2. However opamp3 does not exisit in this silicon but there is still a buffered reference available.</td></tr>
</table>

52.1.151 Baseband Filter Control (XCVR_BBF_CTRL)

Address: 4005_C000h base + 420h offset = 4005_C420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BBF_SPARE_3_2		DCOC_ALPHAC_SCALE_GS_IDX		BBF_TMUX_ON	0	BBF_DCOC_ON	BBF_CUR_CNTL	BBF_RES_TUNE2				BBF_CAP_TUNE			
W																
Reset	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1

XCVR_BBF_CTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–14 BBF_SPARE_3_2	BBF_SPARE_3_2 Spare Bits for future use
13–12 DCOC_ALPHAC_SCALE_GS_IDX	DCOC Alpha-C Scaling DCOC Alpha-C Scaling. This has the same definition as DCOC_ALPHAC_SCALE_IDX. The tracking estimator will switch from DCOC_ALPHAC_SCALE_IDX to DCOC_ALPHAC_SCALE_GS_IDX at the time specified by the DCOC_TRK_EST_GS_CNT. 00 1/2 01 1/4 10 1/8 11 1/16
11 BBF_TMUX_ON	BBF_TMUX_ON This bit enables the test mode for the baseband filter block. The internal signals are brought to the the ATST bus based on the values of the rx_atst_sel. Refer to the description in the rx_atst_sel field.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 BBF_DCOC_ON	BBF_DCOC_ON Not currently connected. Was intended to enable the DCOC at the output of BBF
8 BBF_CUR_CNTL	BBF_CUR_CNTL This bit controls the current in the BBF and default is 1. When the bit is unset the total BBF current decreases by 350uA.

Table continues on the next page...

XCVR_BBF_CTRL field descriptions (continued)

Field	Description
	0 Low current setting. 1 High current setting.
7–4 BBF_RES_ TUNE2	BBF_RES_TUNE2 This is the programmable resistor to change the corner frquency of the passive pole. The total f3db filter corner is determined by bbf_cap_tune and tza_cap_tune.See the description for bbf_cap_tune BBF_CAP_TUNE for the corner frequency values.
BBF_CAP_TUNE	BBF_CAP_TUNE This is programmbale capacitor values to change the corner frequency of the first baseband filter. This bit in combination with bbf_res_tune and tza_cap_tune determine the baseband filter response. The following table illustrates the composite filter responces based on the bit settings

tza_cap	bba_cap	bba_res2	f3db(KHz)
0	0	0	1180
0	0	2	1077
0	0	5	1007
0	0	7	873.1
0	2	0	1119
0	2	2	1030
0	2	5	968
0	2	7	846.2
0	5	0	980.5
0	5	2	917.9
0	5	5	872.5
0	5	7	777.6
0	7	0	836.8
0	7	2	795.8
0	7	5	764.3
0	7	7	695.3
2	0	0	1096
2	0	2	1009
2	0	5	949.8
2	0	7	832.2
2	2	0	1044
2	2	2	968.6
2	2	5	915.7
2	2	7	808.3
2	5	0	924.9
2	5	2	871.2
2	5	5	831.4
2	5	7	746.8

Table continues on the next page...

XCVR_BBF_CTRL field descriptions (continued)

Field	Description			
	tza_cap	bba_cap	bba_res2	f3db(KHz)
	2	7	0	798.2
	2	7	2	762.2
	2	7	5	734.2
	2	7	7	671.9
	5	0	0	934.5
	5	0	2	875.7
	5	0	5	833.6
	5	0	7	745.9
	5	2	0	898.9
	5	2	2	846.7
	5	2	5	808.5
	5	2	7	727.9
	5	5	0	813.3
	5	5	2	774.6
	5	5	5	745
	5	5	7	680
	5	7	0	717.4
	5	7	2	690.4
	5	7	5	669
	5	7	7	619.7
	7	0	0	788.5
	7	0	2	750.1
	7	0	5	721.3
	7	0	7	658.5
	7	2	0	765
	7	2	2	730.2
	7	2	5	703.7
	7	2	7	645.2
	7	5	0	705.7
	7	5	2	678.8
	7	5	5	657.7
	7	5	7	609.4
	7	7	0	635.6
	7	7	2	616
	7	7	5	600.2
	7	7	7	562.6

52.1.152 RX Analog Control (XCVR_RX_ANA_CTRL)

Address: 4005_C000h base + 42Ch offset = 4005_C42Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								LNM_SPARE_3_2_1			IQMC_DC_GAIN_ADJ_EN	RX_ATST_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_RX_ANA_CTRL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 LNM_SPARE_3_2_1	LNM_SPARE_3_2_1 Spare Bits for future use
4 IQMC_DC_GAIN_ADJ_EN	IQMC_DC_GAIN_ADJ_EN If set, the I/Q mismatch uses IQMC_DC_GAIN_ADJ during DCOC calibration. If clear, the I/Q mismatch uses IQMC_GAIN_ADJ.
RX_ATST_SEL	RX_ATST_SEL These bits select the different internal baseband signals to be made available on the ATST bus. The following table indicates the specific connection. All other combinations are not defined. The bit bbf_tmux_on BBF_TMUX_ON needs to be enabled for the test mode to work.

rx_atst_sel	ATST0	ATST1	ATST2	ATST3
0000	half supply voltage	bbf opamp common mode voltatge	bbf_I_out	bbf_I_ouxt
0001	bba_dcoc_I	bba_dcoc_Ix	bba_dcoc_Q	bba_dcoc_Qx
0010	tza_out_I	tza_out_Ix	tza_out_Q	tza_out_Qx
0011	peak det ref hi	peak det ref lo	peak det bias check	tza common mode
0100	bbf_out_I	bbf_out_Ix	bbf_out_Q	bbf_out_Qx
0101	tza_dcoc_I	tza_dcoc_Ix	tza_dcoc_Q	tza_dcoc_Qx

Table continues on the next page...

XCVR_RX_ANA_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>rx_atst_sel</th><th>ATST0</th><th>ATST1</th><th>ATST2</th><th>ATST3</th></tr>
<tr><td>0101</td><td>tza_in_I</td><td>tza_in_Ix</td><td>tza_out_I</td><td>tza_out_Ix</td></tr>
</table>
</td></tr>
</table>

52.1.153 Crystal Oscillator Control Register 1 (XCVR_XTAL_CTRL)

Address: 4005_C000h base + 434h offset = 4005_C434h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	XTAL_READY	0		XTAL_COMP_BIAS_HI					XTAL_ALC_ON	XTAL_ALC_START_512U	0	XTAL_COMP_BIAS_LO				
W																
Reset	x*	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	XTAL_READY_COUNT_SEL		XTAL_BYPASS	XTAL_GM					XTAL_TRIM							
Reset	1	1	0	0	0	0	0	1	0	1	1	1	0	1	1	1

* Notes:
- x = Undefined at reset.

XCVR_XTAL_CTRL field descriptions

Field	Description
31 XTAL_READY	XTAL Ready Indicator

Table continues on the next page...

XCVR_XTAL_CTRL field descriptions (continued)

Field	Description
	The signal goes high after a number of cycles determined by xtal_ready_count_sel
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 XTAL_COMP_BIAS_HI	XTAL_COMP_BIAS (High) These bits used to adjust the bias of the crystal Comparator when the transceiver is ACTIVE
23 XTAL_ALC_ON	XTAL_ALC_ON enable the ALC for the xtal
22 XTAL_ALC_START_512U	XTAL_ALC_START_512U 0 Start XTAL ALC at 256usec 1 Start XTAL ALC at 512usec
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 XTAL_COMP_BIAS_LO	XTAL_COMP_BIAS (Low) These bits used to adjust the bias of the crystal Comparator when the transceiver is IDLE
15–14 XTAL_READY_COUNT_SEL	XTAL Ready Count Select This selects the number of count cycles before xtal_ready goes high 00 1024 clock cycles 01 2048 clock cycles 10 4096 clock cycles 11 8192 clock cycles
13 XTAL_BYPASS	XTAL Bypass When this bit is set, the Crystal Oscillator is disabled and an external clock signal applied on the EXTAL pin is selected as the clock source.
12–8 XTAL_GM	XTAL_GM This is used adjust the gm of the Crystal core. All 0's is minimum gm and all 1's is maximum gm
XTAL_TRIM	XTAL Trim Program the internal capacitor banks to trim the 32M Crytstal frequency. It has a trim range of ~2Khz

52.1.154 Crystal Oscillator Control Register 2 (XCVR_XTAL_CTRL2)

Address: 4005_C000h base + 438h offset = 4005_C438h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	XTAL_SPARE				0	XTAL_ATST_ON	XTAL_ATST_SEL		0						XTAL_REG_ATST_SEL	
W					▨				▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			XTAL_DIG_CLK_OUT_ON	XTAL_ON_OVRD	XTAL_ON_OVRD_ON	XTAL_REG_ON_OVRD	XTAL_REG_ON_OVRD_ON	0			XTAL_REG_BYPASS_ON	XTAL_REG_SUPPLY			
W	▨								▨							
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_XTAL_CTRL2 field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–28
XTAL_SPARE</td><td>XTAL_SPARE

Spare Bits for future use</td></tr>
<tr><td>27
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>26
XTAL_ATST_ON</td><td>XTAL_ATST_ON

This is the test mode for the xtal block. When this bit is set 32M clock is put on ATST2</td></tr>
<tr><td>25–24
XTAL_ATST_SEL</td><td>XTAL_ATST_SEL

These bits are not used in this version of the chip. Since only one signal is brought out for test purposes, xtal_atst_on is used as the enable and sel signal</td></tr>
<tr><td>23–18
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>17–16
XTAL_REG_ATST_SEL</td><td>XTAL_REG_ATST_SEL

These bits control the what internal regualtor signals are connected to the ATST bus.
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

XCVR_XTAL_CTRL2 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td><table><tr><th>register setting</th><th>ATST0</th></tr><tr><td>11</td><td>vbias</td></tr></table></td></tr>
<tr><td>15–13
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>12
XTAL_DIG_CLK_OUT_ON</td><td>XTAL_DIG_CLK_OUT_ON
This bit gates the Crystal clock output</td></tr>
<tr><td>11
XTAL_ON_OVRD</td><td>XTAL_ON_OVRD
enable for the xtal in test mode (XTAL_ON_OVRD_ON=1)</td></tr>
<tr><td>10
XTAL_ON_OVRD_ON</td><td>XTAL_ON_OVRD_ON
mux select for the crystal enable between normal operation (0) and test mode (1)</td></tr>
<tr><td>9
XTAL_REG_ON_OVRD</td><td>XTAL_REG_ON_OVRD
Enable for the xtal regulator in test mode (XTAL_REG_ON_OVRD_ON=1)</td></tr>
<tr><td>8
XTAL_REG_ON_OVRD_ON</td><td>XTAL_REG_ON_OVRD_ON
Mux select for the crystal regulator enable between normal operation (0) and test mode (1). The test signal is called xtal_reg_on_ovrd</td></tr>
<tr><td>7–5
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>4
XTAL_REG_BYPASS_ON</td><td>XTAL_REG_BYPASS_ON
This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.</td></tr>
<tr><td>XTAL_REG_SUPPLY</td><td>XTAL_REG_SUPPLY
Regulator trim bit to change the outputvoltage from 1.05 to 1.4V<table>
<tr><td>0</td><td>1.2V</td></tr>
<tr><td>1</td><td>1.05V</td></tr>
<tr><td>2</td><td>1.075V</td></tr>
<tr><td>3</td><td>1.1V</td></tr>
<tr><td>4</td><td>1.125V</td></tr>
<tr><td>5</td><td>1.15V</td></tr>
<tr><td>6</td><td>1.175V</td></tr>
<tr><td>7</td><td>1.2V</td></tr>
<tr><td>8</td><td>1.225V</td></tr>
<tr><td>9</td><td>1.25V</td></tr>
<tr><td>10</td><td>1.275V</td></tr>
<tr><td>11</td><td>1.3V</td></tr>
<tr><td>12</td><td>1.325V</td></tr>
<tr><td>13</td><td>1.35V</td></tr>
<tr><td>14</td><td>1.375V</td></tr>
<tr><td>15</td><td>1.4V</td></tr>
</table></td></tr>
</table>

52.1.155 Bandgap Control (XCVR_BGAP_CTRL)

Address: 4005_C000h base + 43Ch offset = 4005_C43Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			BGAP_ATST_ON	BGAP_ATST_SEL				BGAP_VOLTAGE_TRIM				BGAP_CURRENT_TRIM			
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

XCVR_BGAP_CTRL field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–13
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>12
BGAP_ATST_ON</td><td>BGAP_ATST_ON

This bit enables the test mux for the bangap block. The different internal nodes of bandgap are connected to the ATST bus based on bgap_atst_sel bits.</td></tr>
<tr><td>11–8
BGAP_ATST_SEL</td><td>BGAP_ATST_SEL

Select what internal signals to bring out to ATST bus.
<table>
<tr><th>bgap_atst_sel</th><th>ATST2</th><th>ATST3</th></tr>
<tr><td>0000</td><td>1ua</td><td>Unfiltered bgap Output</td></tr>
<tr><td>0001</td><td>2uA</td><td>Internal X1 node of Bgap</td></tr>
<tr><td>0010</td><td>5uA</td><td>Internal X2 node of Bgap</td></tr>
<tr><td>0011</td><td>10uA</td><td>No Connect</td></tr>
<tr><td>0100</td><td>1uA ptat</td><td>No Connect</td></tr>
</table></td></tr>
<tr><td>7–4
BGAP_VOLTAGE_TRIM</td><td>BGAP_VOLTAGE_TRIM

Trim the bandgap voltage to 1V in 4mV steps</td></tr>
<tr><td>BGAP_CURRENT_TRIM</td><td>BGAP_CURRENT_TRIM

Trim the 1uA bandgap current</td></tr>
</table>

52.1.156 PLL Control Register (XCVR_PLL_CTRL)

Address: 4005_C000h base + 444h offset = 4005_C444h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL_VCO_SPARE7	HPM_BIAS							0						PLL_VCO_LDO_BYPASS	PLL_REG_BYPASS_ON
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PLL_REG_SUPPLY				0	PLL_LFILT_CNTL			0	PLL_VCO_BIAS		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

XCVR_PLL_CTRL field descriptions

Field	Description
31 PLL_VCO_SPARE7	PLL_VCO_SPARE7 Spare Bit for future use
30–24 HPM_BIAS	HPM Array Bias Provides a (-64/+63 x 976.56 Hz) range of steps to adjust the HPM Array Mid-Point during modulation
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 PLL_VCO_LDO_BYPASS	PLL_VCO_LDO_BYPASS This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.
16 PLL_REG_BYPASS_ON	PLL_REG_BYPASS_ON This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 PLL_REG_SUPPLY	PLL_REG_SUPPLY Regulator trim bit to change the outputvoltage from 1.05 to 1.4V 0 1.2V 1 1.05V 2 1.075V

Table continues on the next page...

XCVR_PLL_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>3</td><td>1.1V</td></tr>
<tr><td>4</td><td>1.125V</td></tr>
<tr><td>5</td><td>1.15V</td></tr>
<tr><td>6</td><td>1.175V</td></tr>
<tr><td>7</td><td>1.2V</td></tr>
<tr><td>8</td><td>1.225V</td></tr>
<tr><td>9</td><td>1.25V</td></tr>
<tr><td>10</td><td>1.275V</td></tr>
<tr><td>11</td><td>1.3V</td></tr>
<tr><td>12</td><td>1.325V</td></tr>
<tr><td>13</td><td>1.35V</td></tr>
<tr><td>14</td><td>1.375V</td></tr>
<tr><td>15</td><td>1.4V</td></tr>
</table>
</td></tr>
<tr><td>7
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>6–4
PLL_LFILT_
CNTL</td><td>PLL Loop Filter Control
Program the resistors in the loop filter to control the bandwith.
<table>
<tr><th>Pll_lfilt_cntl</th><th>First Pole Resistor</th><th>Second Pole Resistor</th></tr>
<tr><td>000</td><td>25.6K</td><td>26K</td></tr>
<tr><td>001</td><td>20.8K</td><td>26K</td></tr>
<tr><td>010</td><td>15.4K</td><td>26K</td></tr>
<tr><td>011</td><td>10.3K</td><td>26K</td></tr>
<tr><td>100</td><td>25.6K</td><td>100</td></tr>
<tr><td>101</td><td>20.8K</td><td>100</td></tr>
<tr><td>110</td><td>15.4K</td><td>100</td></tr>
<tr><td>111</td><td>10.3K</td><td>100</td></tr>
</table>
</td></tr>
<tr><td>3
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>PLL_VCO_BIAS</td><td>PLL VCO Bias Control
Progammable current control for the VCO. The VCO current varies from 1.5mA to 2.1mA</td></tr>
</table>

52.1.157 PLL Control Register 2 (XCVR_PLL_CTRL2)

Address: 4005_C000h base + 448h offset = 4005_C448h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							PLL_TMUX_ON	0		PLL_VCO_ REG_ SUPPLY		PLL_KMOD_ SLOPE	PLL_VCO_KV		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

XCVR_PLL_CTRL2 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 PLL_TMUX_ON	PLL_TMUX_ON This bit enables the testmux inside the PLL and the different internal nodes are connected to ATST bus based on the pll_tmux_sel settings
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 PLL_VCO_REG_ SUPPLY	PLL_VCO_REG_SUPPLY Regulator trim bits to change the outputvoltage from 1.15 to 1.3V 0 1.15V 1 1.2V 2 1.25V 3 1.3V
3 PLL_KMOD_ SLOPE	PLL_KMOD_SLOPE This bit controls the slope of the highport capacitor bank. When this bit is set the Kmod slope changes from 10Khz to 15Khz
PLL_VCO_KV	PLL_VCO_KV These bits control the gain of the VCO. This is an additioanl knob to control the loop bandwitch of the PLL. All 0's correspond to minimum KV and all 1's correspond to max KV programmable

52.1.158 PLL Test Control (XCVR_PLL_TEST_CTRL)

Address: 4005_C000h base + 44Ch offset = 4005_C44Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	PLL_RIPPLE_COUNTER_TEST_MODE	PLL_FORCE_VTUNE_EXTERNALLY	PLL_VCO_TEST_CLK_MODE	0		PLL_REG_ATST_SEL		0		PLL_VCO_REG_ATST		0		PLL_TMUX_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_PLL_TEST_CTRL field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 PLL_RIPPLE_COUNTER_TEST_MODE	PLL_RIPPLE_COUNTER_TEST_MODE PLL Ripple Counter Test Mode
13 PLL_FORCE_VTUNE_EXTERNALLY	PLL_FORCE_VTUNE_EXTERNALLY Force VTUNE externally
12 PLL_VCO_TEST_CLK_MODE	PLL_VCO_TEST_CLK_MODE test mode for the VCO
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 PLL_REG_ATST_SEL	PLL_REG_ATST_SEL These bits control the what internal regualtor signals are connected to the ATST bus.

Table continues on the next page...

XCVR_PLL_TEST_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table>
</td></tr>
<tr><td>7–6
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>5–4
PLL_VCO_REG_ATST</td><td>PLL_VCO_REG_ATST
These bits determine what internal signals are connected to the ATST bus
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table>
</td></tr>
<tr><td>3–2
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>PLL_TMUX_SEL</td><td>PLL_TMUX_SEL
Select what internal signals to bring out to ATST pins. PLL_TMUX_ON needs to be set.
<table>
<tr><th>register setting</th><th>ATST0</th><th>ATST1</th><th>ATST2</th><th>ATST3</th></tr>
<tr><td>00</td><td>precharge_filt</td><td>xor_out</td><td>pll_ref_xtal</td><td>pll_ref_xtal_b</td></tr>
<tr><td>01</td><td>pll_sigma_delta_clk</td><td>pll_loop_div_count[0]</td><td>pll_loop_div_count[1]</td><td>pll_loop_div_count[2]</td></tr>
<tr><td>10</td><td>pll_loop_div_count[3]</td><td>pll_loop_div_count[4]</td><td>pll_loop_div_count[8]</td><td>No Connect</td></tr>
<tr><td>11</td><td>pll_ripple_counter_override_clk</td><td>No Connect</td><td>No Connect</td><td>No Connect</td></tr>
</table>
</td></tr>
</table>

52.1.159 QGEN Control (XCVR_QGEN_CTRL)

Address: 4005_C000h base + 458h offset = 4005_C458h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							QGEN_REG_ BYPASS_ON	QGEN_REG_ATST_SEL				QGEN_REG_SUPPLY			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_QGEN_CTRL field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–9
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>8
QGEN_REG_
BYPASS_ON</td><td>QGEN_REG_BYPASS_ON

This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.</td></tr>
<tr><td>7–4
QGEN_REG_
ATST_SEL</td><td>QGEN_REG_ATST_SEL

These bits control the what internal regualtor signals are connected to the ATST bus.
<table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table>
</td></tr>
<tr><td>QGEN_REG_
SUPPLY</td><td>QGEN_REG_SUPPLY

Regulator trim bits to change the output voltage from 1.05 to 1.4V

0 1.2V
1 1.05V
2 1.075V</td></tr>
</table>

Table continues on the next page...

XCVR_QGEN_CTRL field descriptions (continued)

Field	Description
	3 1.1V 4 1.125V 5 1.15V 6 1.175V 7 1.2V 8 1.225V 9 1.25V 10 1.275V 11 1.3V 12 1.325V 13 1.35V 14 1.375V 15 1.4V

52.1.160 TCA Control (XCVR_TCA_CTRL)

Address: 4005_C000h base + 464h offset = 4005_C464h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TCA_TX_REG_ATST_SEL		TCA_TX_REG_SUPPLY				TCA_TX_REG_BYPASS_ON	TCA_LOW_PWR_ON	TCA_BIAS_CURR	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TCA_CTRL field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 TCA_TX_REG_ATST_SEL	TCA_TX_REG_ATST_SEL These bits determine what internal signals to connect to ATST bus

Table continues on the next page...

XCVR_TCA_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td><table>
<tr><th>register setting</th><th>ATST0</th></tr>
<tr><td>00</td><td>No Connect</td></tr>
<tr><td>01</td><td>vout</td></tr>
<tr><td>10</td><td>vin feedback</td></tr>
<tr><td>11</td><td>vbias</td></tr>
</table></td></tr>
<tr><td>7–4
TCA_TX_REG_
SUPPLY</td><td>TCA_TX_REG_SUPPLY
Regulator trim bit to change the outputvoltage from 1.05 to 1.4V
0 1.2V
1 1.05V
2 1.075V
3 1.1V
4 1.125V
5 1.15V
6 1.175V
7 1.2V
8 1.225V
9 1.25V
10 1.275V
11 1.3V
12 1.325V
13 1.35V
14 1.375V
15 1.4V</td></tr>
<tr><td>3
TCA_TX_REG_
BYPASS_ON</td><td>TCA_TX_REG_BYPASS_ON
This register bit determines if the regulator is in bypass mode. When in bypass mode, the external voltage applied to the input of the regulator is presented at the regulator output.</td></tr>
<tr><td>2
TCA_LOW_
PWR_ON</td><td>TCA_LOW_PWR_ON
Enable the tca low power mode</td></tr>
<tr><td>TCA_BIAS_
CURR</td><td>TCA_BIAS_CURR
Programmable bias currrent for the TCA</td></tr>
</table>

52.1.161 TZA Control (XCVR_TZA_CTRL)

Address: 4005_C000h base + 468h offset = 4005_C468h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TZA_SPARE				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TZA_CUR_ CNTL		TZA_DCOC_ ON	TZA_ GAIN	TZA_CAP_TUNE			
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0

XCVR_TZA_CTRL field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–20 TZA_SPARE	TZA_SPARE Spare Bits for future use
19–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 TZA_CUR_CNTL	TZA_CUR_CNTL Program the current in TZA. The TZA current is varied from 510uA to 2.1mA
5 TZA_DCOC_ON	TZA_DCOC_ON Not currently connected. Was intended to enable the DCOC DAC at the output of the TZA
4 TZA_GAIN	TZA_GAIN Change the TZA gain. It is not used in this version of silicon.
TZA_CAP_TUNE	TZA_CAP_TUNE The bits sets the f3dB filter corner for the TZA block. This in combination with bbf_cap_tune and bbf_res_tune2 determine the baseband filter response.Look for the f3db corner here BBF_CAP_TUNE

52.1.162 TX Analog Control (XCVR_TX_ANA_CTRL)

Address: 4005_C000h base + 474h offset = 4005_C474h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												HPM_CAL_ADJUST			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_TX_ANA_CTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HPM_CAL_ ADJUST	HPM Cal Count Adjust Provides a (-8/+7) range of adjustment to the HPM Calibration Count Difference for the HPM Calibration lookup table

52.1.163 Analog Spare (XCVR_ANA_SPARE)

Address: 4005_C000h base + 47Ch offset = 4005_C47Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										ANA_DTEST					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HPM_LSB_INVERT		DCOC_TRK_EST_GS_CNT			IQMC_DC_GAIN_ADJ										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XCVR_ANA_SPARE field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 ANA_DTEST	ANA_DTEST Not currently implemented, reads back zero.
15–14 HPM_LSB_ INVERT	High port LSB array inversion control Provides individual inversion settings for the two HPM LSB Array Units

Table continues on the next page...

XCVR_ANA_SPARE field descriptions (continued)

Field	Description
13–11 DCOC_TRK_ EST_GS_CNT	DCOC Tracking Estimator Gearshift Count Indicates the number of tracking update corrections after an AGC gain change before the tracking estimator switches from parameters {dcoc_alpha_radius_idx, dcoc_alphac_scaling_idx, dcoc_sign_scaling_idx} to the set of gearshift parameters {dcoc_alpha_radius_gs_idx, dcoc_alphac_scaling_gs_idx, dcoc_sign_scaling_idx}. Note that dcoc_sign_scaling_idx is used in both configurations. 0 Only use {dcoc_alpha_radius_idx, dcoc_alphac_scaling_idx, dcoc_sign_scaling_idx} 1 Switch from {dcoc_alpha_radius_idx, dcoc_alphac_scaling_idx, dcoc_sign_scaling_idx} to {dcoc_alpha_radius_gs_idx, dcoc_alphac_scaling_gs_idx, dcoc_sign_scaling_idx} after the 1 update correction. 2 Switch after 2 update corrections. 3 Switch after 3 update corrections. 4 Switch after 4 update corrections. 5 Switch after 5 update corrections. 6 Switch after 6 update corrections. 7 Switch after 7 update corrections.
IQMC_DC_ GAIN_ADJ	IQ Mismatch Correction DC Gain Coeff I/Q mismatch correction DC gain coefficient. This is the value by which the Q channel data is multiplied during DCOC calibration if the IQMC_DC_GAIN_ADJ_EN bit is set; otherwise IQMC_GAIN_ADJ is used during DCOC calibration. Format is u1.10 so e.g. 11'h400 (the reset value) corresponds to a value of 1.0, 11'h200 corresponds to 0.5, 11'h600 corresponds to 1.5.

Chapter 53
BLE Link Layer Registers

53.1 Bluetooth Low Energy Memory map/register definition

This section describes the registers and data structures in the Bluetooth Link Layer module as instantiated in the **Radio**. The base address of the module is 0x4005B000. The addresses presented here are relative addresses which are relative to this base address.

Instruction Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x00	0x000	COMMAND_REGISTER	WO	Instructions register to send commands to link layer hardware for controlling hardware operations.	0xXX00

Field	Bit	Description	Reset
Reserved	15:8		XX
command	7:0	8-bit command from firmware to the link layer controller. See appendix 1 for the list of instructions and their opcodes. The instruction results in the link layer hardware starting/stopping an operation. Notes on use 1. Few of the commands will require other configuration registers to be set, before the command is written. Refer to	00

		appendix 1 for details of the registers to be set before setting these instructions.	

Event clear Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x04	0x008	EVENT_CLEAR	WO	Clears the source of the interrupt. This register is written by firmware to clear interrupts set in the EVENT_STATUS register. One or more interrupts can be cleared in a single write operation, by writing a 1 at the bit fields specific to the interrupts being cleared. It is not required to write a follow-up write with zero as the previous write is not actually stored.	0x0000

Field	Bit	Description	Reset
Reserved	15:6	Not used.	XX
Dsm_intr_clr	5	Clear deep sleep mode exit interrupt. Write to the register with this bit set to 1, clears the interrupt source.	
Sm_intr_clr	4	Clear sleep-mode-exit interrupt. Write to the register with this bit set to 1, clears the interrupt source.	
Reserved	3:0	Not used.	

- Event status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x04	0x008	EVENT_STATUS	RO	Event (Interrupt) status. Indicates pending events which require servicing by firmware. Each of the status bits is set by the link layer hardware. The bits are set till they are cleared by firmware by writing to appropriate event clear registers.	0x0000

Field	Bit	Description	Reset
Reserved	15:7	Not used.	XX
Enc_intr	6	Encryption module interrupt.	0
Dsm_intr	5	Deep sleep mode exit interrupt. This bit is set, when link layer hardware exits from deep sleep mode. The bit is cleared when firmware writes to EVENT_CLEAR register with this bit position set to 1.	0
Sm_intr	4	Sleep-mode-exit interrupt. This bit is set, when link layer hardware exits from sleep mode. The bit is cleared when firmware writes to EVENT_CLEAR register with this bit position set to 1.	0
Conn_intr	3	Connection interrupt. If bit is set to 1, it indicates an event occurred in the connection operation. This interrupt is aggregation of interrupts for all the connections. The source of the event for the specific connection needs to be read from the CONN_INTR_STATUS register specific to the connection. This bit is cleared, when firmware clears ALL interrupts by writing to the CONN_INTR_CLEAR register.	0
Init_intr	2	Initiator interrupt. If bit is set to 1, it indicates an event occurred in the initiating procedure. The source of the event needs to be read from	0

Table continues on the next page...

		the INIT_STATUS register. This bit is cleared, when firmware clears ALL interrupts by writing to the INIT_INTR_CLEAR register.	
Scan_intr	1	Scanner interrupt. If bit is set to 1, it indicates an event occurred in the scanning procedure. The source of the event needs to be read from the SCAN_STATUS register. This bit is cleared, when firmware clears ALL interrupts by writing to the SCAN_INTR_CLEAR register.	0
Adv_intr	0	Advertiser interrupt. If bit is set to 1, it indicates an event occurred in the advertising procedure. The source of the event needs to be read from the ADV_STATUS register. This bit is cleared, when firmware clears ALL interrupts by writing to the ADV_INTR_CLEAR register.	0

- Event enable register

Addr	**Addr**	**Register Name**	**RW**	**Description**	**Reset**
16-bit	**32-bit**				
0x08	0x010	EVENT_ENABLE	RW	Event indications enable. The register enables/ masks each of the possible event sources from causing an interrupt. The bit fields mask only the events from interrupting the firmware. However hardware can still generate the interrupt. Firmware, when detects one interrupt, can mask all the interrupts temporarily, by clearing the register value to 0x0000. The interrupts can be masked till the first interrupt is processed and	0x0000

				enable the interrupts back. This ensures no new interrupts is missed while firmware is processing one.	

Field	Bit	Description	Reset
Reserved	15:7	Not used.	XX
enc_int_en	6	Encryption module interrupt enable. 1 – Enable encryption module interrupt to firmware. 0 – disable encryption module interrupt to firmware.	0
Dsm_int_en	5	Deep Sleep-mode-exit interrupt enable. 1 – enable deep sleep mode exit event to interrupt the firmware. 0 – disable deep sleep mode exit interrupt to firmware.	0
Sm_int_en	4	Sleep-mode-exit interrupt enable. 1 – enable sleep mode exit event to interrupt the firmware. 0 – disable sleep mode exit interrupt to firmware.	0
Conn_int_en	3	Connection interrupt enable. 1 – enable connection procedure to interrupt the firmware. 0 – disable connection procedure interrupt to firmware.	0
Init_int_en	2	Initiator interrupt enable. 1 – enable initiator procedure to interrupt the firmware. 0 – disable initiator procedure interrupt to firmware.	0
Scn_int_en	1	Scanner interrupt enable. 1 – enable scan procedure to interrupt the firmware. 0 – disable scan procedure interrupt to firmware.	0

Table continues on the next page...

Adv_int_en	0	Advertiser interrupt enable. 1 – enable advertiser procedure to interrupt the firmware. 0 – disable advertiser procedure interrupt to firmware.	0

- Wakeup Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x5C	0x0B8	WAKEUP_CONFIG	RW	Wakeup configuration to configure the various offsets while waking up from sleep mode or deep sleep mode.	0x0000

Field	Bit	Description	Reset
Dsm_offset_to_wakeup_instant	15:10	Number of “slots” before the wake up instant before which the hardware needs to exit from deep sleep mode. The slot is of 0.625ms period. This is a one-time configuration field, which is used every time hardware does an auto-wakeup before the next wakeup instant.	0
Sm_offset_to_wakeup_instant	9	Number of “slots” (1slot = 625 microseconds) before the wake up instant before which the hardware needs to exit from sleep mode. This is a onetime configuration field, which is used every time hardware does an auto-wakeup before the next wakeup instant.	0
	8	Reserved	0
Osc_startup_delay	7:0	Oscillator stabilization/startup delay. This is in X.Y format where X is in terms of number of BT slots (625 us) and Y is in terms of number of clock periods of 16KHz clock input, required for RF oscillator to	00

		stabilize the clock output to the controller on its output pin, after oscillator is turned ON. In this period the clock is assumed to be unstable, and so the controller does not turn on the clock to internal logic till this period is over. This means, the wake up from deep sleep mode must account for this delay before the wakeup instant. Osc_startup_delay[7:5] is number of slots(625us) Osc_startup_delay[4:0 is number of clock periods of 16KHz clock (Warning: Min. value of Osc_startup_delay [4:0] supported is 1 and Max. value is 9. Therefore programmable range is 1 to 9)	

Note: In case of firm ware DSM exit mode: Exit from DSM shall be in synchronization with ref_clk (625 us slots timing) and shall not come out of DSM mode before meeting the oscillator start up delay.It is expected to have 0 to 625 us extra delay to wake-up from DSM, depends upon the assertion of the dsm_exit signal (firmware exit).

- Sleep Threshold register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0x5E	0x0BC	SLEEP_THRESHOLD	RW	Sleep Threshold register. Stores threshold values for entering sleep mode or deep sleep mode. The threshold values are compared with the inactivity period to determine entry into one of the modes.	0x0000

Field	Bit	Description	Reset
Sm_threshold[3:0]	15:12	Number of inactive “slots” above which the device can enter sleep mode. If the number of slots is below this threshold, then device will not enter sleep mode.	0
Dsm_threshold[11:0]	11:0	Number of inactive “slots” above which the device can enter sleep mode. If the number of slots is below this threshold, then device will not enter deep sleep mode.	0

Note: Typically dsm_threshold > sm_threshold value. This means the following behavior is expected

No. of inactive slots(N) Behavior

N < sm_threshold No power save

Sm_threshold < N < dsm_threshold Sleep Mode

N > dsm_threshold Deep sleep mode

- Wakeup control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x60	0x0C0	WAKEUP_CONTROL	RW	Wakeup instant register. Program the 16-bit reference clock value for auto-wakeup from deep sleep mode.	0x0000

Field	Bit	Description	Reset
Wakeup_instant[15:0]	15:0	Instant, with reference to the internal 16-bit clock reference, at which the hardware must wakeup from deep sleep mode. This is calculated by firmware based on the next closest instant where a controller operation is required (like advertiser/ scanner). Firmware reads the next instant of the procedures in the corresponding	0

		*_NEXT_INSTANT registers. This value is used only when hardware auto wakeup from deep sleep mode is enabled in the clock control register.	

Note: it is recommended to program wakeup_instant such a way that the actual instant to wakeup shall be at least two counts (two slots of 625 us) ahead of reference clock when entering DSM. The actual instant to wakeup is "wakeup_instant – dsm_offset_to_wakeup_instant – osc_startup_delay, and it shall be greater than "reference clock + 2"

- Clock control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x62	0x0C4	CLOCK_CONFIG	RW	Clock control and configuration. Controls clock gating and clock switch logic.	0x0000

Field	Bit	Description	Reset
Deep_sleep_mode_en	15	Enable deep sleep mode. 1 – enable, 0 – disable. Enables hardware logic related to deep sleep mode to control the deep sleep mode operation. If disabled, the related logic is not executed and hardware cannot enter deep sleep mode.	0
Sleep_mode_en	14	Enable sleep mode. 1 – enable, 0 – disable. Enables hardware to control sleep mode operation.	0
Dsm_intr_en	13	Enable DSM exit interrupt. 1 – enable, 0 – disable. Enables hardware to generate an interrupt while exiting deep sleep mode. When enabled, interrupt is generated independent of whether exit procedure is initiated by hardware or firmware.	0

Table continues on the next page...

Field	Bit	Description	Reset
sm_intr_en	12	Enable SM exit interrupt. 1 – enable, 0 – disable. Enables hardware to generate an interrupt while exiting sleep mode – irrespective of whether it is initiated by hardware or firmware. The interrupt is captured and stored till it gets cleared. Disabling this bit mask the sleep mode exit event from hardware & firmware.	0
Dsm_auto_sleep_en	11	Enable deep sleep mode auto entry in hardware. 1 – enable hardware to enter DSM automatically 0 – disable hardware to enter DSM automatically.	0
Sm_auto_wkup_en	10	Enable sleep mode auto wakeup enable. 1- enable, 0 – disable. Enables hardware to automatically wakeup from sleep mode at the instant = *wakeup_instant – sm_offset_to_wakeup_instant.* The wakeup_insant is the field in the *wakeup control register* described earlier. The *sm_offset_to_wakeup_instant* value is the field described in the *wakeup configuration register.*	0
Lpo_sel_external	9	Select external sleep clock. 1 – External clock, 0 - internal generated clock. The field is used to select either the low power clock input on sleep_clk input pin(of frequency 16.384KHz) directly to run the DSM logic or to use the internal generated reference clock(of 16KHz) for the same.	0
Lpo_clk_freq_sel	8	Clock frequency select. 0 – 32KHz, 1 – 32.768KHz. Base frequency of the sleep_clk input used for generating the internal reference clock of approximate 16Khz frequency.	0

Table continues on the next page...

Field	Bit	Description	Reset
LLH_idle (READ ONLY field)	7	Indicates if hardware is doing any transmit/receive operation or there is a pending interrupt from hardware. This information is used by firmware to decide to program the hardware into deep sleep mode. 1 – LL hardware is idle. 0 – LL hardware is busy. In this case LL hardware will not enter deep sleep mode, even if firmware gives an enter DSM command. (In this situation hardware generates dsm exit interrupt to inform firmware that DSM entry was not successful).	1
Phy_clk_gate_en	6	Digital PHY clock enable. 1-enable, 0-disable. Enable the Digital PHY to shutdown the clock. When 1, it indicates that controller has an upcoming activity so PHY clock must be turned ON. When 0, it indicates inactivity in the controller. This bit is not used as of now.	0
Sysclk_gate_en	5	Sysclk gate enable. 1- enable, 0 – disable. Enables clock gating of system clock input to the link layer. If 1, it enables the DSM logic to control the clock gate for system clock input from pin. If 0, the DSM logic has no control and the system clock is always ON.	0
Coreclk_gate_en	4	Core clock gate enable. 1 – enable, 0 – disable. Enables gating of clock to the llh_core module in hardware. If 1, the sleep mode/deep sleep mode logic can control the clock gate to shutdown/ wakeup the clock to the module. If 0, the logic has no control and clock is always turned ON.	0
Conn_clk_gate_en	3	Connection block clock gate enable. 1 – enable, 0 – disable.	0

Table continues on the next page...

Field	Bit	Description	Reset
		Enables gating of clock to the connection module (llh_connch_top) in hardware. If 1, the sleep mode logic can control the clock gate to shutdown/wakeup the clock to the engine. If 0, the logic has no control and clock to the module is always turned ON.	
Init_clk_gate_en	2	Initiator block clock gate enable. 1 – enable, 0 – disable. Enables gating of clock to the initiator module (llh_init). If 1, the sleep mode logic can control the clock gate to shutdown/wakeup the clock to the module. If 0, the logic has no control and clock to the module is always turned ON.	0
Scan_clk_gate_en	1	Scan block clock gate enable. 1 – enable, 0 – disable. Enables gating of clock to the scanner module (llh_scan) in hardware. If 1, the sleep mode logic can control the clock gate to shutdown/ wakeup the clock to the module. If 0, the logic has no control and clock to the module is always turned ON.	0
Adv_clk_gate_en	0	Advertiser block clock gate enable. 1 – enable, 0 – disable. Enables gating of clock to the advertiser module (llh_adv) in hardware. If 1, the sleep mode logic can control the clock gate to shutdown/ wakeup the clock to the module. If 0, the logic has no control and clock to the module is always turned ON.	0

- Reference Clock register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x64	0x0C8	TIM_COUNTER_L	RO	16-bit reference clock used for timing reference in	0x0000

				the operation of the hardware.	

Field	Bit	Description	Reset
Clock[15:0]	15:0	16-bit internal reference clock. The clock is a free running clock, incremented by a 0.625ms periodic pulse. It is used as a reference clock to derive all the timing required as per protocol.	0000

- BLE Time Control

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X6C	0XD8	TIME_CONTROL	WO	LLH clock frequency configuration.	0x0000

Field	Bit	Description	Reset
	15:8	Not used --RFU	0
bb_clk_freq_minus_1	7:3	LLH clock configuration. The clock frequency of the clock input to this design is configured in this register. this is used to derive a 1MHz clock	0
	2:0	Not used	0

- Advertising parameters register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x0C	0x018	ADV_PARAMS	RW	Advertising parameters register. Firmware sets the necessary parameters for the advertising procedure into this	0x00E0

				register before issuing start advertise command. The fields in this register correspond to the fields in the LE_Set_Advertising _Parameters HCI command.	

Field	Bit	Description	Reset
Rcv_tx_addr (Read Only)	15	Transmit address field of the received packet extracted from the receive packet. This field is used by firmware to report *peer_addr_type* parameter in the connection complete event.	
Reserved	14:11	Not used.	XX
*adv_low_duty_cycle	10	This bit field is used to specify to the Controller the Low Duty Cycle connectable directed advertising variant being used. 1 – Low Duty Cycle Connectable Directed Advertising. 0 – High Duty Cycle Connectable Directed Advertising.	0
Force_scan_rsp (Write Only)	9	Force scan response packet always. *Used only if TESTER build is enabled.* • Override ADV packet type and send scan response packet type in the header field in all ADV packets 0 – no effect.	0
Rx_addr	8	Peer addresses type. This is the Direct_Address_type field programmed, only if ADV_DIRECT_IND type is sent. • Rxaddr type is random. 0 - Rxaddr type is public.	0

Table continues on the next page...

Field	Bit	Description	Reset
Adv_channel_map	7:5	Advertising channel map indicates the advertising channels used for advertising. By setting the bit, corresponding channel is enabled for use. At least one channel bit should be set. Bit 7- enable channel 39. Bit 6 - enable channel 38. Bit 5 - enable channel 37.	7
Adv_filt_policy	4:3	Advertising filter policy. The set of devices that the advertising procedure uses for device filtering is called the White List . 0x00 - Allow scan request from any device, allow connect request from any device. 0x01- Allow scan request from devices in white list only, allow connect request from any device. 0x10--Allow scan request from any device, allow connect request from devices in white list only. 0x11--Allow scan request from devices in white list only, allow connect request from devices in white list only.	0
Adv_type	2:1	The Advertising type is used to determine the packet type that is used for advertising when advertising is enabled. 00b- Connectable undirected advertising. (adv_ind) 01b- Connectable directed advertising (adv_direct_ind). 10b- Discoverable undirected advertising (adv_discover_ind) 11b- Non connectable undirected advertising (adv_nonconn_ind).	0

Table continues on the next page...

Tx_addr	0	Device Own_address_type field. 1- Address type is random. 0-Address type is public.	0

- Advertising Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x0E	0x01C	ADV_INTERVAL_TIMEOUT	RW	Advertising interval register. It is the i nterval between two consecutive advertising events. For directed advertising, this register holds the timeout value. Has a resolution of 0.625ms. Time = N * 0.625 msec Time Range: 20 ms to 10.24 sec. Firmware updates this value before issuing start advertise command.	0x0020

Field	Bit	Description	Reset
Adv_interval	15:0	Range: 0x0020 to 0x4000 (For ADV_IND) 0x00A0 to 0x4000 (For ADV_SCAN_IND and NONCONN_IND) For directed advertising, firmware programs the default value of 1.28seconds.	0x0010

- Advertising interrupt clear register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x10	0x020	ADV_INTR_CLEAR	WO	Clears the source of the interrupt. This register is written by firmware to clear interrupts set in the ADV_STATUS register. One or more interrupts can be cleared in a single write operation, by setting the bit field to 1 for corresponding interrupt. It is not required to write a follow-up write with bit 0, as the previous bit 1 is not actually stored.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Not used.	XX
adv_timeout	7	Clear adv_timeout interrupt. Applicable in ADV_DIRECT_IND advertising. Write to the register with this bit set to 1, clears the interrupt source.	0
slv_connected	6	Clear slave connected interrupt. Write to the register with this bit set to 1, clears the interrupt source.	0
conn_req_rx_intr	5	Clear connect request packet received interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
scan_req_rx_intr	4	Clear scan request packet received interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
scn_rsp_tx_intr	3	Clear scan response packet transmitted interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
adv_tx_intr	2	Clear adv packet transmitted interrupt. Write to the register with this bit set to 1, clears the interrupt source	0

Table continues on the next page...

adv_close_intr	1	Clear advertising event stop interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
adv_strt_intr	0	Clear advertising event start interrupt. Write to the register with this bit set to 1, clears the interrupt source	0

- Advertising status register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0x10	0x020	ADV_STATUS	RO	Advertising status register shows the status of the interrupts .Each of the status bits is set by the advertising procedure in hardware. The bits are set till they are cleared by firmware by writing to appropriate interrupt clear registers.	0x0000

Field	**Bit**	**Description**	**Reset**
Reserved	15:9	Not used.	XX
Adv_on	8	Advertiser procedure is ON in hardware. Indicates that advertiser procedure is ON in hardware. 1 – on 0 – off	0
adv_timeout	7	If this bit is set it indicates that the directed advertising event has timed out after 1.28 seconds. Applicable in adv_direct_ind advertising only.	0
slv_connected	6	If this bit is set it indicates that connection is created as slave.	0
conn_req_rx_intr	5	If this bit is set it indicates connect request packet is received.	0

Table continues on the next page...

scan_req_rx_intr	4	If this bit is set it indicates scan request packet received.	0
scn_rsp_tx_intr	3	If this bit is set it indicates scan response packet transmitted in response to previous scan request packet received.	0
adv_tx_intr	2	If this bit is set it indicates ADV packet is transmitted.	0
adv_close_intr	1	If this bit is set it indicates current advertising event is closed.	0
adv_strt_intr	0	If this bit is set it indicates a new advertising event started after interval expiry.	0

- Advertising next instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x12	0x024	ADV_NEXT_INSTANT	RO	Shows the instant at which the next advertising event begins. This is with reference to internal reference clock of resolution 625 us..	0x0000

Field	Bit	Description	Reset
next_adv_instant	15:0	Shows the next start of advertising event with reference to the internal reference clock.	0

- Scan Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x14	0x028	SCAN_INTERVAL	RW	Scan interval register. Interval between two consecutive scanning events. Firmware sets the scanning interval value to this	0x0010

				register before issuing start scan command.	

Field	Bit	Description	Reset
scan_interval	15:0	Interval between two consecutive scanning events. Range: 0x0004 to 0x4000 Default: 0x0010 (10 ms) Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	0X0010

- Scan Window register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x16	0x02C	SCAN_WINDOW	RW	Scan window register. Duration of scan in a scanning event, which should be less than or equal to scan interval value . Firmware sets the scan window value to this register before issuing start scan command.	0x0010

Field	Bit	Description	Reset
scan_window	15:0	Duration of scan in a scanning event, which should be less than or equal to scan interval value. Range: 0x0004 to 0x4000 Default: 0x0010 (10 ms) Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	0X0010

- Scan parameters register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x18	0x030	SCAN_PARAM	RW	Scanning parameters register. Firmware sets the necessary parameters for scanning procedure into this register before issuing start scan command. The fields are derived from the LE_Set_Scan_Parameters HCI command.	0x0000

Field	Bit	Description	Reset
Reserved	15:6	Not used.	XX
Dup_filt_en	5	Filter duplicate packets. 1 -Duplicate packet filtering enabled. 0- Duplicate packet filtering not enabled. This field is derived from the *LE_set_scan_enable* command.	0
scan_filt_policy	4:3	The scanner filter policy determines how the scanner processes advertising packets . 0x00 – Accept advertising packets from any device. 0x01- Accept advertising packets from only devices in the whitelist. 0x10-RFU 0x11-RFU Adv_direct_ind packets which are not addressed to this device are ignored.	0
scan_type	2:1	0x00- passive scanning. (default) 0x01- active scanning. 0x10- RFU	0

Table continues on the next page...

		0x11- RFU	
Tx_addr	0	Device' own address type. 1- addr type is random. 0-addr type is public.	0

- Scan interrupt clear register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1C	0x038	SCAN_INTR_CLEAR	WO	Clears the source of the interrupt. This register is written by firmware to clear interrupts set in the SCAN_STATUS register. One or more interrupts can be cleared in a single write operation.	0x0000

Field	Bit	Description	Reset
Reserved	15:5	Not used.	XX
scan_rsp_rx_intr	4	Clear scan_rsp packet received interrupt. Write to the register with this bit set to 1, clears the interrupt source.	0
adv_rx_intr	3	Clear adv packet received interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
scan_tx_intr	2	Clear scan request packet transmitted interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
scan_close_intr	1	Clear scan event close interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
scan_strt_intr	0	Clear scan event start interrupt. Write to the register with this bit set to 1, clears the interrupt source	0

- Scan status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1C	0x038	SCAN_STATUS	RO	Shows the status of the interrupt. This register is read by firmware to learn the pending scan interrupts that are set and are to be served.	0x0000

Field	Bit	Description	Reset
Reserved	15:9 and 7:6	Not used.	XX
Scan_on	8	Scan procedure is active. 1 – scan procedure is active. 0 – scan procedure is not active.	
scan_rsp_adv_rx_intr	5	If this bit is set after both SCAN_RSP and ADV packets are received. This interrupt may be enabled, if firmware desires to be interrupted after complete reception, instead of interrupt for each packet – which are enabled by bits 4 and 3.	0
scan_rsp_rx_intr	4	If this bit is set it indicates SCAN_RSP packet is received. Firmware can read the content of the packet from the INIT_SCN_ADV_RX_FIFO.	0
adv_rx_intr	3	If this bit is set it indicates ADV packet received. Firmware can read the content of the packet from the INIT_SCN_ADV_RX_FIFO.	0
scan_tx_intr	2	If this bit is set it indicates scan request packet is transmitted.	0
scan_close_intr	1	If this bit is set it indicates scan window is closed.	0
scan_strt_intr	0	If this bit is set it indicates scan window is opened.	0

Note:

scan_rsp_adv_rx_intr—This interrupt is generated while active scanning ,after receiving both the adv and scan response packets, in case of receiving adv_ind and adv_discover_ind. Currently not in use.

scan_rsp_rx_intr-- This interrupt is generated while active scanning upon receiving scan response packet.

adv_rx_intr-- This interrupt is generated while active/passive scanning upon receiving adv packets.

- Scan next instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E	0x03C	SCAN_NEXT_INSTANT	RO	Shows the instant w.r.t internal reference clock of resolution 625us at which next scanning event begins.	0x0000

Field	Bit	Description	Reset
next_scan_instant	15:0	Shows the instant w.r.t internal reference clock at which next scanning window begins.	0

- Initiator Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x20	0x040	INIT_INTERVAL	RW	Initiator interval register. Firmware sets the initiator's scanning interval value to this register before issuing create connection command.	0x0000

Field	Bit	Description	Reset
Init_scan_interval	15:0	Interval between two consecutive scanning events. Range: 0x0004 to 0x4000 Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	0X0000

- Initiator window register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x22	0x044	INIT_WINDOW	RW	Initiator window register. Firmware sets the scan window value to this register before issuing the create connection command.	0x0000

Field	Bit	Description	Reset
Init_scan_window	15:0	Duration of scan in a scanning event, which should be less than or equal to scan interval value. Range: 0x0004 to 0x4000 Default: 0x0010 (10 ms) Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	0X0000

- Initiator parameter register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x24	0x048	INIT_PARAM	RW	Initiator parameters register. Firmware sets the necessary parameters for initiation procedure to this register before issuing the create connection command. The fields in this register are derived from the parameters in the LE_Create_Connection HCI command.	0x0000

Field	Bit	Description	Reset
Reserved	15:4	Not used.	XX
init_filt_policy	3	The Initiator_Filter_Policy is used to determine whether the White List is used or not used. 0 -White list is not used to determine which advertiser to connect to. Instead the Peer_Address_Type and Peer Address fields are used to specify the address type and address of the advertising device to connect to. 1 - White list is used to determine the advertising device to connect to. Peer_Address_Type and Peer_Address fields are ignored when whitelist is used.	0
Reserved	2	Not used.	X
rx_addr/rx_tx_addr	1	Peer address type. The rx_addr field is updated by the receiver with the address type of the received connectable advertising packet. 1- addr type is random. 0-addr type is public.	0
Tx_addr	0	Own address type. 1- addr type is random. 0-addr type is public.	0

- Initiator interrupt clear register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x28	0x050	INIT_INTR_CLEAR	WO	Clears the source of the interrupt. This register is written by firmware to clear interrupts set in the INIT_STATUS register. One or more interrupts can	0x0000

				be cleared in a single write operation.	

Field	Bit	Description	Reset
Reserved	15:5	Not used.	XX
master_conn_created	4	Clear master connection created interrupt. Write to the register with this bit set to 1, clears the interrupt source.	0
Reserved	3	Not used.	X
Init_tx_start	2	Clear init transmission start interrupt. Write to the register with this bit set to 1, clears the interrupt source.	0
init_close_window	1	Clear Initiator scan window close interrupt. Write to the register with this bit set to 1, clears the interrupt source	0
init_interval_expire	0	Clear Initiator scan window start interrupt. Write to the register with this bit set to 1, clears the interrupt source	0

- Initiator status register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x28	0x050	INIT_STATUS	RO	Shows the status of the interrupt. This register is read by firmware to learn the pending initiator interrupts that are set and are to be served.	0x0000

Field	Bit	Description	Reset
Reserved	15:5	Not used.	XX
master_conn_created	4	If this bit is set it indicates connection is created as master.	0
Reserved	3	Not used.	X

Table continues on the next page...

Init_tx_start	2	If this bit is set it indicates initiator packet (CONREQ) transmission has started.	0
init_close_window	1	If this bit is set it indicates initiator scan window has finished.	0
init_interval_expire	0	If this bit is set it indicates initiator scan window has started.	0

- Initiator next instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2A	0x054	INIT_NEXT_INSTANT	RO	Shows the instant w.r.t internal reference clock of 625us resolution at which next initiator scanning event begins.	0x0000

Field	Bit	Description	Reset
next_init_instant	15:0	Shows the instant w.r.t internal reference clock at which next initiator scanning event begins.	0

- Device Random address lower register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2C	0x058	DEVICE_RAND_ADDR_L	RW	Lower 16 bit random address of the device.	0x0000

Field	Bit	Description	Reset
Rand_addr	15:0	Lower 16 bit of 48-bit random address of the device.	0

- Device Random address middle register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x2E	0x05C	DEVICE_RAND_ADDR_M	RW	Middle 16 bit random address of the device.	0x0000

Field	Bit	Description	Reset
Rand_addr	15:0	Middle 16 bit of 48-bit random address of the device.	0

- Device Random address higher register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x30	0x060	DEVICE_RAND_ADDR_H	RW	Higher 16 bit random address of the device.	0x0000

Field	Bit	Description	Reset
Rand_addr	15:0	Higher 16 bit of 48-bit random address of the device.	0

- Peer address lower register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x34	0x068	PEER_ADDR_L	RW	Lower 16 bit address of the peer device.	0x0000

Field	Bit	Description	Reset
peer_addr	15:0	Lower 16 bit of 48-bit address of the peer device.	0

- Peer address middle register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				

Table continues on the next page...

0x36	0x06C	PEER_ADDR_M	RW	Middle 16 bit address of the peer device.	0x0000

Field	Bit	Description	Reset
peer_addr	15:0	Middle 16 bit of 48-bit address of the peer device.	0

- Peer address higher register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x38	0x070	PEER_ADDR_H	RW	Higher 16 bit address of the peer device.	0x0000

Field	Bit	Description	Reset
peer_addr	15:0	Higher 16 bit of 48-bit of address of the peer device.	0

The peer address registers are used for multiple purposes. The register is written by firmware to provide the peer address to be used for a hardware procedure. When firmware reads the register, it reads back peer address values updated by hardware.

While doing directed Advertising, the firmware writes the peer address of the device specified by the Direct_Address parameter of the LE_Set_Advertising_Parameters command.

While device is configured as an initiator without white list filtering, the peer address specified in the peer_address field of the create connection command is programmed into this register, which is used by hardware procedures.

While device is configured as an initiator and white list is enabled, firmware can read this register to get the address of the peer device from which connectable ADV packet was received and to which the connection is created.

When a connection is created as a slave, the firmware can read this register to get the address of the peer device to which connection is created.

- White List address type register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x3C	0x078	WL_ADDR_TYPE	RW	Stores the address type of the device addresses stored in whitelist	0x0000

Field	Bit	Description	Reset
	15:8	Reserved	0
type	7:0	8 address type bits corresponding to the device address stored. 1-Address type is random. 0-Address type is public.	0

- White list enable register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x3E	0x7C	WL_ENABLE	RW	Stores the valid entry bit corresponding to each of the device address stored in the whitelist.	0x0000

Field	Bit	Description	Reset
	15:8	Reserved for future use	
en	7:0	Stores the valid entry bit corresponding to each of the eight device addresses stored in the whitelist memory in the hardware. 1. White list entry is valid. 2. White list entry invalid.	0

- Advertising data transmit FIFO

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x70	0x0E0	ADV_TX_DATA_FIFO	WO	IO mapped FIFO of depth 16 (2 byte wide), to store ADV data of maximum length 31 bytes for transmitting. Firmware writes consecutive words by writing to the same address location.	0x0000

Field	Bit	Description	Reset
data	15:0	Advertising data for transmission.	0

- Adv scan response data transmit FIFO

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x74	0x0E8	ADV_SCN_RSP_TX_FIFO	WO	IO mapped FIFO of depth 16 (2 byte wide), to store scan response data of maximum length 31 bytes for transmitting. Firmware writes consecutive words by writing to the same location.	0x0000

Field	Bit	Description	Reset
data	15:0	scan response data for transmission.	0

Note: ADV_TX_DATA_FIFO and ADV_SCN_RSP_TX_FIFO shares same physical FIFO of depth 32. 16 locations for each FIFO are allocated.

ADV_TX_DATA_FIFO address 16’h70	Data byte 1	Length of adv host data
	Data byte 3	Data byte 2
	Data byte 31	Data byte 30
ADV_SCN_RSP_TX_FIFO	Data byte 1	Length of scan response host data

Table continues on the next page...

address 16’h74	Data byte 3	Data byte 2
	Data byte 31	Data byte 30

The length of the payload combined with first payload data and loaded to the advertise channel data transmit FIFO followed by rest of the host data.

Example: Structure of advertising channel transmit FIFO.

bit15 bit8 bit7 bit0

Data byte 1	Length of the payload stored in FIFO
Data byte 3	Data byte 2
Data byte 5	Data byte 4
Data byte 7	Data byte 6

- Conn request data Transmit FIFO

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0x78	0x0F0	CONN_REQ_TX_FIFO	WO	IO mapped FIFO of depth 48, to store connection request data of maximum length 34 bytes for transmitting.	0x0000

Field	**Bit**	**Description**	**Reset**
Data	15:0	Connection request data during transmit operation.	0

- Adv scan response data receive FIFO

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0x7C	0x0F8	INIT_SCN_ADV_RX_FIFO	RO	IO mapped FIFO of depth 64, to store ADV and SCAN_RSP header and payload received by the scanner. The RSSI	0x0000

				value at the time of reception of this packet is also stored. Firmware reads from the same address to read out consecutive words of data.	

Field	Bit	Description	Reset
Data	15:0	adv, scan response data during receive operation.	0

Note: The 16 bit header is first loaded to the advertise channel data receive FIFO followed by the payload data and then 16 bit RSSI.

Example: Structure of advertising channel receive FIFO with payload with even number of bytes.

bit15 bit8 bit7 bit0

Header 2	Header 1
Data byte 2	Data byte 1
Data byte 4	Data byte 3
Data byte 6	Data byte 5
RSSI	RSSI

Example: Structure of advertising channel receive FIFO with payload with odd number of bytes.

bit15 bit8 bit7 bit0

Header 2	Header 1
Data byte 2	Data byte 1
Data byte 4	Data byte 3
	Data byte 5
RSSI	RSSI

- Device public address lower register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0XE0	0X1C0	DEV_PUB_ADDR_L	RW	Lower 16 bit public address of the device.	0x3412

Field	Bit	Description	Reset
public_addr	15:0	Lower 16 bit of 48-bit public address of the device.	0x3412

- Device public address middle register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0XE2	0X1C4	DEV_PUB_ADDR_M	RW	Middle 16 bit public address of the device.	0x0056

Field	Bit	Description	Reset
public_addr	15:0	Middle 16 bit of 48-bit public address of the device.	0x0056

- Device public address higher register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0XE4	0X1C8	DEV_PUB_ADDR_H	RW	Higher 16 bit public address of the device.	0x0000

Field	Bit	Description	Reset
public_addr	15:0	Higher 16 bit of 48-bit public address of the peer device.	0

- Advertising channel transmit power register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				

Table continues on the next page...

0XE6	0X1CC	ADV_CH_TX_POWER	RW	The advertising channel transmit power register sets the transmit power level used for LE advertising channel packets and for DTM mode transmissions. The same register is used for setting Transmit power level of all non-connection channels. This includes: Advertising, scanning, Initiating, and Direct test mode (DTM Transmitter tests).	0x874F

Field	Bit	Description	Reset
adv_transmit_power	15:0	Size: 1 Octet (signed integer) Range: -20 ≤ N ≤ 10 Units: dBm Accuracy: +/- 4 dBm in general. In implementation this is a radio specific value.	0x874F

- Offset to first instant register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0XE8	0X1D0	OFFSET_TO_FIRST_INSTANT	RW	Offset to the first instant register. The first event instant is determined by firmware based on other procedures which may be on with various intervals, so as to not overlap on the existing procedure instants. For this, firmware	0x0006

				determines the offset to the first instant from the current clock and programs the offset in OFFSET_TO_FIRST_INSTANT register. Unit is in time slots of 625us ex: if current clock value is 0004,and offset is 0008,then first event will begin when clock value becomes 000c.	

Field	Bit	Description	Reset
offset_to_first_event	15:0	The offset w.r.t the internal reference clock at which instant the first event occurs. This register will give flexibility to the firmware to position the connection at a desired point with respect to the internal free running clock. It is optional to be updated by firmware. This is not updated in the current firmware. This is for future use.	0x0006

- Advertiser configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xEA	0x1D4	ADV_CONFIG	RW	Advertiser procedure configuration register. Firmware sets the configuration parameters to this register before issuing start adv command.	0x20FF

Field	Bit	Description	Reset
adv_pkt_interval	15:11	Time between the beginnings of two consecutive advertising PDU's. Time = N * 0.625 msec Time Range: <=10msec.	00100b
Reserved	10:9	Not used.	0
Adv_rand_disable	8	Disable randomization of adv interval. When disabled, interval is same as programmed in adv_interval register.	0
adv_timeout_en	7	Enable adv_timeout interrupt. Applicable in adv_direct_ind advertising.	1
slv_connected_en	6	Enable slave connected interrupt.	1
adv_conn_req_rx_en	5	Enable connect request packet received interrupt.	1
adv_scn_req_rx_en	4	Enable scan request packet received interrupt.	1
scn_tx_en	3	Enable scan response packet transmitted interrupt.	1
adv_tx_en	2	Enable adv packet transmitted interrupt.	1
adv_cls_en	1	Enable advertising event stop interrupt.	1
adv_strt_en	0	Enable advertising event start interrupt.	1

- Scan configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xEC	0x1D8	SCAN_CONFIG	RW	Scanner procedure configuration register. Firmware sets the configuration parameters to this register before issuing start scan command	0xE0FF

Field	Bit	Description	Reset

Table continues on the next page...

scan_channel_map	15:13	Advertising channels that are enabled for scanning operation. 15- Enables channel 39 for use. 14- Enables channel 38 for use. 13- Enables channel 37 for use.	111
Reserved	12	Not used.	X
backoff_enable	11	Enable random backoff feature in scanner. 1-enable. 0-disable.	0
Reserved	10:6	Not used.	XX
scn_rsp_rx_en	4	Enable scan_rsp packet received interrupt .	1
adv_rx_en	3	Enable adv packet received interrupt .	1
scn_tx_en	2	Enable scan request packet transmitted interrupt.	1
scn_close_en	1	Enable scan event close interrupt.	1
scn_strt_en	0	Enable scan event start interrupt.	1

- Initiator configuration register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0xEE	0x1DC	INIT_CONFIG	RW	Initiator procedure configuration register. Firmware sets the configuration parameters to this register before issuing create connection command	0x0000

Field	**Bit**	**Description**	**Reset**
init_channel_map	15:13	Advertising channels that are enabled for initiator scanning operation.	0

Table continues on the next page...

		15- Enables channel 39 for use. 14- Enables channel 38 for use. 13- Enables channel 37 for use.	
Reserved	12:5	Not used.	XX
conn_created	4	Enable master connection created interrupt	0
Reserved	3	Reserved	
conn_req_tx_en	2	Enables connection request packet transmission start interrupt.	0
init_close_en	1	Enable Initiator scan window close interrupt.	0
init_strt_en	0	Enable Initiator scan window start interrupt.	0

- Whitelist base address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X1A0	0X340	WHITELIST_BASE_ADDR	RW	It is the starting address of white list memory which holds the white listed device address. For a 48 bit device address, three writes of 16 bits is required at the appropriate offset from this base address. The whitelist device addresses are stored as group of 3-words at offset of N*3, where N=0 to 7, from this base address. While writing the device address, the firmware writes the address in the following order for storage. 1st write - [15:0],	0x0000

				2nd write - [31:16], 3rd write - [47:32] bits of the device address.	

Field	Bit	Description	Reset
Device_addr	15:0	Device address values written to white list memory are written as 16-bit wide address.	0

- Whitelist end address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X1D0	0X3A0	WHITELIST_END_ADDR	RW	It is the last address of white list memory which holds the white list device address. It holds last [47:32] bits of 8th white list device address. It is not accessed by firmware, only used for hardware reference.	0x0000

Field	Bit	Description	Reset
Device_addr	15:0	Device address values written to white list memory are written as 16-bit wide address.	0

- Advertiser Tx memory base address register – Reserved for future use

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X160	0X2C0	ADV_TX_MEM_BASE_ADDR	RW	It is the starting address of ADV Tx memory which holds the data to be transmitted during Advertising operation.	0x0000

Field	Bit	Description	Reset
Data	15:0	Data values written to Tx memory are written as 16-bit wide data.	0

- Connection Tx memory base address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X300	0X600	CONN_TXMEM_BASE_ADDR	RW	It is the starting address of Connection Transmit data memory which holds the data to be transmitted during connection. The connection Transmit memory is individually addressable location for firmware. So firmware writes to consecutive even address values to write the next word (2-byte) of data. Hardware accesses this memory as a FIFO. The hardware buffer index is managed in hardware. No buffer index needs to be maintained for firmware access.	0x0000

Field	Bit	Description	Reset
Data	15:0	Data values written to Tx memory are written as 16-bit wide data.	0

- Connection Rx memory base address register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0X400	0X800	CONN_RXMEM_BASE_ADDR	RW	It is the starting address of Connection Receive data memory/FIFO which holds the data to be received during connection. The connection receive memory/ FIFO is used as a FIFO by both hardware and firmware. Firmware needs to read from the same address to read out the consecutive words in the FIFO.	0x0000

Field	Bit	Description	Reset
Data	15:0	Data values written to Rx memory are written as 16-bit wide data	0

- Conn_req_word0 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E0	0x3C0	CONN_REQ_WORD0	RW	The connect request word0 register must be programmed with the access address value of the connect request packet, before initiating connection. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset

Table continues on the next page...

Access_addr[15:0]	15:0	This field defines the lower 16 bits of the access address that is to be sent in the connect request packet of the initiator.	0x0000

- Conn_req_word1 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E2	0x3C4	CONN_REQ_WORD1	RW	The connect request word1 register must be programmed with the access address value of the connect request packet, before initiating connection. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Access_Address[31:16]	15:0	This field defines the upper16 bits of the access address that is to be sent in the connect request packet of the initiator.	0x0000

- Conn_req_word2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E4	0x3C8	CONN_REQ_WORD2	RW	This field defines the lower byte [7:0] of the CRC initialization value, and tx_window_size [7:0], to be sent in the connect request packet of the initiator. After slave connection these	0x0000

				values can be obtained by reading this register on the advertiser side.	

Field	**Bit**	**Description**	**Reset**
crc_init[7:0]	15:8	This field defines the lower byte [7:0] of the CRC initialization value.	0
Tx_window_size[7:0]	7:0	window_size along with the window_offset is used to calculate the first connection point anchor point for the master. This shall be a multiple of 1.25 ms in the range of 1.25 ms to the lesser of 10 ms and (connInterval – 1.25 ms). Values range from 0 to 10 ms.	0

- Conn_req_word3 register

Addr	**Addr**	**Register Name**	**RW**	**Description**	**Reset**
16-bit	**32-bit**				
0x1E6	0x3CC	CONN_REQ_WORD3	RW	This field must be programmed with the upper byte [23:8] of the CRC initialization value.After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	**Bit**	**Description**	**Reset**
crc_init[23:8]	15:0	This field defines the upper byte [23:8] of the CRC initialization value that is to be sent in the connect request packet of the initiator.	0x0000

- Conn_req_word4 register

Addr	**Addr**	**Register Name**	**RW**	**Description**	**Reset**

Table continues on the next page...

16-bit	32-bit				
0x1E8	0x3D0	CONN_REQ_WORD4	RW	This field defines the 16 bits of the transmit window offset that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Tx_window_offset	15:0	This is used to determine the anchor point for the master transmission. Range: This shall be a multiple of 1.25 ms in the range of 0 ms to connInterval value.	0

- Conn_req_word5 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1EA	0x3D4	CONN_REQ_WORD5	RW	This field defines the 16 bits of the connection interval value that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Conn_interval_val	15:0	The value configured in this register determines the spacing between the connection events. This shall be a multiple of 1.25 ms in the range of 7.5 ms to 4.0 s.	0x0000

- Conn_req_word6 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1EC	0x3D8	CONN_REQ_WORD6	RW	This field defines the 16 bits of the slave latency value that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Slv_latency_val	15:0	The value configured in this field defines the number of consecutive connection events that the slave device is not required to listen for master. The value of connSlaveLatency should not cause a Supervision Timeout. This shall be an integer in the range of 0 to ((connSupervision Timeout/ connInterval)-1). connSlaveLatency shall also be less than 500.	0x0000

- Conn_req_word7 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1EE	0x3DC	CONN_REQ_WORD7	RW	This field defines the 16 bits of the supervision timeout value that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Sup_to_val	15:0	This field defines the maximum time between two received Data packet PDUs before the connection is considered lost. This shall be a multiple of 10 ms in the range of 100 ms to 32.0 s and it shall be larger than (1+connSlaveLatency)*connInterval.	0x0000

- Conn_req_word8 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1F0	0x3E0	CONN_REQ_WORD8	RW	This field defines the channel map for channels [15:0], that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Data_channels[15:0]	15:0	This register field indicates which of the data channels are in use. This stores the information for the lower 16 (15:0) data channel indices. ‘1’ indicates the corresponding data channel is used and ‘0’ indicates the channel is unused.	Data_channels

- Conn_req_word9 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x1F2	0x3E4	CONN_REQ_WORD9	RW	This field defines the channel map for channels [31:16], that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Data_channels[31:16]	15:0	This register field indicates which of the data channels are in use. This stores the information for the middle 16 (31:16) data channel indices. '1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	Data_channels

- Conn_req_word10 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1F4	0x3E8	CONN_REQ_WORD10	RW	This field defines the channel map for channels [36:32], that is to be sent in the connect request packet of the initiator. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Data_channels[36:32]	4:0	This register field indicates which of the data channels are in use. This stores the information for the upper 5 (36:32) data channel indices.	Data_channels

		'1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	

- Conn_req_word11 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1F6	0x3EC	CONN_REQ_WORD11	RW	The connect request word0 register must be programmed with Connection parameters sca, hop_increment value of the connect request packet, before initiating connection. After slave connection this value can be obtained by reading this register on the advertiser side.	0x0000

Field	Bit	Description	Reset
Sca[2:0]	7:5	This field defines the sleep clock accuracies given in ppm.	0
hop_increment[4:0]	4:0	This field is used for the data channel selection process.	0

Transmit window offset register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x40	0x080	TRANSMIT_WINDOW_OFFSET	RW	Stores the transmit window offset parameter used during connection setup and connection update procedures.	0x0000

				The register is updated by hardware when device is slave.	

Field	Bit	Description	Reset
window_offset	15:0	This is used to determine the first anchor point for the master transmission, from the time of connection creation. Range: This shall be a multiple of 1.25 ms in the range of 0 ms to connInterval value.	0

Transmit window size register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x42	0x084	TRANSMIT_WINDOW_SIZE	RW	Stores the transmit window size parameter used during connection setup and connection update procedures. The register is updated by hardware when device is slave.	0x0000

Field	Bit	Description	Reset
window_size	7:0	window_size along with the window_offset is used to calculate the first connection point anchor point for the master. This shall be a multiple of 1.25 ms in the range of 1.25 ms to the lesser of 10 ms and (connInterval – 1.25 ms). Values range from 0 to 10 ms.	0

Data channel map 0 (lower word) register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x44	0x088	DATA_CHANNELS_L0	RW	Stores the channel map for channels [15:0]. The register is updated by firmware whenever a new/updated channel map is available for the connection.	0x0000

Field	Bit	Description	Reset
Data_channels	15:0	This register field indicates which of the data channels are in use. This stores the information for the lower 16 (15:0) data channel indices. ‘1’ indicates the corresponding data channel is used and ‘0’ indicates the channel is unused.	0x0000

Data channel map 0(middle word) register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x46	0x08C	DATA_CHANNELS_M0	RW	Stores the channel map for channels[31:16]. The register is updated by firmware whenever a new/updated channel map is available for the connection.	0x0000

Field	Bit	Description	Reset
Data_channels	15:0	This register field indicates which of the data channels are in use. This stores the information for the middle 16 (31:16) data channel indices. ‘1’ indicates the corresponding data channel is used and ‘0’ indicates the channel is unused.	0x0000

Data channel map 0(upper word) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x48	0x090	DATA_CHANNELS_H0	RW	Stores the channel map for channels[36:32]. The register is updated by firmware whenever a new/updated channel map is available for the connection.	0x0000

Field	Bit	Description	Reset
Data_channels	15:5	Unused	0
Data_channels	4:0	This register field indicates which of the data channels are in use. This stores the information for the upper 5 (36:32) data channel indices. ‘1’ indicates the corresponding data channel is used and ‘0’ indicates the channel is unused.	0x00

Data channel map 1 (lower word) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x4C	0x098	DATA_CHANNELS_L1	RW	Stores the channel map for channels[15:0]. The register is updated by firmware whenever a new/updated channel map is available for the connection.	0x0000

Field	Bit	Description	Reset

Table continues on the next page...

Data_channels	15:0	This register field indicates which of the data channels are in use. This stores the information for the lower 16 data channel indices. '1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	0x0000

Data channel map 1 (middle word) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x4E	0x09C	DATA_CHANNELS_M1	RW	Stores the channel map for channels[31:16]. The register is updated by firmware whenever a new/updated channel map is available for the connection.	0x0000

Field	Bit	Description	Reset
Data_channels	15:0	This register field indicates which of the data channels are in use. This stores the information for the middle 16 data channel indices. '1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	0x0000

Data channel map 1 (upper word) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50	0x0A0	DATA_CHANNELS_H1	RW	Stores the channel map for channels[36:32]. The register is updated by firmware whenever a new/updated	0x0000

				channel map is available for the connection.	

Field	Bit	Description	Reset
Data_channels	15:5	Unused	0
Data_channels	4:0	This register field indicates which of the data channels are in use. This stores the information for the upper 5 data channel indices. ‘1’ indicates the corresponding data channel is used and ‘0’ indicates the channel is unused.	0x00

Note: The Data channel map 0 and data channel map 1 are two sets of channel maps stored, common for all the connections. At any given time, only two maps can be maintained and the connections will use one of the two sets as indicated by the channel map index field in the CE_CNFG_STS registers specific to the link. Firmware must also manage to update this field along with the map.

Connection channel Status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x56	0x0AC	CONN_STATUS	RO	Indicates the status of the connection channel data path and other common connection channel operations.	0x0000

Field	Bit	Description	Reset
Rx_packet_counter	15:12	This field stores the count for the number of receive packets in the receive FIFO that are still not ready by firmware. The counter value is incremented by hardware for every good packet it stores in the FIFO.	0

Table continues on the next page...

		After firmware reads a packet, it decrements the counter by issuing the PACKET_RECEIVED command from the commander.	
Reserved	11:0	Reserved for future use	0

Connection configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xF0	0x1E0	CONN_CONFIG	RW	This register fields can be set to configure the LLH in data transfer scenarios.	0x11F

Field	Bit	Description	Reset
md_bit_ctr	8	This register field indicates whether the MD (More Data) bit needs to be controlled by 'software' or, 'hardware and software logic combined'. 1 - MD bit is exclusively controlled by software, ie based on status of *CE_CNFG_STS_REGISTER[6] - md bit.* 0 - MD Bit in the transmitted pdu is controlled by software and hardware logic. MD bit is set in transmitted packet, only if the software has set the md bit in CE_CNFG_STS_REGISTER[6] and either of the following conditions is true, 1. If there are packets queued for transmission. 2. If there is an acknowledgement awaited from the remote side for the packet transmitted.	0x1

Table continues on the next page...

rx_intr_threshold	7:4	This register field allows setting a threshold for the packet received interrupt to the firmware. For example if the value programmed is 0x2 – then HW will generate interrupt only on receiving the second packet. In any case if the received number of packets in a conn event is less than the threshold or there are still packets (less than threshold) pending in the Rx FIFO, HW will generate the interrupt at the ce_close. Min value possible is 1. Max value depends on the Rx FIFO capacity.	0x1
rx_pkt_limit	3:0	Defines a limit for the number of Rx packets that can be received by the LLH. Default maximum value is 0xF.Minimum value shall be ‘1’ or no packet will be stored in the Rx FIFO.	0xF

Note:

The DUT should not send empty packets with MD bit set to 1. This will extend the connection event and increase the power consumption which is unnecessary. However there could be scenarios where maximum throughput is the target. In that case we would need to extend the connection event and allow additional time for the software to queue additional data. So both the behaviors are kept in the implementation and can be selected using md_bit_ctr *(ie CONN_CONFIG[8]).*

md_bit_ctr = *1 - MD bit is exclusively controlled by software.*

If this bit is set, the MD bit in the transmitted packets is exclusively based on the status of md bit (CE_CNFG_STS_REGISTER[6]).

In this mode, empty packets with MD bit set would be transmitted by us during the time an acknowledgement is being processed in the other end. This feature will extend the connection event since the remote end host will get see the MD bit and so stays in the same connection event, and this will allow us more time for our software to process the acknowledgement and queue additional data from the host to hardware. This is useful when we need to maximize the data transmitted in a connection interval.

sw_cntrl_md = 0 - MD bit is controlled by software and hardware logic.

Note that MD bit is not set in the transmitted packet if software has not set the md bit in CE_CNFG_STS_REGISTER[6] as 0b.

In this mode of operation an empty packet will still be sent since the send status clearing and more data check is at same time.

Connection channel transmit power register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0xF2	0x1E4	CONN_CH_TX_POWER	RW	Connection channel transmit power. This register controls transmit power on all connection channel transmissions.	0x0000

Field	**Bit**	**Description**	**Reset**
Conn_tx_power[15:0]	15:0	Transmit power to be used for all packets transmitted on the connection channel.	0x0000

Connection Interrupt mask register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0xF8	0x1F0	CONN_INTR_MASK	RW	Connection Interrupt enable register. This register controls enabling of interrupts and other enables common for all connections.	0x0000

Field	**Bit**	**Description**	**Reset**
ping_nearly_expird_intr	15	If this bit is set ping timer nearly expired interrupt is enabled.	0
ping_timer_expird_intr	14	If this bit is set ping timer expired interrupt is enabled.	0

Table continues on the next page...

Reserved	13:10	Unused.	0
rx_bad_pdu_int_en	9	If this bit is set packet receive bad pdu interrupt is enabled. Effective only when bit 6 is set.	0
rx_good_pdu_int_en	8	If this bit is set packet receive good pdu interrupt is enabled. Effective only when bit 6 is set.	0
conn_updt_intr_en	7	If this bit is set connection update interrupt is enabled.	0
ce_rx_int_en	6	If this bit is set interrupt is enabled for reception of packet in a connection event. Bit 8 and 9 are sub-mask bits below this mask.	0
ce_tx_ack_int_en	5	If this bit is set transmission acknowledgement interrupt is enabled: This interrupt is generated to indicate to the firmware that a non-empty packet transmitted is successfully acknowledged by the remote device. For negative acknowledgements from remote device, this interrupt indication is not generated.	0
close_ce_int_en	4	If this bit is set connection event closed interrupt is enabled.	0
start_ce_int_en	3	If this bit is set connection event start interrupt is enabled	0
map_updt_int_en	2	If this bit is set, channel map update interrupt is enabled.	0
conn_estb_int_en	1	If this bit is set connection establishment interrupt is enabled.	0
conn_cl_int_en	0	If this bit is set connection closed interrupt is enabled.	0

Slave timing control register

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0xFA	0x1F4	SLAVE_TIMING_CONTROL	RW	Slave timing control register. This register controls slave related timing.	0xBE96

Field	Bit	Description	Reset
Slave_time_adj_val	15:8	Timing adjust value. The internal micro second counter is adjusted to this value whenever slave receives a good access address match at connection anchor point. This will ensure the slave gets synchronized to master timing.	0xBE
Slave_time_set_val	7:0	Programmable adjust value to the clock counter when slave is connected	0x96

Window widen for offset register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0xB2	0x164	WINDOW_WIDEN_ WINOFF	RW	Window widen value corresponding to transmitwindowoffset. Firmware calculates the possible drift due to transmitwindowoffset to the first packet after connection/ connection update and programs the value into this register. The value is in microseconds.	0x000A

Field	Bit	Description	Reset
Reserved	15:12	Unused	
Window_widen	11:0	This field stores the additional number of microseconds the slave must extend its listening window to listen for a master packet for receiving the first packet after connection creation. This value is calculated based on the window offset value to the first anchor point. This is used at connection setup directly. During connection setup, this value is added with	00A

		window_widen_intvl register value to calculate the window widening size.	

Connection Index register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x58	0x0B0	CONN_INDEX	RW	Index of the connection to which the connection-specific parameter is being written to or read from. This register must be updated before writing the connection-specific registers (refer to register summary before for the connection specific register set). (RESERVED FOR FUTURE USE)	0x0000

Field	Bit	Description	Reset
conn_index	15:0	This field is used to index the multiple connections existing. Range is 0 to maximum number of connections supported. For a single connection device, conn_index is 0.	0

*Connection Interrupt clear register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x54	0x0A8	CONN_INTR_CLEAR	WO	Clear connection interrupts. Write to the register to clear one more connection interrupts. This register is implemented per	0x0000

				connection. To clear interrupt for a specific connection, connection index register must be programmed before writing to this register.	

Field	Bit	Description	Reset
conn_ping_timer_nearly_expire	15	If this bit is written with 1, it clears the conn_ping_timer_nearly_expire interrupt.	0
conn_ping_timer_expire	14	If this bit is written with 1, it clears the conn_ping_timer_expire interrupt.	0
Reserved	13:8	Unused	0
con_updt_done	7	If this bit is written with 1, it clears the connection updated interrupt.	0
ce_rx	6	If this bit is written with 1, it clears the connection event received interrupt.	0
ce_tx_ack	5	If this bit is written with 1, it clears the ce transmission acknowledgement interrupt.	0
close_ce	4	If this bit is written with 1, it clears the connection event closed interrupt.	0
start_ce	3	If this bit is written with 1, it clears the connection event started interrupt.	0
map_updt_done	2	If this bit is written with 1, it clears the map update done interrupt.	0
conn_estb	1	If this bit is written with 1, it clears the connection established interrupt.	0
conn_closed	0	If this bit is written with 1, it clears the connection updated interrupt.	0

*Connection Interrupt status register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				

Table continues on the next page...

0x54	0x0A8	CONN_INTR_STATUS	RO	Connection Interrupt status register. To read interrupt for a specific connection, connection index register must be programmed before reading from this register.	0x0000

Field	Bit	Description	Reset
ping_nearly_expird_intr	15	If this is set, it indicates that ping timer has nearly expired.	0
ping_timer_expird_intr	14	If this is set, it indicates that ping timer has expired.	0
rx_pdu_status	13:11	Status of PDU received. This information is valid along with receive interrupt. Xx1 – Bad Packet (packet with CRC error) 000 – empty PDU 010 - new data (non-empty) PDU 110 – Duplicate Packet	0
discon_status	10:8	Reason for disconnect – indicates the reason the link is disconnected by hardware. 001 – connection failed to be established 010 - supervision timeout 011 – kill connection by host 100 – kill connection after ACK transmitted 101 – PDU response timer expired	0
con_updt_done	7	This bit is set when the last connection event with previous connection parameters is reached. The bit is set immediately after the receive operation at the anchor point of the last connection event.	0
ce_rx	6	If this bit is set it indicates that a packet is received in the connection event.	0

Table continues on the next page...

ce_tx_ack	5	If this bit is set it indicates that the connection event transmission acknowledgement is received for the previous non-empty packet transmitted.	0
close_ce	4	If this bit is set it indicates that the connection event closed interrupt has happened.	0
start_ce	3	If this bit is set it indicates that the connection event started interrupt has happened.	0
map_updt_done	2	If this bit is set it indicates that the channel map update is completed at the instant specified by the firmware.	0
conn_estb_updt	1	If this bit is set it indicates that the connection has been established. The bit is also set when a connection update procedure is completed, at the start of the first anchor point with the updated parameters.	0
conn_closed	0	If this bit is set it indicates that the link is disconnected.	0

Connection Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x80	0x100	CONN_INTERVAL	RW	Connection Interval registers. Firmware writes the connection interval specific to the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x0000

Field	Bit	Description	Reset
connection Interval	15:0	The value configured in this register determines the spacing between the connection events.	0x0000

		This shall be a multiple of 1.25 ms in the range of 7.5 ms to 4.0 s.	

Supervision timeout register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x82	0x104	SUP_TIMEOUT	RW	Supervision timeout for the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x0000

Field	Bit	Description	Reset
supervision_timeout	15:0	This field defines the maximum time between two received Data packet PDUs before the connection is considered lost. This shall be a multiple of 10 ms in the range of 100 ms to 32.0 s and it shall be larger than (1+connSlaveLatency)*connInterval.	0x0000

Slave Latency register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x84	0x108	SLAVE_LATENCY	RW	Slave latency for the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x0000

Field	Bit	Description	Reset
slave_latency	15:0	The value configured in this field defines the number of consecutive connection events that the slave device is not required to listen for master. The value of connSlaveLatency should not cause a Supervision Timeout. This shall be an integer in the range of 0 to ((connSupervision Timeout/ connInterval)-1). connSlaveLatency shall also be less than 500.	0x0000

Connection event length register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x86	0x10C	CE_LENGTH	RW	Connection event length for the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x00 00

Field	Bit	Description	Reset
Connection event Length	15:0	This field defines the length of Connection event. This value is derived from the CE length HCI parameters received from the host. This determines the number of master transmit slots in a connection event, subject to either of the MD bits being set. If both MD bits are set to 0, this has no effect. Units : 625 us. Note: The connection event length as specified by the CE_LENGTH shall not exceed CONN_INTERVAL – 1.25 ms.	0x0000

The CE-length parameter, according to the Bluetooth specification, is the length of the connection event.

Take an example to illustrate this scenario:

Assume a connection with interval = 100ms. that the application has put allowed 20ms of CE-length.

Here, the CE-length can be up to 100ms (100ms - 150us to be exact).

If the connection is maintained for 5 minutes, there could be 10*60*5 = 3000 connection-intervals.

The CE-length need not maintained constant during all the 3000 connection events.

Here are the typical cases that determine the value of CE-length:

(1) No data packets exchanged. we are just maintaining time and frequency synchronization. In this case, only a packet pair will be exchanged every connection interval. Here, CE-length = 1.

(2) Average of 10 packets to be sent per connection event.

We can pump data in multiple ways here:

2.1: Send data at uniform rate : In this case, the CE-length will be enough to accommodate 10 packets, which will take about 7ms. As this is less than application enforced limit of 20ms, we can comfortably push all the 10 data packets in this connection interval. So data will be pumped to the other BT device at the same rate as is received from my application.

2.2: Can send data in bursts. Assume that we accumulate data for 1 second and pump out at the end of 1 second(this is not done by our Bluetooth stack, the application needs to buffer the data). So, at 10th connection interval, we have 100 packets accumulated. We are now ready to pump this data. 100 packets take about 70 ms. This is above the application enforced 20ms. So, the hardware can pump data that can fill up 20ms. The remaining data will be deferred to the next connection interval.

So, in this case, you would see a CE-length spread over time like this (Per connection interval):

0,0,0,0,0,0,0,0,0,0, 20,20,20,10,0,0,0,0,0,0, 20,20,20,10,0,0,0,0,0,0,
20,20,20,10,0,0,0,0,0,0,

and so on.

(3) We are receiving data at the same rate as in (2). This case is to honor data sent by the other BT-device by giving it more time in the current connection interval.

In (2) and (3) you will see non-empty packets either transmitted or received. We can also utilize the CE-length for different reasons:

(4) A transaction is in progress, and we are expecting a response packet very soon. In this case, we may be exchanging only empty packets now, and in the next few packet-pairs.

In this case, you will the CE-length to be large, and a non-empty packet may not be exchanged in all the slots.

Access address (lower) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x88	0x110	PDU_ACCESS_ADDR_L_REGISTER	RW	Access address bits 15:0 for the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x0000

Field	Bit	Description	Reset
PDU Access Address Lower bits	15:0	This field defines the lower 16 bits of the access address for each Link layer connection between any two devices.	0x0000

Access address (upper) register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x8A	0x114	PDU_ACCESS_ADDR_H_REGISTER	RW	Access address bits 32:16 for the connection. The connection index register must be programmed with index of the connection, before programming the register.	0x0000

Field	Bit	Description	Reset
PDU Access Address Lower bits	15:0	This field defines the higher 16 bits of the access address for each Link layer connection between any two devices.	0x0000

Connect Event Instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x8C	0x118	CONN_CE_INSTANT	RW	This is the instant used for connection update procedure and channel map update procedure.	0x0000

Field	Bit	Description	Reset
Ce_instant	15:0	This is the value of the free running Connection Event counter when the new parameters of ‘connection update’ and/or ‘Channel map update’ will be effective. Range : 0x0000 to 0xFFFF	0x0000

Connect Event Counter register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x92	0x124	CONN_CE_COUNTER	RW	This is the free running counter, connEventCounter as defined by Bluetooth spec.	0x0000

Field	Bit	Description	Reset
connectionEventCounter	15:0	This is the free running counter, connEventCounter as defined by Bluetooth spec. Firmware will read the instantaneous Event counter from this register, during connection update and channel map update	0x0000

		procedure. Firmware will use this value to calculate the instant from which the new parameters (for connection update and channel map update) will be effective.	

Connection configuration & status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x8E	0x11C	CE_CNFG_STS_REGISTER	RW	Connection specific configuration and status information register. This register facilitates the pause/resume mechanism of data PDUs by LL , typically used during “enable encryption” procedure. In case multiple connections are supported, “connection index register” has to be written by the f/w before programming/ reading this register	0x0000

Field	Bit	Description	Reset
current_pdu_index	15:12	Read Only field. The index of the transmit packet buffer that is currently in transmission/ waiting for transmission.	0
Reserved	11	Reserved for future use.	0
conn_active	10	Read Only field . This bit is ‘1’ whenever the connection is active.	0
force_nesn0	9	not used	0
pause_data	8	Pause data. 1 – pause data, 0 – do not pause.	0

Table continues on the next page...

		Pause the data transfer on the connection. The current_pdu_index in hardware does not move to next index until pause_data is cleared.	
map_index/curr_index	7	Mixed info field. Written by firmware to select the channel map register set to be used by hardware for this connection. 1 – use channel map register set 1. 0 – use channel map register set 0. When firmware reads this field, it returns the current map index being used in hardware.	0
md	6	MD bit set to ‘1’ indicates device has more data to be sent.	0
mas_slv	5	mas_slv bit set to ‘1’ indicates that device is configured as a master or a slave. 1 – master, 0 – slave.	0
data_list_head_up	4	Update the first packet buffer index ready for transmission to start/resume data transfer after a pause. The bit must be toggled every time the firmware needs to indicate the start/resume. This requires a read modify write operation.	0
Data_list_index/ last_ack_index	3:0	Data list index for start/ resume. This field must be valid along with data_list_head_up and indicate the transmit packet buffer index at which the data is loaded. The default number of buffers in the IP is 5,but may be customized for a customer. The buffers are indexed 0 to 4. Hardware will start the next data transmission from the index indicated by this field.	0

Next CE instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x90	0x120	NEXT_CE_INSTANT	RO	16-bit internal reference clock value at which the next connection event will occur on a connection. The connection index register must be programmed with index of the connection, before reading the register.	0x0000

Field	Bit	Description	Reset
Instant	15:0	16-bit internal reference clock value at which the next connection event will occur on a connection. The connection index register must be programmed with index of the connection, before reading the register.	0x0000

Connection parameter 1 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xF4	0x1E8	CONN_PARAM1	RW	Connection parameters like sca, hop_increment and crc_init exchanged in connect_request of this connection.	0x0000

Field	Bit	Description	Reset
crc_init[7:0]	15:8	This field defines the lower byte (7:0) of the CRC initialization vector.	0
hop_increment[4:0]	7:3	Hop increment for connection channel.	
sca[2:0]	2:0	Sleep Clock accuracy value.	

Connection parameter 2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xF6	0x1EC	CONN_PARAM2	RW	Connection parameter crc_init bits 24:7 exchanged in connect_request of this connection. The connection index register must be programmed with index of the connection, before reading the register.	0x0000

Field	Bit	Description	Reset
crc_init[23:8]	15:0	This field defines the upper two bytes (23:8) of the CRC initialization vector.	0x0000

Connection Update New Interval

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1D2	0x3A4	CONN_UPDATE_NEW_INTERVAL	RW	The connection interval that will be effective after the connection update instant.	0x0000

Field	Bit	Description	Reset
Conn_interval[15:0]	15:0	This register will have the new connection interval that the hardware will use after the connection update instant. Before the instant, the connection interval in the register CONN_INTERVAL will be used by hardware.	0x0000

Connection Update New Latency

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x1D4	0x3A8	CONN_UPDATE_ NEW_LATENCY	RW	The slave latency that will be effective after the connection update instant.	0x0000

Field	Bit	Description	Reset
Slave_latencyl[15:0]	15:0	This register will have the new slave latency parameter that the hardware will use after the connection update instant. Before the instant, the connection interval in the register SLAVE_LATENCY will be used by hardware.	0x0000

Connection Update New Su To

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1D6	0x3AC	CONN_UPDATE_ NEW_SU_TO	RW	The Supervision timeout that will be effective after the connection update instant.	0x0000

Field	Bit	Description	Reset
Conn_interval[15:0]	15:0	This register will have the new supervision timeout that the hardware will use after the connection update instant. Before the instant, the connection interval in the register SUP_TIMEOUT will be used by hardware.	0x0000

Connection Update New slaveLatency x connInterval value

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x1D8	0x3B0	CONN_UPDATE_NEW_SL_INTERVAL	RW	The (slaveLatency * connInterval) value that will be effective after the connection update instant.	0x0000

Field	Bit	Description	Reset
sl_conn_interval_val[15:0]	15:0	This register will have the new SL*CI value that the hardware will use after the connection update instant. Before the instant, the connection interval in the register SL_CONN_INTERVAL will be used by hardware.	0x0000

Window Widen for Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xB0	0x160	WINDOW_WIDEN_INTVL	RW	Window widening value based on connection interval of the connection. The connection index register must be programmed with index of the connection, before reading the register.	0x000A

Field	Bit	Description	Reset
Reserved	15:12	Not used	
Window_widen	11:0	This value defines the increased listening time for the slave. The windowWidening shall be smaller than ((connInterval/2)-T_IFS us) This value is calculated by firmware based on the drift, the connection interval value. The value is the unit widening	0x000A

		value for one connection interval duration. In case of slave latency, this value is accumulated till the next anchor point at which the slave will listen.	

PDU response timer register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x502	0xA04	PDU_RESP_TIMER	RW	PDU response timer register. This timer is used to monitor the time to get the response for the control procedures for which timeout rules are specified in the specification.	0x0000

Field	Bit	Description	Reset
Pdu_resp_time_val	15:0	This register is loaded with the count value to monitor the time to get a response for the control PDU sent to a peer device. Firmware starts the timer by issuing the command, RESP_TIMER_ON, after it has queued a control PDU for transmission that requires a response. If a response is received, firmware stops and clears the timer by issuing the command RESP_TIMER_OFF. If this timer expires, it results in hardware closing the connection and triggering a conn_closed interrupt. The *discon_status* field in the Connection status register is set with the appropriate reason. Units : Milliseconds. Resolution : 1.25 ms	0

Next response timeout instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x504	0xA08	NEXT_RESP_TIMER_EXP	RO	16-bit internal reference clock value at which the next PDU response timeout event will occur on a connection.	0x0000

Field	Bit	Description	Reset
Next_response_instant	15:0	This field defines the clock instant at which the next PDU response timeout event will occur on a connection. This is with reference to the 16-bit internal reference clock.	0x0000

Next Supervision timeout instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x506	0xA0C	NEXT_SUP_TO	RO	16-bit internal reference clock value at which the next supervision timeout event will occur on a connection.	0x0000

Field	Bit	Description	Reset
Next_timeout_instant	15:0	This field defines the clock instant at which the next connection supervision timeout event will occur on a connection. This is with reference to the 16-bit internal reference clock.	0x0000

Data list SENT Status register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x94	0x128	DATA_LIST_SENT_STATUS	WO	The register is used by firmware to indicate that a packet buffer is queued (loaded) with data for transmission. Firmware sets a SENT bit in hardware corresponding to the packet buffer queued with a packet for transmission.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Unused	0
Set/Clear	7	Used to set the SENT bit in hardware for the selected packet buffer. 1 – packet queued When firmware has a packet to send, firmware first loads the next available packet buffer. Then the hardware SENT bit is set by writing 1 to this bit field along with the list_index field that identified the buffer index. This indicates that a packet has been queued in the data buffer for sending. This packet is now ready to be transmitted. The SENT bit in hardware is cleared by hardware only when it has received an acknowledgement from the remote device. Firmware typically does not clear the bit. However, It only clears the bit on its own if it needs to 'flush' a packet from the buffer, without waiting to receive acknowledgement from the remote device, firmware clears BIT7 along with the list_index specified.	0

Table continues on the next page...

		Note: This register has a different meaning in the Read-path	
Reserved	6:4	Unused	0
List_index	3:0	Indicates the buffer index for which the SENT bit is being updated by firmware. The default number of buffers in the IP is 5. The index range is 0-4.	0

Data list ACK update register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x96	0x12C	DATA_LIST_ACK_STATUS	WO	Clear ACK indication for the packet, as reported by link layer hardware.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Unused	0x0000
Set/clear	7	Firmware uses the field to clear the ACK bit in the hardware to indicate that the acknowledgement for the transmit packet has been received and processed by firmware. Firmware clears the ACK bit in the hardware by writing in this register only after the acknowledgement is processed successfully by firmware. For clearing ack for a packet transmitted in fifo-index : ‘3’, firmware will write ‘3’ in the ‘list-index’ field and set this bit (BIT7) to 0. This is the indication that the corresponding packet buffer identified by List-Index is cleared of previous transmission and can be re-used for another packet from now on.	0

Table continues on the next page...

		The ACK bit in hardware is set by hardware when it has successfully transmitted a packet. Note: This register has a different meaning in the Read-path	
Reserved	6:4	Unused	0
List_index	3:0	Indicates the buffer index for which the ACK bit is being updated by firmware. The default number of buffers in the IP is 5. The index range is 0-4.	0

Data list SENT status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x94	0x128	DATA_LIST_SENT_STATUS	RO	Status of SENT bit of all the transmit buffers available in the hardware.	0x0000

Field	Bit	Description	Reset
Reserved	15:5	Unused	0x0000
tx_sent	4:0	The bits in this field indicate the status of the SENT bit in the hardware for each packet buffer. The bit values are 1 – queued 0 – no packet / packet ack received by hardware Example1: If the read value is : 0x03, then packets in buffer 0 and buffer 1 are in the queue to be transmitted. All the other FIFOs are empty or hardware has cleared them after receiving acknowledgement.	0

Data list ACK update register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x96	0x12C	DATA_LIST_ACK_ STATUS	RO	Status of ACK bit of all the transmit buffers available in the hardware.	0x0000

Field	Bit	Description	Reset
Reserved	15:5	Unused	0x0000
tx_ack	4:0	If a particular bit is set, then the packet in the selected buffer has been transmitted (at least once) by the hardware and hardware is waiting for acknowledgement. Example1 : If the read value is : 0x03, then packets in FIFO-0 and FIFO-1 are acknowledged by the remote device. These acknowledgements are pending to be processed by firmware. Example2 : If the read value is : 0x02, then packet FIFO-1 is acknowledged by the remote device. This acknowledgement is pending to be processed by firmware. Note : This register has a different meaning in the Write-path.	0

The SENT bit and ACK bit have to be taken together to understand the meaning of packet status. The table below describes how the two bits are sequentially updated by either hardware/firmware to complete one data transmission.

SENT	ACK	Description
0	0	Buffer is empty. No packet is queued in the buffer
1	0	Packet is queued by firmware.
1	1	Packet is transmitted by hardware. Hardware is waiting for acknowledgement.
0	1	Hardware has received ACK. Firmware has not yet processed the ACK.
0	0	Firmware has processed the ack. The buffer is again empty.

Data list Index 0 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x98	0x130	LIST_INDEX0	WO	Reserved for future use.	0x0000

Data list Index 1 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x9A	0x134	LIST_INDEX1	WO	Reserved for future use.	0x0000

Data buffer descriptor 0-7 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA0-0xAE	0x140-0x15C	DATA_MEM_DESCRIPTOR0- DATA_MEM_DESCRIPTOR7	RW	Descriptor for packet stored in the each of the transmit buffer which includes the packet specific information like length and LLID.	0x0000
DATA_MEM_DESCRIPT 5 to DATA_MEM_DESCRIPT 7 is RFU.					

Field	Bit	Description	Reset
conn_handle	15:8	Unused. Reserved for future use.	0x0
Enc	7	Unused. Reserved for future use.	
Data_length	6:2	This field indicates the length of the data packet. Range: 0x0 to 0x1F.	0
LLID	1:0	The LLID indicates whether the packet is an LL Data PDU or an LL Control PDU. 00b= Reserved.	

		01b=LL Data PDU: Continuation fragment of an L2CAP message or an Empty PDU. 10b=LL Data PDU: Start of an L@CAP message or a complete L2CAP message with no fragmentation. 11b=LL Control PDU.	

Feature Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x508	0xA10	LLH_FEATURE_CONFIG	RW	Enabling/Disabling of different features implemented in connection.	0x0002

Field	Bit	Description	Reset
Reserved	7:2	RFU	0
sl_dsm_en	1	Enable/Disable Slave Latency Period DSM.	1
Quick_transmit	0	Quick transmit feature in slave latency is enabled by setting this bit. When slave latency is enabled, this feature enables the slave to transmit in the immediate connection interval, in case required, instead of waiting till the end of slave latency	0

Win_min_step_size register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50A	0xA14	WIN_MIN_STEP_SIZE	RW	The step size with which receive window is gradually decreased/ increased to achieve minimum safe window.	0x2064

Field	Bit	Description	Reset
Windows_min_fw	15:8	The minimum window size allowed. While the slave receive window is decremented, the windows_min_fw sets the lowest value of the window widen value to ensure packets are not missed. The unit is in microseconds.	0x20
Stepup	7:4	If packets are missed, the reference window is gradually increased by step up size, until it receives 2 consecutive good packets. The unit is in microseconds.	6
Stepdn	3:0	After receiving 2 consecutive good packets the reference window is gradually decremented by step down size until it reaches window minimum. The unit is in microseconds.	4

Slave Window Adjustment

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50C	0xA18	SLV_WIN_ADJ	RW	Programmable adjust value to add to the calculated window widen value. This allows the firmware to add flexibility to calculated value of drift - to compensate for any underestimated drift in the manufacturer specification of crystal drift.	0x0010

Field	Bit	Description	Reset
Reserved	15:11	Unused	0x00
Slv_win_adj	10:0	Window Adjust value. This value is added to the	0x10

		calculated slave window widening value to be used as final window widen value.	

slaveLatency x connInterval value

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50E	0xA1C	SL_CONN_INTERVAL	RW	Programmable Register which holds the (slaveLatency * connInterval) value for the connection.	

Field	Bit	Description	Reset
Sl_conn_interval_val	15:0	This field defines the (SL*CI) product for the ongoing connection. This value is used in calculation of next connection instant during slave latency.	0x0000

*LE Ping timer address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x510	0xA20	CONN_PING_TIMER_ADDR	RW	The register used to configure the LE Authenticated payload Timeout (LE APTO) which is the Maximum amount of time specified between packets authenticated by a MIC. This value of ping timer is in the order of 10ms, valid range 0x1 ~ 0xFFFF	0x0000

*LE Ping connection timer offset

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x512	0xA24	CONN_PING_TIMER_OFFSET	RW	The value of ping timer nearly expired offset in the order of 10ms, valid range 0x0 ~ 0xFFFF. This is the time period after which the ping timer nearly expired interrupt is generated.	0x0000

*LE Ping timer next expiry instant

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x514	0xA28	CONN_PING_TIMER_NEXT_EXP	RO	The value of ping timer next expiry instant in the terms of native clock value (least 16 bit value of the 17 bit ping counter). This together with CONN_PING_TIMER_NEXT_EXP_WRAP will provide the correct status of ping timer duration.	0x0000

*LE Ping timer next expiry wrap count

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x516	0xA2C	CONN_SEC_CURRENT_WRAP	RO	This register holds the current position of the Ping timer.	0x0000

Field	Bit	Description	Reset
RESERVED	15:9	RFU	0x0

Table continues on the next page...

CONN_SEC_NEAR_WRAP	8:5	This field provides the time offset of the nearly expired event from the authentication payload timeout event. This offset is in the order of 40959.375 ms and specifies the time offset after starting the ping timer the nearly expired event will be generated. Time= N*40959.375 ms.	0x0
CONN_SEC_CURRENT_WRAP	4:1	This field provides the current position of the ping timer and the value is in the order of 40959.375 ms. Time= N*40959.375 ms For Example if the APTO configured in 655,350 ms and this field returns 10, it means another 6 more units are remaining for the APTO event to be generated.	0x0
WRAP_VALID	0	This field will be ‘1’ from nearly expired event to the authenticated payload timeout or till the next reload of the ping timer.	0x0

Packet Counter 0-2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x200-0x204	0x400-0x408	PACKET_COUNTER0- PACKET_COUNTER2	RW	It is the packet counter value (39:0) which is used to form nonce value for every proper transmission (no retransmission). Firmware maintains the packet counter per connection as per the specification rules. This is part of encryption feature.	0x0000

Field	Bit	Description	Reset
Packet_counter[15:0]	15:0	Lower 16-bits of the packet counter value passed as part of Nonce for the packet to be encrypted.	0
Field	**Bit**	**Description**	**Reset**
Packet_counter[31:16]	15:0	Middle 16-bits of the packet counter value passed as part of Nonce for the packet to be encrypted.	0

Field	Bit	Description	Reset
Unused	15:8		
Packet_counter[39:32]	7:0	Upper 8 bits of the packet counter value passed as part of Nonce for the packet to be encrypted.	0

Master Initialization Vector0-1 register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	**32-bit**				
0x208-0x20A	0x410-0x414	IV_MASTER0-IV_MASTER1	RW	Initialization vector 31:0 for master	0x0000

Field	Bit	Description	Reset
IV_master[15:0]	15:0	This is the lower 16-bits of the IVm field, which contains the master’s portion of the initialization vector.	0
Field	**Bit**	**Description**	**Reset**
IV_master[31:16]	15:0	This is the upper 16-bits of the IVm field, which contains the master’s portion of the initialization vector.	0

Slave Initialization Vector0-1 register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	**32-bit**				
0x20C-0x20E	0x418-0x41C	IV_SLAVE0-IV_SLAVE1	RW	Initialization vector 31:0 for Slave	0x0000

Field	Bit	Description	Reset
IV_slave[15:0]	15:0	This is the lower 16-bits of the IVs field, which contains the slave's portion of the initialization vector.	0
Field	**Bit**	**Description**	**Reset**
IV_slave[31:16]	15:0	This is the upper 16-bits of the IVs field, which contains the slave's portion of the initialization vector.	0

Encryption key register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	**32-bit**				
0x210-0x21E	0x420-0x43C	ENC_KEY0-7	WO	The encryption key / session key which is used in ECB encryption, CCM encryption and CCM decryption.	0x0000

Input data register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	**32-bit**				
0x220-0x23A	0x440-0x474	INPUT_DATA0- INPUT_DATA13	WO	27-byte plain text field for ECB encryption, CCM encryption and cipher text field for CCM decryption. Write-only register fields.	0x0000

Address	Field	Bit	Description	Reset
0x220	Input_data[15:0]	15:0	Bytes 1 and 0 of the input plain text to be encrypted.	0
0x222	Input_data[31:16]	15:0	Bytes 3 and 2 of the input plain text to be encrypted.	0
…	…	…	…	…

Table continues on the next page...

0x238	Input_data	15:0	Byte 25 and 24 of the input plain text to be encrypted	
0x23A	Input_data	7:0	Byte 26 of input plain test	

Output data register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x220-0x23A	0x440-0x474	OUTPUT_DATA0- OUTPUT_DATA13	RO	It is the plain text field for CCM encryption and cipher text field for CCM decryption. Read-only register fields.	0x0000

Address	Field	Bit	Description	Reset
0x220	Output_data[15:0]	15:0	Bytes 1 and 0 of the decrypted plain text/ encrypted text.	0
0x222	ouput_data[31:16]	15:0	Bytes 3 and 2 of the decrypted plain text/ encrypted text.	0
…	…	…	…	…
0x238	output_data	15:0	Bytes 25 and 24 of the decrypted plain text/ encrypted text.	
0x23A	output_data	7:0	Bytes 26 of the decrypted plain text/ encrypted text.	

MIC input register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0x23C-0x23E	0x478-0x47C	MIC_IN0-MIC_IN1	RW	It is the encrypted MIC received from transmitter for CCM decryption and authentication. This bit is implemented as RW to facilitate the tester behavior	0x0000

Field	Bit	Description	Reset
Mic_in[15:0]	15:0	This is the lower 16-bits of the MIC field used for CCM decryption.	0

Field	Bit	Description	Reset
Mic_in[31:16]	15:0	This is the upper 16-bits of the MIC field used for CCM decryption.	0

MIC output register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x240-0x242	0x480-0x484	MIC_OUT0-MIC_OUT1	RO	It is the encrypted MIC generated by the transmitter during CCM encryption.	0x0000

Field	Bit	Description	Reset
Mic_out[15:0]	15:0	This is the lower 16-bits of the MIC generated during CCM encryption.	0
Field	**Bit**	**Description**	**Reset**
Mic_out[31:16]	15:0	This is the lower 16-bits of the MIC generated during CCM encryption.	0

Encryption parameters register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x244	0x488	ENC_PARAMS	RW	Parameters required for encryption like length, LLID of the input data.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use.	0
Direction	7	The *directionBit* shall be set to ‘1’ for Data Channel PDUs sent by the master and set to ‘0’ for Data Channel PDUs sent by the slave .	0
Payload length	6:2	Length of the input data.	0
Data PDU header	1:0	LLID of the packet.	0

Encryption configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x248	0x490	ENC_CONFIG	RW	Encryption block configuration. Configuration and control information for the operation of encryption block.	0x0000

Field	Bit	Description	Reset
Reserved	15:3	Reserved for future use.	0
Dec/enc	2	Decryption/Encryption. 1. Encrypt 2. Decrypt	0
Ecb/ccm	1	Ox1 ECB 0x0 CCM	0
Start_proc	0	0x1 start the AES processing	0

Encryption interrupt enable register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x24C	0x498	ENC_INTR_EN	RW	Encryption interrupt enable register. Controls interrupts from encryption block.	0x0000

Field	Bit	Description	Reset

Table continues on the next page...

Reserved	15:3	Reserved for future use.	0
Ccm_proc_intr_en	2	CCM processed interrupt enable. 1 – enable 0 – disable	0
Ecb_proc_intr_en	1	ECB processed interrupt enable. • Enable 0 – disable	0
Auth_pass_intr_en	0	Authentication interrupt enable. 1 – Enable 0 – Disable	0

Encryption interrupt clear register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x250	0x4A0	ENC_INTR_CLEAR	WO	Encryption interrupt clear register. Clear one or more interrupts set by the encryption block.	0x0000

Field	Bit	Description	Reset
Reserved	15:4	Reserved for future use.	0
In_Data_Clear	3	Clears the input data. Used for Zero padding of encryption for less than block sized data.	0
Ecb_proc_intr_clear	1	ECB processed interrupt clear. Writing 1 to this register clears the interrupt.	0
Auth_pass_intr_clear	0	Authentication interrupt clear. Writing 1 to this register clears the interrupt.	0

Encryption interrupt status register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x250	0x4A0	ENC_INTR_STATUS	RO	Encryption interrupts status register. Indicates	0x0000

				one or more interrupts set by the encryption block.	

Field	Bit	Description	Reset
Reserved	15:3	Reserved for future use.	0
Ccm_proc_intr	2	CCM processed interrupt.	0
Ecb_proc_intr	1	ECB processed interrupt.	0
Auth_pass_intr	0	Authentication interrupt. 0x1- indicates MIC matched 0x0 –indicated MIC mismatched	0

DTM control register

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0xB8	0x170	LE_RF_TEST_MODE	RW	LE Direct Test Mode (DTM) configuration and control register. Used to control the direct test mode (DTM) operation.	0x0000

Field	Bit	Description	Reset
Test_length[5:0]	15:10	0x00-0x25 Length in bytes of payload data in each packet 0x26-0xFF Reserved for future use	0
Pkt_payload[2:0]	9:7	Payload type as per the HCI parameter. 0x00 Pseudo-Random bit sequence 9 0x01 Pattern of alternating bits ‘11110000 0x02 Pattern of alternating bits ‘10101010’ 0x03 Pseudo-Random bit sequence 15	0

Table continues on the next page...

		0x04 Pattern of All ‘1’ bits 0x05 Pattern of All ‘0’ bits 0x06 Pattern of alternating bits ‘00001111 0x07 Pattern of alternating bits ‘0101’ 0x08-0xFF Reserved for future use	
Test_type	6	Mixed Info Field. 1 – Indicates DTM test ON 0 – Indicates DTM test OFF	0
Test_frequency[5:0]	5:0	N = (F – 2402) / 2 Range: 0x00 – 0x27. Frequency Range : 2402 MHz to 2480 MHz	0

DTM receive packet count register

Addr	**Addr**	**Register Name**	**RW**	**Description**	**Reset**
16-bit	**32-bit**				
0xBA	0x174	DTM_RX_PKT_COUNT	RO	Count of the number of LE packets received when device is configured in receive test mode.	0x0000

Field	**Bit**	**Description**	**Reset**
Rx_packet_count[15:0]	15:0	Number of packets received in receive test mode.	0

Connection channel test control register

Addr	**Addr**	**Register Name**	**RW**	**Description**	**Reset**
16-bit	**32-bit**				
0xBC	0x178	CONN_TEST_CONTROL	RW	Connection test control register. To introduce test behavior in the operation of connection.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use	0
Nonempty_pdu_rxnack	7	NACK the received packet, if rx packet is a NON-EMPTY PDU. • Always NACK (no acknowledgement) 0 - No change from normal behavior	0
Empty_pdu_rxnack	6	NACK received packet, if rx packet is an EMPTY PDU. • Always NACK 0 - No change from normal behavior	0
Nonempty_pdu_retx	5	Retransmit previous transmitted non-empty PDU irrespective of received NESN (whether acknowledged or not). 1 – Always retransmit 0 – No change from normal behavior	0
empty_pdu_retx	4	Retransmit previous transmitted empty PDU irrespective of received NESN (whether acknowledged or not). 1 – Always retransmit 0 – No change from normal behavior	0
Nonempty_crc_err	3	Cause CRC field error in transmitted non-empty PDU (only). 1 – Causes CRC error 0 – no change from normal behavior	0
empty_crc_err	2	Cause CRC field error in transmitted empty PDU (only). 1 – Causes CRC error 0 – no change from normal behavior	0
Nonempty_acc_err	1	Causes Access address error in transmitted non-empty PDU. 1 – Causes access address error 0 – no change from normal behavior	0

Table continues on the next page...

empty_acc_err	0	Causes Access address error in transmitted empty PDU. 1 – Causes access address error 0 – no change from normal behavior	0

Advertising channel test control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xBE	0x17C	ADVCH_TEST_CONTROL	RW	Advertising channel test control register. To introduce advertising channel test control operation.	0x0000

Field	Bit	Description	Reset
Reserved	15:7	Reserved for future use	0
Scan_tx_hdr	6	Corrupt the tx header of SCAN_REQ packet 1 – corrupt the address 0 – no change	
Peer_addr_err	5	Corrupt the peer address field in the SCAN_REQ packet 1 – corrupt the address 0 – no change	0
Rcv_txaddr_err	4	Corrupt the received transmit address type indication 1 - corrupt the tx address type (invert the bit) 0 – no change	0
Scnrsp_tx_err	3	Introduce CRC error in scan response packet. 1 – corrupt CRC 0 – no change	0
Rx_crc_err	2	Receive packet CRC error indication. 1 – Indicate CRC error of received packet, irrespective of good/bad CRC	0

Table continues on the next page...

		0 – No change from normal behavior	
Tx_crc_err	1	Corrupt transmit packet CRC, irrespective of packet type. • Corrupt CRC 0 – No change in normal behavior This is done by corrupting the crc_init value.	0
Tx_acc_err	0	Corrupt transmit packet access address. • Corrupt access address 0 – No change in normal behavior	0

DTM Error Count

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xC0	0x180	DTM_CRC_ERR_COUNT	RO	Indicates number of packets received with CRC error in the DTM mode	0x0000

Field	Bit	Description	Reset
dtm_crc_err_pkt_count	[15:0]	DTM CRC error packet count. Used for Debug purpose only.	0

Channel Address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xC4	0x188	CH_ADDR/ TXRX_HOP	RO	Contains value of the transmit and receive hop frequency	0x0000

Field	Bit	Description	Reset
reserved	15	Reserved for future use	0

Table continues on the next page...

hop_ch_rx	[14:8]	Receive channel index. Channel index on which previous packet is received.	0
reserved	7	Reserved for future use	0
hop_ch_tx	[6:0]	Transmit channel index. Channel index on which previous packet is transmitted.	0

Divider Value Register for SCA

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x500	0xA00	DIV_VAL_ADDR	RW	This value will go to divide by N module to derive various sleep clock frequencies	0x1F3C

Reference Clock	Divider Value	Output Sleep Frequency(KHz)	PPM	SCA	
	Decimal				Hex
16.384 KHz	7808	1E80	16.39351	580.59674	
7809	1E81	16.39135	448.73375	0	
7810	1E82	16.38920	317.56129	0	
7811	1E83	16.38722	196.25739	1	
7812	1E84	16.38501	61.54619	4	
7813	1E85	16.38287	-69.19713	4	
7814	1E86	16.38097	-185.16635	1	
7815	1E87	16.37870	-323.70446	0	
7816	1E88	16.37668	-446.48984	0	
16 KHz	7996	1F3C	16.00811	507.13706	
7997	1F3D	16.00610	380.94506	0	
7998	1F3E	16.00414	258.94704	0	
7999	1F3F	16.00221	137.93902	2	
8000	1F40	15.99992	-4.79998	7	
8001	1F41	15.99786	-133.74211	2	
8002	1F42	15.99585	-259.13283	0	
8003	1F43	15.99392	-380.01553	0	
	8004	1F44	15.99192	-505.02482	

A macro ‘SCA_DRIFT “is used to select the clock source for the Sleep Clock.

When the SCA_DRIFT macro is enabled then the sleep clock source is derived from the 64MHz based on the register DIV_VAL_ADDR setting. Otherwise sleep clock source is the on board 32.768 KHz

From 64 MHz of RF clock either 16 KHz or 16.384 KHz clock is derived using DCM (Digital Clock Manager, clock multiplier by 2) and a clock divider (divide by N). 64 MHz clock is given to DCM and this multiplied clock of 128 MHz is given to divide by N module. With 64 MHz clock only, it was not possible to cover lower, middle and upper range of required SCA (0 to 7) that led to use a DCM to get a higher clock and desired SCA values.

Clock divider value is given into a Programmable Read/Write register DIV_VAL_ADDR, whose address is 0x500 for 16 bit address bus.

Config.xml file is having all power up configuration parameters for BlueLitE IP. This will be loaded into Hardware after Reset.

For the validation, In order to introduce SCA other than ‘0’ which is by default configuration of BlueLitE IP, one has to change corresponding “sca” field in config.xml file also. E.g. if one has selected SCA of 4 from table 2 given, then DIV_VAL_ADDR has to be programmed with 0x1E84 for positive ~61 PPM or 0x1E85 for negative ~69 PPM and “sca” field in config.xml has to be updated with 0x04.

Whenever SCA value is chosen from sleep clock frequency of 16 KHz, at that time internal adjustment for sleep clock frequency is not required. For that “clock_config” field in config.xml has to be updated with 0xA020.

Packet configuration Register for additional Preamble and Payload

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xFE	0x1FC	PKT_PROC_CONFIG	RW	Packet processing configuration register for additional preamble and payload	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use	0

Table continues on the next page...

<table>
<tr><td>Dummy_pld_ctrl</td><td>7</td><td>Disable/Enable additional payload , 1 - Enable, 0 - Disable</td><td>0</td></tr>
<tr><td></td><td>[6:4]</td><td><table>
<tr><td>Bit value [6:4]</td><td># of Dummy Payload bits</td></tr>
<tr><td>000</td><td>1</td></tr>
<tr><td>001</td><td>2</td></tr>
<tr><td>010</td><td>3</td></tr>
<tr><td>011</td><td>4</td></tr>
<tr><td>100</td><td>5</td></tr>
<tr><td>101</td><td>6</td></tr>
<tr><td>110</td><td>7</td></tr>
<tr><td>111</td><td>8</td></tr>
</table></td><td>0</td></tr>
<tr><td>Preamble_ctrl</td><td>3</td><td>Disable/Enable additional preamble, 1 - Enable, 0 - Disable</td><td>0</td></tr>
<tr><td></td><td>[2:0]</td><td><table>
<tr><td>Bit value [2:0]</td><td># of Dummy Preamble bits</td></tr>
<tr><td>000</td><td>1</td></tr>
<tr><td>001</td><td>2</td></tr>
<tr><td>010</td><td>3</td></tr>
<tr><td>011</td><td>4</td></tr>
<tr><td>100</td><td>5</td></tr>
<tr><td>101</td><td>6</td></tr>
<tr><td>110</td><td>7</td></tr>
<tr><td>111</td><td>8</td></tr>
</table></td><td>0</td></tr>
</table>

Enable DUMMY_BITS, DUMMY_PREAMBLE and DUMMY_PLD based on requirements. By default they are disabled

Receive trigger control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0XFC	0X1F8	RECEIVE_TRIG_CTRL	RW	Receivers trigger control register. The threshold value for access address match, the access match trigger	0x0000

				timeout is programmed to this register by firmware.	

Field	Bit	Description	Reset
Acc_trigger_timeout	15:8	If access address match does not occur then within this time from the start of receive operation, the receive operation times out and stops. An internal counter value of 1usec resolution is continuously compared with the value programmed. Max value :0Xff	0
tim_adj_trig	7	Not used –RFU	0
	6	RFU	
Acc_trigger_threshold	5:0	Access address match threshold value. Number of bits of access address that should match with the expected access address to trigger an access code match. Max value : 32 (for 32-bit access address) Lower values may be programmed for bad radios or channels but care must be taken to ensure there are no 'false' matches due to reduced number of bits required to match. Note : BQB spec mandates this to be 32. So ensure that the standard versions have 32. For debugging or RF tuning, we can experiment with smaller values.	0

Transmit/Receive data delay Register

0xC8	0x190	TX_RX_ON_DELAY	RW	Controls the delay in link layer from internal reference point, to start the transmit and receive operation.	0x0000

Field	Bit	Description	Reset
Txon_delay[7:0]	15:8	Transmit delay – Delay from internal trigger of transmit to transmission of first bit on air. It is used to control the T_IFS. The delay is in resolution of 1 microsecond.	0x00
Rxon_delay[7:0]	7:0	Receive delay – Delay from start of receive to expected first bit of receive packet at the controller. Used to control the turn on time of radio to optimize on power. The delay is in resolution of 1 microsecond.	0x00

Transmit/receive synthesizer delay register

0xCC	0x198	TX_RX_SYNTH_DELAY	RW	Controls the link layer behavior after waiting for RF synthesizer settling time. The delay relates to the PLL settling time, from the trigger to start transmit or receive operation. The action to be taken at the expiry of this delay is specific to each radio (e.g. we may set an external control pin to the radio).	0x0000

Field	Bit	Description	Reset
Tx_synth_delay[7:0]	15:8	Transmit synthesizer delay – Delay from start of transmit to stabilization of PLL. This may be used if any radio specific operation needs to be performed after PLL stabilization before bit transmission.	0x00
Rx_synth_delay[7:0]	7:0	Receive synthesizer delay – Delay from start of receive to receiver synthesizer stability.	0x00

Radio Control Register

0xD6	0x1AC	RADIO_CONTROL / RADIO_CNTRL	RW	Register to initiate the radio register read/write operation. A write to this register always initiates a radio register write or read operation depending on whether the radio_rw bit is set or cleared. Mixed-info-field/ register	0x0000

Field	Bit	Description	Reset
Radio_rw	15	To configure start of radio register read or write. If this bit is set, a radio register write is initiated. If this bit is cleared, radio register read is initiated. 0 – Radio register read 1 – Radio register write Mixed-info-field/register	0x0
Radio_reg_addr[6:0]	14:8	Address of the radio register to be read or written Mixed-info-field/register	0x00
Radio_reg_data[7:0]	7:0	Lower 8 bits of data to be written to radio register Mixed-info-field/register	0x00

Caution:

1. For More than 8-bit data, the Radio Data Register should be updated before writing this register during radio register write operation.

Radio Control Register

0xD6	0x1AC	RADIO_ACCESS/ RADIO_CNTRL	RO	Indicates the status of the radio register write or read operations. Mixed-info-field/ register	0x0000

Field	Bit	Description	Reset
Reserved	15:11		XX
rssi_rd_sts	10	When set to 1, indicates that RSSI register is updated with the latest value. The bit is cleared when this register is read. Mixed-info-field/register *Clear-on-Read-field*	0x0
si_done_host_sts	9	The host initiated radio register access(write or read) is completed. The bit is cleared when this register is read. Mixed-info-field/register *Clear-on-Read-field*	0x0
Rif_pos_over	8	Deprecated-Do-Not-Use	0x0
Radio_reg_data[7:0]	7:0	Lower 8 bits of data read from the radio register Mixed-info-field/register	0x00

Radio Select Register

0xD8	0x1B0	RADIO_SELECT/ RADIO_SEL	RW	Register to select the radio. This register also has fileds which are used to configure the RF Test parameters Mixed-info-field/ register	0x0000

Field	Bit	Description	Reset
Reserved	15:3		XX
Radio_sel[2:0]	2:0	The Radio select value of the radio mounted currently. This is done at power on by firmware. The firmware will check the radio mounted on the board by reading some dip switch settings and accordingly program the radio_sel value to the BB. Mixed-info-field/register	00

RSSI Register

0xD8	0x1B0	RADIO_RSSI_READ	RO	Indicates the RSSI value read from the radio for the last packet received. Mixed-info-field/ register	0x0000

Field	Bit	Description	Reset
Rssi_data	15:0	Indicates the RSSI value read from the radio for the last packet received. The meaning is specific to the RF IC used. Mixed-info-field/register	0x0

Radio Data Register

0xDA	0x1B4	RADIO_DATA	RW	Register to configure the data to be written to the radio register. During radio register read, the data read from the radio register is available in this register.	0x0000

Field	Bit	Description	Reset
Reserved	15:8		XX
Radio_reg_data[15:8]	7:0	Radio_reg_data[15:8] – higher byte of the value to be written to the radio register or higher byte of the value read from the radio register.	0x00

Radio Configuration Register

0xDC	0x1B8	RADIO_CONFIG/ RADIO_CONFG	RW	Register to program the configurable parameters for the selected radio.	0x0004

Field	Bit	Description	Reset
Reserved	15:8		
Radio_config	7:0 2 1:0	Configurable parameters of Radio. This register may be used to change any real time configuration parameters of the radio, while programming the radio at the beginning or end of transmit/receive operation. Selects mt_custom_radio2 or Rudraneel radio 1 – Rudraneel radio 0 – mt_custom_radio2 Selects modulation scheme, for LE it is 2’b0	0x0004

Radio Configuration Register

0x12E	0x25C	RADIO_CNFG_REG	RW	Register to program the configurable parameters for the selected radio.	0x884E/ 0x81BE

Field	Bit	Description	Reset
Radio_cnfg_reg	7:0	Configurable parameters of Radio. This register may be used to change any real time configuration parameters of the radio, while programming the radio at the beginning or end of transmit/receive operation. Reset value depends on selected radio. .	0x884E/ 0x81BE

DPLL Configuration Register

0x12C	0x258	DPLL_CONFIG/ RADIO_OP_CONTROL	RW	Register to program the configurable parameters for the selected radio.	0x72E0

Field	Bit	Description	Reset
Dpll_config_init	15:12	Sets upper water mark for DPLL	7
	11:8	Sets lower water mark for DPLL	2
	7:0	DPLL operation config value	0xE0

SPI Mode Control Register

0x132	0x264	SPI_MOD_CNTRL_REG	WO	Register to program the SPI mode control register values	0x04BF

Field	Bit	Configurable value		Description	Reset
A-D Delay	15:14	00	No Delay	Address to Data Delay – These two bits indicate the interval between the address and data transfers	0x04BF
		01	One SCLK Delay		
		10	Two SCLK Delay		
		11	Three SCLK Delay		
M/L F	13	0	MSB First	MSB/LSB First. Indicates whether the shifting of bits happens in MSB First fashion or LSB First fashion	
		1	LSB First		
A/D F	12	0	Address First	Address/Data First. Indicates whether the address is TX first or Data is TX first	
		1	Data First		
Number of address bits	11:7	00000	Width of SPI Address – 0	Indicates the width of the SPI Address (Variable in the range 0 to 16 bits). The firmware has to make sure it writes a valid value i.e. in the range	
		00001	Width of SPI Address – 1		
		10000	Width of SPI Address – 16		
Number of DATA bits	6:2	00000	Width of SPI Data – 1	Indicates the width of the SPI Data (Variable in the range 1 to 32 bits)	
		00001	Width of SPI Data – 2		
		11111	Width of SPI Data – 32		

Table continues on the next page...

3W/4W	1	0	SPI configured as a 4-wire Interface	3-wire/ 4-wire Interface	
		1	SPI configured as a 3-wire Interface		
SPE	0	0	SPI system is disabled	Generic SPI Enable/Disable. The SPI remains in an idle state when it is disabled	
		1	SPI system is enabled		

SPI Control Register

0x134	0x268	SPI_CNTRL_REG	WO	Register to program the SPI clk control and enable control Register values to perform SPI task	0x1207

Field	Bit	Configuration value	Description	Reset
Reserved	15:14			0x1207
C-D Delay	13	0	The first bit is shifted out before the first clock edge	Clock to Data Delay – This bit indicates the delay between the first clock edge and the shifting of first bit during SPI operation
		1	The first bit is shifted out after the first clock edge	
ECC	12	0	Extra Clock cycle Disabled	Extra Clock Cycle – this bit is used to enable/ disable an extra SCLK cycle after all the SPI address and data bits are TX.
		1	Extra Clock cycle enabled	
SE	11	0	The bits are shifted on the falling edge of SCLK	Shifting Edge. The bits are shifted on the shifting edge as configured in this bit
		1	The bits are shifted on the rising edge of SCLK	
E-C A Delay	10:9	00/11	The 'enable' assertion and the first clock edge happen together	Enable-Clock Assertion Delay. These 2bits are used to configure the delay between assertion of 'enable' and first SPI clock edge
		01	The first clock edge occurs half SCLK period after the 'enable' assertion	
		10	The first clock edge occurs one SCLK period after the 'enable' assertion.	

Table continues on the next page...

CIS	8	0	Clock is low(0) in its idle state	SPI Clock Idle State	
		1	Clock is high(1) in its idle state		SPIClockishigh(1)initsidlestate
DE A-D	7	0	No De-Assertion	De-Assertion of Enable during SPI address to Data transfer interval	

TEDR	6	0	Toggle Disabled	Toggle Enable during read – This bit indicates that the enable should be toggled between SPI address and data transfers only during a SPI read operation	
		1	Toggle Enabled		
DES	5	0	The Enable will be in its active state both during the address and data transfer phases	Dual Enable State – This bit is mainly provided for the L3 interface. It should be set when the Generic SPI is to be configured as a L3 interface	
		1	The Enable will be in its active state during the address transfer only. It will		

Table continues on the next page...

			be in its idle state during data transfer.		
EDP	4:3	00	No De-Assertion	Enable De-Assertion Period between Continuous SPI Transfers	
		01	Half SCLK period		
		10	One SCLK Period		
		11	Two SCLK Periods		
E-C D Delay	2:1	01	The 'enable' deassertion happens half SCLK period after the last clock edge.	Enable-Clock Deassertion Delay. These 2bits are used to configure the delay between last SPI clock edge and deassertion of 'enable' signal in SPI FSM	
		10	The 'enable' deassertion happens one SCLK period after the last clock edge.		
		11	The 'enable' deassertion happens half SCLK period before the last clock edge.		
EIS	0	0	Enable is low(0) in its idle state	Enable Idle State of SPI FSM	
		1	Enable is high(1) in its idle state		

External RSSI register

0x520	0xA40	EXT_RSSI_REG	RW	Register to program the enable control for sending the External RSSI instead of the internal RSSI from Modem	0x0000

Field	Bit	Description	Reset
Reserved	15:1		
ext_rssi_sel	0	When set to 1, ext_rssi[15:0] value is written to the Rx FIFO along with the packet	0

RF_ACTIVE_PERIOD register

0x5A	0x0B4	rf_active_period	WO	• Register to specify the time offset before the start of a transceiver operation (i.e., a Tx/Rx) at which "RF_ACTIVE" output signal shall be asserted by the LLH. • • The purpose of the “RF_ACTIVE” signal is to indicate to the host of any upcoming transceiver activity. The host may schedule not to do any power intensive operations during this time to reduce the system peak power. • • The time offset is specified in the units of BT slots (625us) • • Note – The RF_ACTIVE signal will be asserted at the specified time offset before a transceiver operation and remain asserted till the end of the operation.	0x0000

Field	**Bit**	**Description**	**Reset**
Reserved	15:7	RFU	xx
rf_active_slots	6	Polarity bit for rf_active 0: active high 1: active low.	
rf_active_slots	5:0	Number of BT slots (625 us) in advance of the actual start of the slot (tx/Rx) to assert rf_active. ”rf_active” is always de-asserted when the value is 6’h0 (all zeros) and always asserted when the value is 6’h3F(all ones) Therefore the Min. value is .625ms and the Max value is 62*. 625ms.	0

note: MODEM REGISTERS WILL NOT BE ACCESSIBLE IN CONTROLLER (LEC_LLH) ENVIRONMENT. THEY WILL BE ACCESSIBLE FROM LEC_TOP (LEC_LLH + LEC_PHY) INTEGRATED ENVIRONMENT.

Device Identification – Lower Word

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**
0X600	0XC00	MOD_DEV_ID_LW	RO	This defines the lower word of device ID for the modem chip.	0x0001

Field	**Bit**	**Description**	**Reset**
MOD_DEV_ID_LW	15:0	This defines the lower word of device ID for the modem chip.	0x0001

Device Identification – Upper Word

Addr **16-bit**	**Addr** **32-bit**	**Register Name**	**RW**	**Description**	**Reset**

Table continues on the next page...

0X602	0XC04	MOD_DEV_ID_UW	RO	This defines the upper word of device ID for the modem chip.	0x0000

Field	Bit	Description	Reset
MOD_DEV_ID_UW	15:0	This defines the upper word of device ID for the modem chip.	0x0000

RSSI Value – Lower Word

Addr 16-bit	Addr -bit	Register Name	RW	Description	Reset
0X604	0XC08	RSSI_VAL_LW	RO	This defines the lower word of the RSSI value measured.	0x0000

Field	Bit	Description	Reset
RSSI_VAL_LW	15:0	This defines the lower word of the RSSI value measured in dB	0x0000

RSSI Value – Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X606	0XC0C	RSSI_VAL_UW	RO	RFU.	0x0000

Field	Bit	Description	Reset
RSSI_VAL_UW	15:0	RFU	0x0000

Transmit Function Control – Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X608	0XC10	TX_FUNC_CNTL_LW	RW	This defines various transmit functionality	0x0000

Field	Bit	Description	Reset
RFU	15:3	Reserved for future use	0x000
FORMAT	2	0 -2's Complement 1 – Offset Binary	0
POLAR_INV	1	The bit indicates whether phase inversion needs to be performed for the polarity. 0 – Phase inversion is disabled. 1 – Phase inversion is enabled.	0
RECT_POLAR	0	The bit indicates the RF platform is rectangular / polar modulator. 0 – Rectangular modulator. 1 – Polar modulator.	0

Transmit Function Control – Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X60A	0XC14	TX_FUNC_CNTL_UW	RW	RFU	0x0000

Field	Bit	Description	Reset
RFU	15:0	Reserved for future use	0x0000

Transmit Intermediate Frequency - Modulation Index Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X60C	0XC18	TX_IFMD_CNTL_LW	RW	This defines the transmit modulation index control	0x0000

Field	Bit	Description	Reset
RFU	15:14	Reserved for future use	0

Table continues on the next page...

TX_IF_FREQ_PHASE_VAL	13:0	This indicates the transmit frequency phase value for IF conversion during transmit in 8.6 format.	0

Transmit Intermediate Frequency - Modulation Index Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X60E	0XC1C	TX_IFMD_CNTL_UW	RW	This defines the transmit modulation index control	0x0000

Field	Bit	Description	Reset
RFU	15:11	Reserved for future use	0
MOD_INX_PHASE_VAL	10:0	Modulation index value for transmit operation in 5.6 format.	0

Receive Function Control – Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X61C	0XC38	RX_FUNC_CNTL_LW	RW	This defines various Receiver Functionality	0x0C28

Field	Bit	Description	Reset
RSSI_THRESHOLD	15:8	This defines the RSSI threshold .	0X0C
RFU	7:4	Reserved for future use.	0X5
Filter Select	3	To select between 700 KHz and 850 KHz filter 1 – Select 850 KHz filter 0 – Select 700 KHz filter	0
RFU	2	Reserved for future use	0
EL_FULL_PKT	1	This bit enables / disables early late tracking for full packet. 0 – Early late full packet tracking enabled.	0

Table continues on the next page...

		1 – Early late full packet tracking disabled.	
REAL_COMPLEX	0	This bit defines the input type. 0 – Real input. 1 – Complex input.	0

Receive Function Control – Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X61E	0XC3C	RX_FUNC_CNTL_UW	RW	Upper byte defines the Phase Variation Reference Value Lower byte defines the Delta threshold limit for receiver algorithm	0x28C8

Field	Bit	Description	Reset
Phase_Variation_Reference	15:10	This defines the Phase Variation Reference Value	0x0A
DELTA_THRES_VAL	9:0	This defines the delta threshold for receiver algorithm in 8-bit 2's complement format	0xC8

Receive Intermediate Frequency - Mean value Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X620	0XC40	RX_IFMV_CNTL_LW	RW	This indicates the receiver frequency phase value for IF conversion during transmit in 8.6 format.	0x0000

Field	Bit	Description	Reset
RFU	15:14	Reserved for future use	0
RX_IF_FREQ_PHASE_VAL	13:0	Receiver IF frequency value in 8.6 formats.	0

Receive Intermediate Frequency - Mean value Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X622	0XC44	RX_IFMV_CNTL_UW	RW	RFU	0x006E

Field	Bit	Description	Reset
RFU	15:0	Reserved for future use	0x006E

Modem Control – Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X624	0XC48	MOD_CNTL_LW	RW	This defines the various modem control fields	0x0000

Id	Bit	Description	Reset
RFU	15:7	Reserved for future use	0
SPI_INTF_16_32	6	SPI interface for 16 / 32 bit. 0 - 16 bit interface. 1 – 32 bit interface.	0
DEEP_SLEEP_POL	5	This bit indicates the polarity of the deep sleep mode signal for external platform. 0 – Active Low. 1 – Active High.	0
SLEEP_POL	4	This bit indicates the polarity of the sleep mode signal for external platform. 0 – Active Low. 1 – Active High.	0
SM_CNTL	3	This bit controls the sleep mode control from modem. 0 - Sleep mode control from modem is disabled. 1 - Sleep mode control from modem is enabled.	0
RFU	2	Reserved for future use .	0
RFU	1	Reserved for future use.	0

Table continues on the next page...

RST_POLARITY	0	This register controls the polarity of the reset for RF platform. 0 - Active Low. 1 - Active High.	0

Modem Control – Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW		Reset
0X626	0XC4C	MOD_CNTL_UW	RW	RFU	0x0000

Field	Bit	Description	Reset
RFU	15:0	Reserved for future use	0x0000

Modem Operation – Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X628	0XC50	MOD_OPER_LW	RW	This register defines various modem operation fields	0x0000

Field	Bit	Description	Reset
RFU	15:4	Reserved for future use	0
DYN_PWR_CNTL	3	This bit enables dynamic power control. Whenever Tx or Rx operation is idle and clock gating of modem is enabled, it will disable the Tx or Rx core clock whenever the bit is set. 0 - Dynamic power control is disabled. 1 - Dynamic power control is enabled.	0
SM_ENB	2	This bit is used to enable sleep mode. 0 - Sleep mode is disabled. 1 - Sleep mode is enabled.	0

Table continues on the next page...

RX_ENB	1	This bit is used to enable receive. Only effective when Modem works as standalone. If integrated with Link Layer H/W, then Link Layer H/W's receive enable bit is considered and this bit is masked. 0 - Receive is disabled. 1 - Receive is enabled.	0
TX_ENB	0	This bit is used to enable transmit. Only effective when Modem works as standalone. If integrated with Link Layer H/W, then Link Layer H/W's transmit enable bit is considered and this bit is masked. 0 - Transmit is disabled. 1 - Transmit is enabled.	0

Modem Operation – Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X62A	0XC54	MOD_OPER_UW	WO	This defines the trigger pattern for Software Reset generation	0x0000

Field	Bit	Description	Reset
SW_RST	15:0	This register is used for controlling software reset of the modem IP. This is a pattern based write register which triggers a write based on the following pattern (16'hABCD). The pattern can be redefined by modifying RB_SOFT_RESET_PATTERN parameter in modem Reg bank based on requirement.	0x0000

AGC control - Lower Word

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0X62C	0XC58	AGC_CNTL_LW	RW	This defines gain control of various blocks	0x0000

Field	Bit	Description	Reset
RFU	15:9	Reserved for Future Use	0
SEL	8	0 – Automatic Gain Control 1 – Manual Gain Control	0
RFU	7:4	Reserved for Future Use	0
LNA	3	1-bit LNA gain control if mode is manual	0
MIXER	2	1-bit Mixer gain control if mode is manual	0
VGA	1:0	2-bit VGA gain control if mode is manual	0

AGC control - Upper Word

Addr	Addr	Register Name	RW	Description	Reset
16-bit	32-bit				
0X62E	0XC5C	AGC_CNTL_UW	RW	Controls RSSI and Saturation threshold	0x0140

Field	Bit	Description	Reset
RFU	15:9	Reserved for Future Use	0
AFC timer	8:3	Timer set to freeze the AFC algorithm and preamble search algorithm	0x28
AFC Timer Enable/Disable	2	Control bit to enable the AFC timer 1 – Enable 0 - Disable	0
iAgcAfc Enable/Disable	1	iAgcAfcCntl (RSSI detect second threshold) 1 – Enable 0 - Disable	0
iAgcRssiDet Enable/Disable	0	iAgcRssiDet (RSSI detect primary threshold) 1 – Enable 0 - Disable	0

AFC control - Lower Word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X630	0XC60	AFC_CNTL_LW	RW	This defines the frequency offset control mode and offset frequency phase IF value	0x0000

Field	Bit	Description	Reset
SEL	15	0 – Manual Frequency Offset Control 1 - Automatic Frequency Offset Control	0
RFU	14	Reserved for Future Use	0
FREQ_OFF_PHASE_VAL	13:0	This indicates the frequency phase value to be corrected for manual frequency offset control mode, in 8.6 format.	0

AFC control - Upper Word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X632	0XC64	AFC_CNTL_UW	RW	This defines preamble search threshold	0x0080

Field	Bit	Description	Reset
RFU	15:12	Reserved for Future Use	0
PRE_SEARCH_THRESHOLD	11:0	This indicates the preamble search threshold during correlation.	0x080

Command	Opcode	Description
STOP_ADV	0x41	Stop advertiser operation.
STOP_SCAN	0x43	Stop the scanner operation.
		The associated configuration registers must be programmed before the command is issued. [refer to llh_set_adv_parameters function in lec_llh/simulation/llh_driver/ ll_hardware_api.c file] [refer to llh_set_scan_parameters function in lec_llh/simulation/llh_driver/ ll_hardware_api.c file]

Table continues on the next page...

START_INIT	0x44	Start connection creation operation. The associated configuration registers must be programmed before the command is issued. [refer to create_connection function in lec_llh/simulation/llh_driver/ll_hardware_api.c file]
STOP_INIT	0x45	Cancel connection creation operation.
DTM_TX_START	0x46	Start Direct Test Mode Transmit Test. The associated configuration registers must be programmed before the command is issued. [refer to dtm_tx_test function in lec_llh/simulation/llh_driver/ll_hardware_api.c file]
DTM_RX_START	0x47	Start Direct Test Mode Receive Test. The associated configuration registers must be programmed before the command is issued. [refer to dtm_rx_test function in lec_llh/simulation/llh_driver/ll_hardware_api.c file]
DTM_STOP	0x48	Stop Direct Test Mode.
UPDATE_CHAN_MAP	0x4B	Update channel map for the connection.
UPDATE_CONN_INSTANT	0x4C	Start connection update procedure for the connection.
PACKET_RECEIVED	0x4D	Indicates a received connection packet is read by firmware from connection receive FIFO.
ENTER_DSM	0x50	Enter deep sleep mode.
ENTER_SM	0x51	Enter sleep mode.
EXIT_SM	0x52	Exit sleep mode
ENC_CLK_ON	0x53	Turn on clock to encryption block
ENC_CLK_OFF	0x54	Turn off clock to encryption block
ADV_CLK_ON	0x55	Turn on clock to advertiser block in NAP mode.
ADV_CLK_OFF	0x56	Turn off clock to advertiser block in NAP mode.
SCAN_CLK_ON	0x57	Turn on clock to scanner block in NAP mode.
SCAN_CLK_OFF	0x58	Turn off clock to scanner block in NAP mode.
INIT_CLK_ON	0x59	Turn on clock to initiator block in NAP mode.

Table continues on the next page...

INIT_CLK_OFF	0x5a	Turn off clock to initiator block in NAP mode.
CONN_CLK_ON	0x5b	Turn on clock to connection block in NAP mode.
CONN_CLK_OFF	0x5c	Turn off clock to connection block in NAP mode.
UPDATE_CONN	0x68	Update connection parameters. Deprecated.
KILL_CONN	0x70	Kill connection immediately.
KILL_CONN_AFTER_TX	0x71	Kill connection after a transmit operation is over.
RESET_US_COUNTER	0xc3	Reset microsecond counter
RESP_TIMER_ON	0x72	Start PDU response timer. The PDU_RESP_TIMER register must be programmed with timeout value before issuing this command.
RESP_TIMER_OFF	0x73	Stop PDU response timer.
RESET_READ_PTR	0x74	Reset the white list memory read pointer to 0.
CONN_PING_TIMER_ON	0x75	Start connection ping timer
CONN_PING_TIMER_OFF	0x76	Stop connection ping timer
SOFT_RESET	0x80	Software reset. Resets all the hardware registers (except a few registers related to radio initialization).

Chapter 54
BLE RF Registers

54.1 BLE_RF Memory map/register definition

This section describes the registers and data structures in the BLE RF module as instantiated in the **Radio**. This module is separate from the BLE Link Layer module but shares an address space. The base address of the module depends on the particular memory map of the MCU. The addresses presented here are relative addresses which are relative to the base of the BLE_RF address space.

Table 54-1. Revision History

Date	Revision
3 Mar 2014	Added BLE RF section to the Radio Reference Manual.

BLE_RF_REGS memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
D00	Bluetooth Low Energy Part ID (BLE_RF_REGS_BLE_PART_ID)	16	R/W	0001h	54.1.1/1381
D04	DSM Status (BLE_RF_REGS_DSM_STATUS)	16	R	See section	54.1.2/1382
D08	Bluetooth Low Energy AFC (BLE_RF_REGS_BLE_AFC)	16	R/W	See section	54.1.3/1383
D0C	Bluetooth Low Energy BSM (BLE_RF_REGS_BLE_BSM)	16	R/W	0000h	54.1.4/1384

54.1.1 Bluetooth Low Energy Part ID (BLE_RF_REGS_BLE_PART_ID)

Address: 0h base + D00h offset = D00h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	BLE_PART_ID															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

BLE_RF_REGS_BLE_PART_ID field descriptions

Field	Description
BLE_PART_ID	BLE Part ID

54.1.2 DSM Status (BLE_RF_REGS_DSM_STATUS)

Address: 0h base + D04h offset = D04h

Bit	15	14	13	12	11	10	9	8
Read	0							
Write	▨							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0						RIF_LL_ ACTIVE	ORF_ SYSCLK_ REQ
Write	▨						▨	▨
Reset	0	0	0	0	0	0	x*	x*

* Notes:

- x = Undefined at reset.

BLE_RF_REGS_DSM_STATUS field descriptions

Field	Description
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 RIF_LL_ACTIVE	Link Layer Active Reflects the state of the BLE LL output of the same name, the signal to be used by the host as an 'early' indication to prevent host to do any operations while the BLE block is doing transceiver operations, so as to reduce the peak power and noise.
0 ORF_SYSCLK_ REQ	RF Oscillator Requested Reflects the state of the BLE LL output of the same name, the control signal used to enable/disable the RF Oscillator for entry and exit from DSM (deep sleep mode).

54.1.3 Bluetooth Low Energy AFC (BLE_RF_REGS_BLE_AFC)

Address: 0h base + D08h offset = D08h

Bit	15	14	13	12	11	10	9	8
Read	LATCH_ AFC_ON_ ACCESS_ MATCH	0	BLE_AFC					
Write								
Reset	0	0	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read	BLE_AFC							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

BLE_RF_REGS_BLE_AFC field descriptions

Field	Description
15 LATCH_AFC_ ON_ACCESS_ MATCH	Latch AFC Estimation on Access Address Match 0 BLE_AFC[13:0] is updated whenever preamble is detected 1 BLE_AFC[13:0] is latched at access address match, and will not be updated until the next access address match.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BLE_AFC	BLE AFC Result This field holds the result of the most recent BLE RX AFC (Automatic Frequency Correction) estimation. A new AFC estimation will be generated whenever preamble is detected. If LATCH_AFC_ON_ACCESS_MATCH=1, BLE_AFC will be latched on access address match, and will not change until the next access address match. Otherwise, BLE_AFC will be updated whenever preamble is detected. This is a 14-bit, signed, two's complement value.

54.1.4 Bluetooth Low Energy BSM (BLE_RF_REGS_BLE_BSM)

Address: 0h base + D0Ch offset = D0Ch

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0							BSM_EN_ BLE
Write								
Reset	0	0	0	0	0	0	0	0

BLE_RF_REGS_BLE_BSM field descriptions

Field	Description
15–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 BSM_EN_BLE	BLE Bit Streaming Mode Enable When enabled, the 3 BSM outputs (BSM_DATA, BSM_FRAME, and BSM_BLK) appear on the BSM interface pins of the SoC. In Apache, these are alternate, muxed-GPIO pins, so the appropriate port programming is required. 0 BLE Bit Streaming Mode disabled 1 BLE Bit Streaming Mode enabled

Chapter 55
Zigbee Registers

55.1 Zigbee Memory map/register definition

This section describes the registers and data structures in the Zigbee module as instantiated in the Radio. The addresses presented here are absolute addresses.

ZLL memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_D000	INTERRUPT REQUEST STATUS (ZLL_IRQSTS)	32	R/W	See section	55.1.1/1389
4005_D004	PHY CONTROL (ZLL_PHY_CTRL)	32	R/W	0802_FF00h	55.1.2/1393
4005_D008	EVENT TIMER (ZLL_EVENT_TMR)	32	R	See section	55.1.3/1397
4005_D00C	TIMESTAMP (ZLL_TIMESTAMP)	32	R	See section	55.1.4/1397
4005_D010	T1 COMPARE (ZLL_T1CMP)	32	R/W	00FF_FFFFh	55.1.5/1398
4005_D014	T2 COMPARE (ZLL_T2CMP)	32	R/W	00FF_FFFFh	55.1.6/1398
4005_D018	T2 PRIME COMPARE (ZLL_T2PRIMECMP)	32	R/W	0000_FFFFh	55.1.7/1399
4005_D01C	T3 COMPARE (ZLL_T3CMP)	32	R/W	00FF_FFFFh	55.1.8/1399
4005_D020	T4 COMPARE (ZLL_T4CMP)	32	R/W	00FF_FFFFh	55.1.9/1400
4005_D024	PA POWER (ZLL_PA_PWR)	32	R/W	0000_0008h	55.1.10/ 1400
4005_D028	CHANNEL NUMBER 0 (ZLL_CHANNEL_NUM0)	32	R/W	0000_0012h	55.1.11/ 1400
4005_D02C	LQI AND RSSI (ZLL_LQI_AND_RSSI)	32	R	See section	55.1.12/ 1401
4005_D030	MAC SHORT ADDRESS 0 (ZLL_MACSHORTADDRS0)	32	R/W	FFFF_FFFFh	55.1.13/ 1402
4005_D034	MAC LONG ADDRESS 0 LSB (ZLL_MACLONGADDRS0_LSB)	32	R/W	FFFF_FFFFh	55.1.14/ 1402
4005_D038	MAC LONG ADDRESS 0 MSB (ZLL_MACLONGADDRS0_MSB)	32	R/W	FFFF_FFFFh	55.1.15/ 1403
4005_D03C	RECEIVE FRAME FILTER (ZLL_RX_FRAME_FILTER)	32	R/W	See section	55.1.16/ 1403

Table continues on the next page...

ZLL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_D040	CCA AND LQI CONTROL (ZLL_CCA_LQI_CTRL)	32	R/W	0866_004Bh	55.1.17/ 1405
4005_D044	CCA2 CONTROL (ZLL_CCA2_CTRL)	32	R/W	See section	55.1.18/ 1406
4005_D048	FAD CONTROL (ZLL_FAD_CTRL)	32	R/W	0000_0804h	55.1.19/ 1407
4005_D04C	SNF CONTROL (ZLL_SNF_CTRL)	32	R/W	See section	55.1.20/ 1409
4005_D050	BSM CONTROL (ZLL_BSM_CTRL)	32	R/W	See section	55.1.21/ 1409
4005_D054	MAC SHORT ADDRESS 1 (ZLL_MACSHORTADDRS1)	32	R/W	FFFF_FFFFh	55.1.22/ 1410
4005_D058	MAC LONG ADDRESS 1 LSB (ZLL_MACLONGADDRS1_LSB)	32	R/W	FFFF_FFFFh	55.1.23/ 1410
4005_D05C	MAC LONG ADDRESS 1 MSB (ZLL_MACLONGADDRS1_MSB)	32	R/W	FFFF_FFFFh	55.1.24/ 1411
4005_D060	DUAL PAN CONTROL (ZLL_DUAL_PAN_CTRL)	32	R/W	See section	55.1.25/ 1412
4005_D064	CHANNEL NUMBER 1 (ZLL_CHANNEL_NUM1)	32	R/W	0000_007Fh	55.1.26/ 1416
4005_D068	SAM CONTROL (ZLL_SAM_CTRL)	32	R/W	8080_4000h	55.1.27/ 1416
4005_D06C	SOURCE ADDRESS MANAGEMENT TABLE (ZLL_SAM_TABLE)	32	R/W	See section	55.1.28/ 1418
4005_D070	SAM MATCH (ZLL_SAM_MATCH)	32	R/W	See section	55.1.29/ 1421
4005_D074	SAM FREE INDEX (ZLL_SAM_FREE_IDX)	32	R	See section	55.1.30/ 1423
4005_D078	SEQUENCE CONTROL AND STATUS (ZLL_SEQ_CTRL_STS)	32	R/W	See section	55.1.31/ 1424
4005_D07C	ACK DELAY (ZLL_ACKDELAY)	32	R/W	See section	55.1.32/ 1428
4005_D080	FILTER FAIL CODE (ZLL_FILTERFAIL_CODE)	32	R	See section	55.1.33/ 1429
4005_D084	RECEIVE WATER MARK (ZLL_RX_WTR_MARK)	32	R/W	0000_00FFh	55.1.34/ 1430
4005_D08C	SLOT PRELOAD (ZLL_SLOT_PRELOAD)	32	R/W	0000_0074h	55.1.35/ 1431
4005_D090	ZIGBEE SEQUENCE STATE (ZLL_SEQ_STATE)	32	R	See section	55.1.36/ 1432
4005_D094	TIMER PRESCALER (ZLL_TMR_PRESCALE)	32	R/W	0000_0003h	55.1.37/ 1434
4005_D098	LENIENCY LSB (ZLL_LENIENCY_LSB)	32	R/W	0000_0000h	55.1.38/ 1435

Table continues on the next page...

ZLL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_D09C	LENIENCY MSB (ZLL_LENIENCY_MSB)	32	R/W	0000_0000h	55.1.39/ 1437
4005_D0A0	PART ID (ZLL_PART_ID)	32	R	See section	55.1.40/ 1438
4005_D100	PACKET BUFFER (ZLL_PKT_BUFFER0)	32	R/W	See section	55.1.41/ 1439
4005_D104	PACKET BUFFER (ZLL_PKT_BUFFER1)	32	R/W	See section	55.1.41/ 1439
4005_D108	PACKET BUFFER (ZLL_PKT_BUFFER2)	32	R/W	See section	55.1.41/ 1439
4005_D10C	PACKET BUFFER (ZLL_PKT_BUFFER3)	32	R/W	See section	55.1.41/ 1439
4005_D110	PACKET BUFFER (ZLL_PKT_BUFFER4)	32	R/W	See section	55.1.41/ 1439
4005_D114	PACKET BUFFER (ZLL_PKT_BUFFER5)	32	R/W	See section	55.1.41/ 1439
4005_D118	PACKET BUFFER (ZLL_PKT_BUFFER6)	32	R/W	See section	55.1.41/ 1439
4005_D11C	PACKET BUFFER (ZLL_PKT_BUFFER7)	32	R/W	See section	55.1.41/ 1439
4005_D120	PACKET BUFFER (ZLL_PKT_BUFFER8)	32	R/W	See section	55.1.41/ 1439
4005_D124	PACKET BUFFER (ZLL_PKT_BUFFER9)	32	R/W	See section	55.1.41/ 1439
4005_D128	PACKET BUFFER (ZLL_PKT_BUFFER10)	32	R/W	See section	55.1.41/ 1439
4005_D12C	PACKET BUFFER (ZLL_PKT_BUFFER11)	32	R/W	See section	55.1.41/ 1439
4005_D130	PACKET BUFFER (ZLL_PKT_BUFFER12)	32	R/W	See section	55.1.41/ 1439
4005_D134	PACKET BUFFER (ZLL_PKT_BUFFER13)	32	R/W	See section	55.1.41/ 1439
4005_D138	PACKET BUFFER (ZLL_PKT_BUFFER14)	32	R/W	See section	55.1.41/ 1439
4005_D13C	PACKET BUFFER (ZLL_PKT_BUFFER15)	32	R/W	See section	55.1.41/ 1439
4005_D140	PACKET BUFFER (ZLL_PKT_BUFFER16)	32	R/W	See section	55.1.41/ 1439
4005_D144	PACKET BUFFER (ZLL_PKT_BUFFER17)	32	R/W	See section	55.1.41/ 1439
4005_D148	PACKET BUFFER (ZLL_PKT_BUFFER18)	32	R/W	See section	55.1.41/ 1439
4005_D14C	PACKET BUFFER (ZLL_PKT_BUFFER19)	32	R/W	See section	55.1.41/ 1439

Table continues on the next page...

ZLL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_D150	PACKET BUFFER (ZLL_PKT_BUFFER20)	32	R/W	See section	55.1.41/ 1439
4005_D154	PACKET BUFFER (ZLL_PKT_BUFFER21)	32	R/W	See section	55.1.41/ 1439
4005_D158	PACKET BUFFER (ZLL_PKT_BUFFER22)	32	R/W	See section	55.1.41/ 1439
4005_D15C	PACKET BUFFER (ZLL_PKT_BUFFER23)	32	R/W	See section	55.1.41/ 1439
4005_D160	PACKET BUFFER (ZLL_PKT_BUFFER24)	32	R/W	See section	55.1.41/ 1439
4005_D164	PACKET BUFFER (ZLL_PKT_BUFFER25)	32	R/W	See section	55.1.41/ 1439
4005_D168	PACKET BUFFER (ZLL_PKT_BUFFER26)	32	R/W	See section	55.1.41/ 1439
4005_D16C	PACKET BUFFER (ZLL_PKT_BUFFER27)	32	R/W	See section	55.1.41/ 1439
4005_D170	PACKET BUFFER (ZLL_PKT_BUFFER28)	32	R/W	See section	55.1.41/ 1439
4005_D174	PACKET BUFFER (ZLL_PKT_BUFFER29)	32	R/W	See section	55.1.41/ 1439
4005_D178	PACKET BUFFER (ZLL_PKT_BUFFER30)	32	R/W	See section	55.1.41/ 1439
4005_D17C	PACKET BUFFER (ZLL_PKT_BUFFER31)	32	R/W	See section	55.1.41/ 1439

55.1.1 INTERRUPT REQUEST STATUS (ZLL_IRQSTS)

Zigbee Interrupt Request Status

Address: 4005_D000h base + 0h offset = 4005_D000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	RX_FRAME_LENGTH							TMR4MSK	TMR3MSK	TMR2MSK	TMR1MSK	TMR4IRQ	TMR3IRQ	TMR2IRQ	TMR1IRQ
W													w1c	w1c	w1c	w1c
Reset	0	x*	x*	x*	x*	x*	x*	x*	1	1	1	1	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRCVALID	CCA	SRCADDR	PI	TMRSTATUS	0	PB_ERR_IRQ	0	RX_FRM_PEND	PLL_UNLOCK_IRQ	FILTERFAIL_IRQ	RXWTRMRKIRQ	CCAIRQ	RXIRQ	TXIRQ	SEQIRQ
W							w1c			w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	x*	x*	x*	x*	x*	0	x*	0	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_IRQSTS field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 RX_FRAME_ LENGTH	Receive Frame Length Contents of the PHR (PHY header), or FrameLength field, of the most recently received packet. Read-only.
23 TMR4MSK	Timer Comperator 4 Interrupt Mask bit 0 allows interrupt when comparator matches event timer count 1 Interrupt generation is disabled, but a TMR4IRQ flag can be set
22 TMR3MSK	Timer Comperator 3 Interrupt Mask bit 0 allows interrupt when comparator matches event timer count 1 Interrupt generation is disabled, but a TMR3IRQ flag can be set
21 TMR2MSK	Timer Comperator 2 Interrupt Mask bit 0 allows interrupt when comparator matches event timer count 1 Interrupt generation is disabled, but a TMR2IRQ flag can be set
20 TMR1MSK	Timer Comperator 1 Interrupt Mask bit 0 allows interrupt when comparator matches event timer count 1 Interrupt generation is disabled, but a TMR1IRQ flag can be set
19 TMR4IRQ	Timer 4 IRQ Timer Comparator 4 Interrupt Status bit: Indiates T4CMP comparator value matched event timer counter. This is write '1' to clear bit
18 TMR3IRQ	Timer 3 IRQ Timer Comparator 3 Interrupt Status bit: Indiates T3CMP comparator value matched event timer counter. This is write '1' to clear bit
17 TMR2IRQ	Timer 2 IRQ Timer Comparator 2 Interrupt Status bit: Indiates comparator value matched event timer counter. This flag is shared between the T2CMP (24-bit) and T2PRIMECMP (16-bit) compare registers. This is write '1' to clear bit
16 TMR1IRQ	Timer 1 IRQ Timer Comparator 1 Interrupt Status bit: Indiates T1CMP comparator value matched event timer counter. This is write '1' to clear bit
15 CRCVALID	CRC Valid Status Code Redundancy Check Valid: This flag indicates the compare result between the FCS field, in the most-recently received frame, and the internally calculated CRC value. This flag is cleared at next receiver warm up. 0 Rx FCS != calculated CRC (incorrect) 1 Rx FCS = calculated CRC (correct)
14 CCA	CCA Status Channel IDLE/BUSY indicator. This indicator is valid at CCAIRQ and also at SEQIRQ. This flag is cleared at next receiver warm up.

Table continues on the next page...

ZLL_IRQSTS field descriptions (continued)

Field	Description
	0 IDLE 1 BUSY
13 SRCADDR	Source Address Match Status If Source Address Management is engaged, meaning at least one of the following bits is set: SAP0_EN SAA0_EN SAP1_EN SAA1_EN Then SRCADDR will be set to 1 if the packet just received is a poll request (PI=1), *and* at least one of the following conditions is met: SAP0_EN and SAP0_ADDR_PRESENT SAA0_EN and SAA0_ADDR_ABSENT SAP1_EN and SAP1_ADDR_PRESENT SAA1_EN and SAA1_ADDR_ABSENT If SRCADDR=1, this indicates to SW that the Packet Processor has determined that an auto-TxACK frame must be transmitted with the FramePending subfield of the FrameControlField set to 1. HW will assemble and transmit this Ack packet. If the above conditions are not met, SRCADDR will be cleared to 0.
12 PI	Poll Indication 0 the received packet was not a data request 1 the received packet was a data request, regardless of whether a Source Address table match occurred, or whether Source Address Management is enabled or not
11 TMRSTATUS	Composite TMR Status 0 no TMRxIRQ is asserted 1 At least one of the TMRxIRQ is asserted (TMR1IRQ, TMR2IRQ, TMR3IRQ, or TMR4IRQ)
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 PB_ERR_IRQ	Packet Buffer Underrun Error IRQ 0 A Packet Buffer Underrun Error Interrupt has not occurred 1 A Packet Buffer Underrun Error Interrupt has occurred
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 RX_FRM_PEND	Status of the frame pending bit of the frame control field for the most-recently received packet. Read-only.
6 PLL_UNLOCK_IRQ	PLL Un-lock Interrupt Status bit. A '1' indicates an unlock event has occurred in the PLL. This is write a '1' to clear bit. 0 A PLL Unlock Interrupt has not occurred 1 A PLL Unlock Interrupt has occurred
5 FILTERFAIL_IRQ	Receiver Packet Filter Fail Interrupt Status bit. A '1' indicates that the most-recently received packet has been rejected due to elements within the packet. This is write a '1' to clear bit. In Dual PAN mode, FILTERFAIL_IRQ applies to either or both networks, as follows: **A:** If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=0, FILTERFAIL_IRQ applies to PAN0.

Table continues on the next page...

ZLL_IRQSTS field descriptions (continued)

Field	Description
	B: If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=1, FILTERFAIL_IRQ applies to PAN1. **C:** If PAN0 and PAN1 occupy the same channel, FILTERFAIL_IRQ is the logical 'AND' of the individual PANs' Filter Fail status. 0 A Filter Fail Interrupt has not occurred 1 A Filter Fail Interrupt has occurred
4 RXWTRMRKIRQ	Receiver Byte Count Water Mark Interrupt Status bit. A '1' indicates that the number of bytes specified in the RX_WTR_MARK register has been reached. This is write a '1' to clear bit. 0 A RX Watermark Interrupt has not occurred 1 A RX Watermark Interrupt has occurred
3 CCAIRQ	Clear Channel Assessment Interrupt Status bit. A '1' indicates completion of CCA operation. This is write '1' to clear bit. 0 A CCA Interrupt has not occurred 1 A CCA Interrupt has occurred
2 RXIRQ	Receiver Interrupt Status bit. A '1' indicates the completion of a receive operation. This is write a '1' to clear bit. 0 A RX Interrupt has not occurred 1 A RX Interrupt has occurred
1 TXIRQ	Transmitter Interrupt Status bit. A '1' indicates the completion of a transmit operation. This is write a '1' to clear bit. 0 A TX Interrupt has not occurred 1 A TX Interrupt has occurred
0 SEQIRQ	Sequence-end Interrupt Status bit. A '1' indicates the completion of an autosequence. This interrupt will assert whenever the Sequence Manager transitions from non-idle to idle state, for any reason. This is write a '1' to clear bit. 0 A Sequencer Interrupt has not occurred 1 A Sequencer Interrupt has occurred

55.1.2 PHY CONTROL (ZLL_PHY_CTRL)

PHY Control Register

Address: 4005_D000h base + 4h offset = 4005_D004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRCV_MSK	TC3TMOUT	PANCORDNTR0	CCATYPE			PROMISCUOUS	TC2PRIME_EN	TMR4CMP_EN	TMR3CMP_EN	TMR2CMP_EN	TMR1CMP_EN	0		PB_ERR_MSK	0
W						TMRLOAD										
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRC_MSK	PLL_UNLOCK_MSK	FILTERFAIL_MSK	RX_WMRK_MSK	CCAMSK	RXMSK	TXMSK	SEQMSK	TMRTRIGEN	SLOTTED	CCABFRTX	RXACKRQD	AUTOACK	XCVSEQ		
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

ZLL_PHY_CTRL field descriptions

Field	Description
31 TRCV_MSK	Transceiver Global Interrupt Mask Transceiver Global Interrupt Mask 0 Enable any unmasked interrupt source to assert zigbee interrupt 1 Mask all interrupt sources from asserting zigbee interrupt
30 TC3TMOUT	TMR3 Timeout Enable TMR3 Timeout Enable 0 TMR3 is a software timer only 1 Enable TMR3 to abort Rx or CCCA operations.

Table continues on the next page...

ZLL_PHY_CTRL field descriptions (continued)

Field	Description
29 PANCORDNTR0	Device is a PAN Coordinator on PAN0 Device is a PAN Coordinator on PAN0. Allows device to receive packets with no destination address, if Source PAN ID matches.
28–27 CCATYPE	Clear Channel Assessment Type Clear Channel Assessment Type. Selects one of four possible functions for CCA or Energy Detect, per below. 0 ENERGY DETECT 1 CCA MODE 1 2 CCA MODE 2 3 CCA MODE 3
26 TMRLOAD	Event Timer Load Enable A low to high transition of this bit causes the contents of register T1CMP[23:0] to be loaded into the Event Timer.This is a self clearing bit, always reads zero. **Note:** the TMRLOAD feature requires the Apache RF Oscillator to be running; TMRLOAD should not be attempted in the Apache radio gasket-bypass mode.
25 PROMISCUOUS	Promiscuous Mode Enable Bypasses most packet filtering. 0 normal mode 1 all packet filtering except frame length checking (FrameLength>=5 and FrameLength<=127) is bypassed.
24 TC2PRIME_EN	Timer 2 Prime Compare Enable 0 Don't allow a match of the lower 16 bits of Event Timer to T2PRIMECMP to set TMR2IRQ 1 Allow a match of the lower 16 bits of Event Timer to T2PRIMECMP to set TMR2IRQ
23 TMR4CMP_EN	Timer 4 Compare Enable 0 Don't allow an Event Timer Match to T4CMP to set TMR4IRQ 1 Allow an Event Timer Match to T4CMP to set TMR4IRQ
22 TMR3CMP_EN	Timer 3 Compare Enable 0 Don't allow an Event Timer Match to T3CMP to set TMR3IRQ 1 Allow an Event Timer Match to T3CMP to set TMR3IRQ
21 TMR2CMP_EN	Timer 2 Compare Enable 0 Don't allow an Event Timer Match to T2CMP or T2PRIMECMP to set TMR2IRQ 1 Allow an Event Timer Match to T2CMP or T2PRIMECMP to set TMR2IRQ
20 TMR1CMP_EN	Timer 1 Compare Enable 0 Don't allow an Event Timer Match to T1CMP to set TMR1IRQ 1 Allow an Event Timer Match to T1CMP to set TMR1IRQ
19–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 PB_ERR_MSK	Packet Buffer Error Interrupt Mask

Table continues on the next page...

ZLL_PHY_CTRL field descriptions (continued)

Field	Description
	0 Enable Packet Buffer Error to assert a zigbee interrupt 1 Mask Packet Buffer Error from generating a zigbee interrupt
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CRC_MSK	CRC Mask CRC Mask 0 sequence manager ignores CRCVALID and considers the receive operation complete after the last octet of the frame has been received. 1 sequence manager requires CRCVALID=1 at the end of the received frame in order for the receive operation to complete successfully; if CRCVALID=0, sequence manager will return to preamble-detect mode after the last octet of the frame has been received.
14 PLL_UNLOCK_ MSK	PLL Unlock Interrupt Mask 0 allows PLL unlock event to generate a zigbee interrupt 1 A PLL unlock event will set the PLL_UNLOCK_IRQ status bit, but a zigbee interrupt is not generated
13 FILTERFAIL_ MSK	FilterFail Interrupt Mask FilterFail Interrupt Mask 0 allows Packet Processor Filtering Failure to generate a zigbee interrupt 1 A Packet Processor Filtering Failure will set the FILTERFAIL_IRQ status bit, but a zigbee interrupt is not generated
12 RX_WMRK_MSK	RX Watermark Interrupt Mask RX Watermark Interrupt Mask 0 allows a Received Byte Count match to the RX_WTR_MARK threshold register to generate a zigbee interrupt 1 A Received Byte Count match to the RX_WTR_MARK threshold register will set the RXWTRMRKIRQ status bit, but a zigbee interrupt is not generated
11 CCAMSK	CCA Interrupt Mask CCA Interrupt Mask 0 allows completion of a CCA operation to generate a zigbee interrupt 1 Completion of a CCA operation will set the CCAIRQ status bit, but an zigbee interrupt
10 RXMSK	RX Interrupt Mask 0 allows completion of a RX operation to generate a zigbee interrupt 1 Completion of a RX operation will set the RXIRQ status bit, but a zigbee interrupt is not generated
9 TXMSK	TX Interrupt Mask 0 allows completion of a TX operation to generate a zigbee interrupt 1 Completion of a TX operation will set the TXIRQ status bit, but a zigbee interrupt is not generated
8 SEQMSK	Sequencer Interrupt Mask 0 allows completion of an autosequence to generate a zigbee interrupt 1 Completion of an autosequence will set the SEQIRQ status bit, but a zigbee interrupt is not generated

Table continues on the next page...

ZLL_PHY_CTRL field descriptions (continued)

Field	Description
7 TMRTRIGEN	Timer2 Trigger Enable 0 programmed sequence initiates immediately upon write to XCVSEQ. 1 allow timer TC2 (or TC2') to initiate a preprogrammed sequence (see XCVSEQ register).
6 SLOTTED	Slotted Mode Slotted Mode, for beacon-enabled networks. Applies only to Sequences T, TR, and R, ignored during all other sequences. Used, in concert with CCABFRTX, to determine how many CCA measurements are required prior to a transmit operation. Also used during R sequence to determine whether the ensuing transmit acknowledge frame (if any) needs to be synchronized to a backoff slot boundary.
5 CCABFRTX	CCA Before TX Applies only to Sequences T and TR, ignored during all other sequences. 0 no CCA required, transmit operation begins immediately. 1 at least one CCA measurement is required prior to the transmit operation (see also SLOTTED).
4 RXACKRQD	Receive Acknowledge Frame required Applies only to Sequence TR, ignored during all other sequences. 0 An ordinary receive frame (any type of frame) follows the transmit frame. 1 A receive Ack frame is expected to follow the transmit frame (non-Ack frames are rejected).
3 AUTOACK	Auto Acknowledge Enable Applies only to Sequence R and Sequence TR, ignored during other sequences 0 sequence manager will not follow a receive frame with a Tx Ack frame, under any conditions; the autosequence will terminate after the receive frame. 1 sequence manager will follow a receive frame with an automatic hardware-generated Tx Ack frame, assuming other necessary conditions are met.
XCVSEQ	Zigbee Transceiver Sequence Selector The Transceiver Sequence Selector register selects an autosequence for the sequence manager to execute. Sequence initiation can be immediate, or scheduled (see TMRTRIGEN). A write of XCVSEQ=IDLE will abort any ongoing sequence. A write of XCVSEQ=IDLE must always be performed after a sequence is complete, and before a new sequence is programmed. Any write to XCVSEQ other than XCVSEQ=IDLE during an ongoing sequence, shall be ignored. The mapping of XCVSEQ to sequence types is as follows: 0 I (IDLE) 1 R (RECEIVE) 2 T (TRANSMIT) 3 C (CCA) 4 TR (TRANSMIT/RECEIVE) 5 CCCA (CONTINUOUS CCA) 6 Reserved 7 Reserved

55.1.3 EVENT TIMER (ZLL_EVENT_TMR)

Holds the current value of the 24-bit event timer. The hardware latches the upper 2 bytes of EVENT_TMR on each read of least signficant byte.

Address: 4005_D000h base + 8h offset = 4005_D008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								EVENT_TMR																							
W																																
Reset	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_EVENT_TMR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
EVENT_TMR	Event Timer

55.1.4 TIMESTAMP (ZLL_TIMESTAMP)

Holds the latched value of the Event Timer current time corresponding to the beginning of the just received Rx packet, at SFD detect.

Address: 4005_D000h base + Ch offset = 4005_D00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TIMESTAMP																							
W																																
Reset	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_TIMESTAMP field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TIMESTAMP	Timestamp

55.1.5 T1 COMPARE (ZLL_T1CMP)

TMR1 compare value. If TMR1CMP_EN=1 and the Event Timer matches this value, TMR1IRQ is set.

Address: 4005_D000h base + 10h offset = 4005_D010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T1CMP																							
W	▨																															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_T1CMP field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
T1CMP	TMR1 Compare Value

55.1.6 T2 COMPARE (ZLL_T2CMP)

TMR2 compare value. If TMR2CMP_EN=1 and TC2PRIME_EN=0 and the Event Timer matches this value, TMR2IRQ is set.

Address: 4005_D000h base + 14h offset = 4005_D014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T2CMP																							
W	▨																															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_T2CMP field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
T2CMP	TMR2 Compare Value

55.1.7 T2 PRIME COMPARE (ZLL_T2PRIMECMP)

TMR2 compare value. If TMR2CMP_EN=1 and TC2PRIME_EN=1 and the Event Timer matches this value, TMR2IRQ is set.

Address: 4005_D000h base + 18h offset = 4005_D018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																T2PRIMECMP															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_T2PRIMECMP field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
T2PRIMECMP	TMR2 Prime Compare Value

55.1.8 T3 COMPARE (ZLL_T3CMP)

TMR3 compare value. If TMR3CMP_EN=1 and the Event Timer matches this value, TMR3IRQ is set.

Address: 4005_D000h base + 1Ch offset = 4005_D01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T3CMP																							
W																																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_T3CMP field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
T3CMP	TMR3 Compare Value

55.1.9 T4 COMPARE (ZLL_T4CMP)

TMR4 compare value. If TMR4CMP_EN=1 and the Event Timer matches this value, TMR4IRQ is set.

Address: 4005_D000h base + 20h offset = 4005_D020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T4CMP																							
W																																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_T4CMP field descriptions

Field	Description
31–24 Reserved	TMR4 Compare Value This field is reserved. This read-only field is reserved and always has the value 0.
T4CMP	TMR4 Compare Value

55.1.10 PA POWER (ZLL_PA_PWR)

PA Target Power used to transmit Zigbee packets

Address: 4005_D000h base + 24h offset = 4005_D024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												PA_PWR			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ZLL_PA_PWR field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PA_PWR	PA Power

55.1.11 CHANNEL NUMBER 0 (ZLL_CHANNEL_NUM0)

This is the mapped channel number used to transmit and receive Zigbee packets. If Dual PAN is engaged, this register applies to PAN0. CHANNEL_NUM0 should be in the range:

11 <= CHANNEL_NUM0 <= 26

Address: 4005_D000h base + 28h offset = 4005_D028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CHANNEL_NUM0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

ZLL_CHANNEL_NUM0 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CHANNEL_ NUM0	Channel Number for PAN0

55.1.12 LQI AND RSSI (ZLL_LQI_AND_RSSI)

Address: 4005_D000h base + 2Ch offset = 4005_D02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CCA1_ED_FNL								RSSI								LQI_VALUE							
W																																
Reset	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_LQI_AND_RSSI field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 CCA1_ED_FNL	RSSI Value Output register to show final averaged RSSI value or compensated value of the same at the end of a CCA Mode1 or Energy Detect computation.
15–8 RSSI	RSSI Value RSSI Output
LQI_VALUE	LQI Value Link Quality Indicator for the most recently received packet. (LQI is also available in the Packet Buffer, at the end of the received packet data)

55.1.13 MAC SHORT ADDRESS 0 (ZLL_MACSHORTADDRS0)

Address: 4005_D000h base + 30h offset = 4005_D030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACSHORTADDRS0																MACPANID0															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACSHORTADDRS0 field descriptions

Field	Description
31–16 MACSHORTADDRS0	MAC SHORT ADDRESS for PAN0 MAC Short Address for PAN0, for 16-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.
MACPANID0	MAC PAN ID for PAN0 MAC PAN ID for PAN0. The packet processor compares the incoming packet's Destination PAN ID against the contents of this register to determine if the packet is addressed to this device; or if the incoming packet is a Beacon frame, the packet processor compares the incoming packet Source PAN ID against this register. Also, if PANCORDNTR0=1, and the incoming packet has no Destination Address field, and if the incoming packet is a Data or MAC Command frame, the packet processor compares the incoming packet Source PAN ID against this register.

55.1.14 MAC LONG ADDRESS 0 LSB (ZLL_MACLONGADDRS0_LSB)

MAC Long Address for PAN0, for 64-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.

Address: 4005_D000h base + 34h offset = 4005_D034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACLONGADDRS0_LSB																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACLONGADDRS0_LSB field descriptions

Field	Description
MACLONGADDRS0_ LSB	MAC LONG ADDRESS for PAN0 LSB

55.1.15 MAC LONG ADDRESS 0 MSB (ZLL_MACLONGADDRS0_MSB)

MAC Long Address for PAN0, for 64-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.

Address: 4005_D000h base + 38h offset = 4005_D038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACLONGADDRS0_MSB																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACLONGADDRS0_MSB field descriptions

Field	Description
MACLONGADDRS0_ MSB	MAC LONG ADDRESS for PAN0 MSB

55.1.16 RECEIVE FRAME FILTER (ZLL_RX_FRAME_FILTER)

Address: 4005_D000h base + 3Ch offset = 4005_D03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FRM_VER		ACTIVE_ PROMISCUOUS	NS_FT	CMD_FT	ACK_FT	DATA_FT	BEACON_FT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

ZLL_RX_FRAME_FILTER field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 FRM_VER	Frame Version Selector Frame Version selector. The incoming packet's Frame Control Field is parsed to obtain the FrameVersion subfield, and that value is compared against this register, in accordance with the following: 00: Any FrameVersion accepted (0,1,2 or 3) 01: Only accept FrameVersion 0 packets (2003 compliant) 10: Only accept FrameVersion 1 packets (2006 compliant) 11: Accept FrameVersion 0 and 1 packets, reject all others Frames received with FrameVersion 2 or 3 will be treated identically to FrameVersion 1, with respect to parsing of the Auxiliary Security Header. Other than this Header, all 4 frame versions will be treated identically
5 ACTIVE_ PROMISCUOUS	Active Promiscuous 0 normal operation 1 Provide Data Indication on all received packets under the same rules which apply in PROMISCUOUS mode, however acknowledge those packets under rules which apply in non-PROMISCUOUS mode
4 NS_FT	Not Specified Frame Type Enable 0 reject all reserved frame types 1 Not-specified (reserved) frame type enabled. No packet filtering is performed, except for frame length checking (FrameLength>=5 and FrameLength<=127).
3 CMD_FT	MAC Command Frame Type Enable 0 reject all MAC Command frames 1 MAC Command frame type enabled.
2 ACK_FT	Ack Frame Type Enable 0 reject all Acknowledge frames 1 Acknowledge frame type enabled.
1 DATA_FT	Data Frame Type Enable 0 reject all Data frames 1 Data frame type enabled.
0 BEACON_FT	Beacon Frame Type Enable 0 reject all Beacon frames 1 Beacon frame type enabled.

55.1.17 CCA AND LQI CONTROL (ZLL_CCA_LQI_CTRL)

Address: 4005_D000h base + 40h offset = 4005_D040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				CCA3_AND_NOT_OR	0			LQI_OFFSET_COMP							
W																
Reset	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CCA1_THRESH							
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1

ZLL_CCA_LQI_CTRL field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 CCA3_AND_NOT_OR	CCA Mode 3 AND not OR Determines the way CCA3 is required to be detected 0 CCA1 or CCA2 1 CCA1 and CCA2
26–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 LQI_OFFSET_COMP	LQI Offset Compensation Programmable amount to offset RSSI based LQI value
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CCA1_THRESH	CCA Mode 1 Threshold Programmable energy threshold register for CCA mode 1.

55.1.18 CCA2 CONTROL (ZLL_CCA2_CTRL)

CCA Mode 2 Control Register

Address: 4005_D000h base + 44h offset = 4005_D044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CCA2_CORR_THRESH								0	CCA2_MIN_NUM_CORR_TH			CCA2_NUM_CORR_PEAKS			
W																
Reset	1	0	0	0	0	0	1	0	0	0	1	1	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_CCA2_CTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 CCA2_CORR_THRESH	CCA Mode 2 Correlation Threshold Programmable threshold to be compared against number of correlation peaks that exceeded cca2_corr_thresh for detecting CCA mode 2. Number of peaks detected = cca2_min_num_corr_th + 1; Example: If it is programmed to 3, CCA2 logic looks for at least 4 correlation peaks that crossed the threshold, to indicate channel is idle or busy.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 CCA2_MIN_NUM_CORR_TH	CCA Mode 2 Threshold Number of Correlation Peaks Programmable threshold to be compared against number of correlation peaks that exceeded cca2_corr_thresh for detecting CCA mode 2. Number of peaks detected = cca2_min_num_corr_th + 1; Example: If it is programmed to 3, CCA2 logic looks for at least 4 correlation peaks that crossed the threshold, to indicate channel is idle or busy.
CCA2_NUM_CORR_PEAKS	CCA Mode 2 Number of Correlation Peaks Detected Counts of number of peaks that crossed cca2_corr_thresh in CCA Mode 2 operation

55.1.19 FAD CONTROL (ZLL_FAD_CTRL)

Fast Antenna Diversity Control Register

Address: 4005_D000h base + 48h offset = 4005_D048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ANTX_POL				ANTX_ CTRLMODE	ANTX_HZ	ANTX_EN		0					FAD_NOT_GPIO	ANTX	FAD_EN
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0

ZLL_FAD_CTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15–12
ANTX_POL
	Antenna Diversity PAD Polarity
	
	Control the polarity of the Antenna or Switch control pins:
	ANTX_POL[0]=0 : don't invert the ANT_A output
	ANTX_POL[1]=0 : don't invert the ANT_B output
	ANTX_POL[2]=0 : don't invert the TX_SWITCH output
	ANTX_POL[3]=0 : don't invert the RX_SWITCH output
	
	ANTX_POL[0]=1 : invert the ANT_A output
	ANTX_POL[1]=1 : invert the ANT_B output
	ANTX_POL[2]=1 : invert the TX_SWITCH output
	ANTX_POL[3]=1 : invert the RX_SWITCH output

11
ANTX_
CTRLMODE
	Antenna Diversity Control Mode
	
	When ANTX_CTRLMODE=1 (dual mode):
	ANT_A=NOT(ANTX) AND (GPIO3_TRIG_EN OR GPIO2_TRIG_EN)
	ANT_B=ANTX AND (GPIO3_TRIG_EN OR GPIO2_TRIG_EN)
	TX_SWITCH=GPIO2_TRIG_EN
	RX_SWITCH=GPIO3_TRIG_EN
	
	When ANTX_CTRLMODE=0 (single mode):
	ANT_A=NOT(ANTX) AND (GPIO3_TRIG_EN OR GPIO2_TRIG_EN)
	ANT_B=ANTX AND (GPIO3_TRIG_EN OR GPIO2_TRIG_EN)
	TX_SWITCH=GPIO2_TRIG_EN

Table continues on the next page...

ZLL_FAD_CTRL field descriptions (continued)

Field	Description
	RX_SWITCH=(GPIO3_TRIG_EN OR GPIO2_TRIG_EN) GPIO2_TRIG_EN and GPIO3_TRIG_EN are outputs of the Transceiver Sequence Manager (TSM). The TSM timing registers associated with GPIO2_TRIG_EN and GPIO3_TRIG_EN should be programmed with the desired TX_SWITCH and RX_SWITCH timing, before enabling Fast Antenna Diversity.
10 ANTX_HZ	FAD PAD Tristate Control 0 ANT_A, ANT_B, RX_SWITCH and TX_SWITCH are actively driven outputs. 1 Antenna controls high impedance- Set ANT_A, ANT_B, RX_SWITCH and TX_SWITCH in high impedance.
9–8 ANTX_EN	FAD Antenna Controls Enable When FAD_NOT_GPIO=1, ANTX_EN[1:0] determines which pairs of FAD related outputs are enabled: 00 all disabled (held low) 01 only RX/TX_SWITCH enabled 10 only ANT_A/B enabled 11 all enabled
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 FAD_NOT_GPIO	FAD/GPIO Selector This bit FAD_NOT_GPIO determines whether the 4 FAD-related pads function in FAD mode or in TSM/GPIO mode. The 4 FAD-related pads are: ANT_A, ANT_B, TX_SWITCH, and RX_SWITCH. If FAD_NOT_GPIO=1, these pads assume their Zigbee FAD functionality If FAD_NOT_GPIO=0, these pads are assigned as TSM GPIO outputs, as shown: ANT_A -> GPIO0_TRIG_EN ANT_B -> GPIO1_TRIG_EN TX_SWITCH -> GPIO2_TRIG_EN RX_SWITCH -> GPIO3_TRIG_EN To use these pads in TSM GPIO mode, the TSM timing registers associated with GPIO0-3 should be programmed with the desired timings.
1 ANTX	Antenna Selection If FAD_EN=0, the ANTX bit is used to take manual (software) control of the antenna selection, overriding the FAD state machine; in this case, the readback value of ANTX is whatever was last written by the host. If FAD_EN=1, the FAD state machine controls antenna selection, and the readback value of ANTX reflects the machine-selected antenna.
0 FAD_EN	FAD Enable Enable Fast Antenna Diversity for Zigbee

55.1.20 SNF CONTROL (ZLL_SNF_CTRL)

SNIFF Mode Control Register

Address: 4005_D000h base + 4Ch offset = 4005_D04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															SNF_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_SNF_CTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SNF_EN	SNF Enable SNIFF Mode Enable. **Note:** SNIFF Mode not currently supported. SNF_EN has no effect

55.1.21 BSM CONTROL (ZLL_BSM_CTRL)

Bit Streaming Mode Control Register

Address: 4005_D000h base + 50h offset = 4005_D050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															BSM_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_BSM_CTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 BSM_EN	BSM Enable 0 Zigbee Bit Streaming Mode Disabled 1 Zigbee Bit Streaming Mode Enabled

55.1.22 MAC SHORT ADDRESS 1 (ZLL_MACSHORTADDRS1)

Address: 4005_D000h base + 54h offset = 4005_D054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACSHORTADDRS1																MACPANID1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACSHORTADDRS1 field descriptions

Field	Description
31–16 MACSHORTADDRS1	MAC SHORT ADDRESS for PAN1 MAC Short Address for PAN1, for 16-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.
MACPANID1	MAC PAN ID for PAN1 MAC PAN ID for PAN1. The packet processor compares the incoming packet's Destination PAN ID against the contents of this register to determine if the packet is addressed to this device; or if the incoming packet is a Beacon frame, the packet processor compares the incoming packet Source PAN ID against this register. Also, if PANCORDNTR1=1, and the incoming packet has no Destination Address field, and if the incoming packet is a Data or MAC Command frame, the packet processor compares the incoming packet Source PAN ID against this register.

55.1.23 MAC LONG ADDRESS 1 LSB (ZLL_MACLONGADDRS1_LSB)

MAC Long Address for PAN1, for 64-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.

Address: 4005_D000h base + 58h offset = 4005_D058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACLONGADDRS1_LSB																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACLONGADDRS1_LSB field descriptions

Field	Description
MACLONGADDRS1_ LSB	MAC LONG ADDRESS for PAN1 LSB

55.1.24 MAC LONG ADDRESS 1 MSB (ZLL_MACLONGADDRS1_MSB)

MAC Long Address for PAN1, for 64-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.

Address: 4005_D000h base + 5Ch offset = 4005_D05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MACLONGADDRS1_MSB																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ZLL_MACLONGADDRS1_MSB field descriptions

Field	Description
MACLONGADDRS1_ MSB	MAC LONG ADDRESS for PAN1 MSB MAC LONG ADDRESS for PAN1 MSB

55.1.25 DUAL PAN CONTROL (ZLL_DUAL_PAN_CTRL)

Address: 4005_D000h base + 60h offset = 4005_D060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RECD_ON_PAN1	RECD_ON_PAN0	DUAL_PAN_REMAIN					
W																
Reset	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DUAL_PAN_DWELL								0		ZB_DP_CHAN_OVRD_SEL	ZB_DP_CHAN_OVRD_EN	CURRENT_NETWORK	PANCORDNTR1	DUAL_PAN_AUTO	ACTIVE_NETWORK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	x*	0	0	0

* Notes:

- x = Undefined at reset.

ZLL_DUAL_PAN_CTRL field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 RECD_ON_PAN1	Last Packet was Received on PAN1 Indicates the packet which was just received, was received on PAN1. In Dual PAN mode operating on 2 different channels, RECD_ON_PAN1 will be set if CURRENT_NETWORK=1 when the packet was received, regardless of FILTERFAIL status. In DUAL PAN mode operating with same channel on both networks, CURRENT_NETWORK will be ignored and RECD_ON_PAN1 will be set only if a valid packet was received on PAN1 (PAN1's FILTERFAIL_FLAG is deasserted). RECD_ON_PAN1 remains valid until the start of the next autoseqeuence.
22 RECD_ON_PAN0	Last Packet was Received on PAN0 Indicates the packet which was just received, was received on PAN0. In Dual PAN mode operating on 2 different channels, RECD_ON_PAN0 will be set if CURRENT_NETWORK=0 when the packet was received, regardless of FILTERFAIL status. In DUAL PAN mode operating with same channel on both networks, CURRENT_NETWORK will be ignored and RECD_ON_PAN0 will be set only if a valid packet was received on PAN0 (PAN0's FILTERFAIL_FLAG is deasserted). RECD_ON_PAN0 remains valid until the start of the next autoseqeuence.
21–16 DUAL_PAN_ REMAIN	Time Remaining before next PAN switch in auto Dual PAN mode This read-only register indicates time remaining before next PAN switch in auto Dual PAN mode. The units for this register, depend on the PRESCALER setting (bits [1:0]) in the DUAL_PAN_DWELL register, according to the following table:

Table continues on the next page...

ZLL_DUAL_PAN_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DUAL_PAN_DWELL PRESCALER</th><th>DUAL_PAN_REMAIN UNITS</th></tr>
<tr><td>00</td><td>0.5ms</td></tr>
<tr><td>01</td><td>2.5ms</td></tr>
<tr><td>10</td><td>10ms</td></tr>
<tr><td>11</td><td>50ms</td></tr>
</table>
The readback value indicates that between N-1 and N timebase units remain until the next PAN switch. For example, a DUAL_PAN_REMAIN readback value of 3, with a DUAL_PAN_DWELL PRESCALER setting of 2 (10ms), indicates that between 20ms (2*10ms) and 30ms (3*10ms), remain until the next automatic PAN switch.
</td></tr>
<tr><td>15–8
DUAL_PAN_
DWELL</td><td>Dual PAN Channel Frequency Dwell Time

Channel Frequency Dwell Time. In Auto Dual PAN mode, hardware will toggle the PAN, after dwelling on the current PAN for the interval described below (assuming Preamble/SFD not detected). A write to DUAL_PAN_DWELL, always re-initilizes the DWELL TIMER to the programmed value. If a write to DUAL_PAN_DWELL occurs during an autosequence, the DWELL TIMER will begin counting down immediately. If a write to DUAL_PAN_DWELL occurs when there is no autosequence underway, the DWELL TIMER will not begin counting until the next autosequence begins; it will begin counting at the start of the sequence warmup.
<table>
<tr><th>PRESCALER (bits [1:0])</th><th>TIMEBASE (bits [7:2])</th><th>RANGE (min) - (max)</th></tr>
<tr><td>00</td><td>0.5ms</td><td>0.5 - 32ms</td></tr>
<tr><td>01</td><td>2.5ms</td><td>2.5 - 160ms</td></tr>
<tr><td>10</td><td>10ms</td><td>10 - 640ms</td></tr>
<tr><td>11</td><td>50ms</td><td>50ms - 3.2seconds</td></tr>
</table>
A write to DUAL_PAN_DWELL also causes the value of ACTIVE_NETWORK to get latched into the hardware. This latched value will be the starting point for the automatic dual-pan mode (i.e., start on PAN0 or on PAN1). The starting value takes effect immediately (if sequence is underway and DUAL_PAN_AUTO=1), or is otherwise delayed until sequence starts and DUAL_PAN_AUTO=1.
</td></tr>
<tr><td>7–6
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>5
ZB_DP_CHAN_
OVRD_SEL</td><td>Dual PAN Channel Override Selector

This bit works with <b>ZB_DP_CHAN_OVRD_EN</b> to allow one of the two Dual PAN channels to use Direct Frequency programming. See description for <b>ZB_DP_CHAN_OVRD_EN</b>.</td></tr>
<tr><td>4
ZB_DP_CHAN_
OVRD_EN</td><td>Dual PAN Channel Override Enable

In Dual PAN mode, in case there is a need to generate a frequency which may be offset from the 16 prescribed 5MHz-spaced channels, to, for example, avoid interference on one of the Dual PAN channels, a method has been provided to do that on Apache, by designating one of the two PAN channels to use the Apache transceiver's set of direct frequency-programming registers, instead of CHANNEL_NUMx. Programming the direct frequency-programming registers -- integer, numerator, and denominator, allows an RF frequency to be selected with much more precision than the 5MHz granularity of the Zigbee mapped-channel registers, CHANNEL_NUM0 and CHANNEL_NUM1. Two bits have been provided in</td></tr>
</table>

Table continues on the next page...

ZLL_DUAL_PAN_CTRL field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>Zigbee space to realize this feature: ZB_DP_CHAN_OVRD_SEL and ZB_DP_CHAN_OVRD_EN. When ZB_DP_CHAN_OVRD_EN=1, this enables one of the Dual PAN channels to use the direct frequency programming. The ZB_DP_CHAN_OVRD_SEL bit determines which channel uses the direct programming, according to the following table:
<table>
<tr><th>ZB_DP_CHAN_OVRD_EN</th><th>ZB_DP_CHAN_OVRD_SEL</th><th>PAN0 Frequency Determined by ...</th><th>PAN1 Frequency Determined by ...</th></tr>
<tr><td>0</td><td>X</td><td>CHANNEL_NUM0[6:0]</td><td>CHANNEL_NUM1[6:0]</td></tr>
<tr><td>1</td><td>0</td><td>DIRECT FREQUENCY PROGRAMMING</td><td>CHANNEL_NUM1[6:0]</td></tr>
<tr><td>1</td><td>1</td><td>CHANNEL_NUM0[6:0]</td><td>DIRECT FREQUENCY PROGRAMMING</td></tr>
</table>
Direct Frequency Programming is accomplished by setting the PLL's Integer, Numerator, and Denominator registers to the appropriate values for the desired RF frequency.</td></tr>
<tr><td>3
CURRENT_NETWORK</td><td>Indicates which PAN is currently selected by hardware

This read-only bit indicates which PAN is currently selected by hardware in automatic Dual PAN mode

0 PAN0 is selected
1 PAN1 is selected</td></tr>
<tr><td>2
PANCORDNTR1</td><td>Device is a PAN Coordinator on PAN1

Device is a PAN Coordinator on PAN1. Allows device to receive packets with no destination address, if Source PAN ID matches.</td></tr>
<tr><td>1
DUAL_PAN_AUTO</td><td>Activates automatic Dual PAN operating mode

Activates automatic Dual PAN operating mode. In this mode, PAN-switching is controlled by hardware at a pre-programmed rate, determined by DUAL_PAN_DWELL.
0: Manual Dual PAN mode (or Single PAN mode).
1: Auto Dual PAN Mode

Whenever DUAL_PAN_AUTO=0, CURRENT_NETWORK=ACTIVE_NETWORK at all times. In other words, software directly controls which PAN is selected. Whenever DUAL_PAN_AUTO=1, CURRENT_NETWORK is controlled by hardware.</td></tr>
<tr><td>0
ACTIVE_NETWORK</td><td>Active Network Selector

Selects the PAN on which to transceive, by activating a PAN parameter set (PAN0 or PAN1). In Manual Dual PAN mode (or Single PAN mode), this bit selects the active PAN parameter set (channel and addressing parameters) which governs all autosequences. In Auto Dual PAN mode, this bit selects the PAN on which to begin transceiving, latched at the point at which DUAL_PAN_DWELL register is written.

0 Select PAN0
1 Select PAN1</td></tr>
</table>

55.1.26 CHANNEL NUMBER 1 (ZLL_CHANNEL_NUM1)

This is the mapped channel number used to transmit and receive Zigbee packets. This register applies to PAN1 only. CHANNEL_NUM1 should be in the range:

11 <= CHANNEL_NUM1 <= 26

Note: This register should not be programmed, and left in its default state, if Dual PAN mode is not in use.

Address: 4005_D000h base + 64h offset = 4005_D064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CHANNEL_NUM1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

ZLL_CHANNEL_NUM1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CHANNEL_ NUM1	Channel Number for PAN1

55.1.27 SAM CONTROL (ZLL_SAM_CTRL)

Source Address Management Control Register

Address: 4005_D000h base + 68h offset = 4005_D068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SAA1_START								SAP1_START							
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAA0_START								0				SAA1_EN	SAP1_EN	SAA0_EN	SAP0_EN
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_SAM_CTRL field descriptions

Field	Description
31–24 SAA1_START	First Index of SAA1 partition
23–16 SAP1_START	First Index of SAP1 partition
15–8 SAA0_START	First Index of SAA0 partition
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SAA1_EN	Enables SAA1 Partition of the SAM Table 0 Disables SAA1 Partition 1 Enables SAA1 Partition
2 SAP1_EN	Enables SAP1 Partition of the SAM Table 0 Disables SAP1 Partition 1 Enables SAP1 Partition
1 SAA0_EN	Enables SAA0 Partition of the SAM Table 0 Disables SAA0 Partition 1 Enables SAA0 Partition
0 SAP0_EN	Enables SAP0 Partition of the SAM Table 0 Disables SAP0 Partition 1 Enables SAP0 Partition

55.1.28 SOURCE ADDRESS MANAGEMENT TABLE (ZLL_SAM_TABLE)

Source Address Management Table

Address: 4005_D000h base + 6Ch offset = 4005_D06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SAM_BUSY	0			ACK_FRM_PND_CTRL	ACK_FRM_PND			SAM_CHECKSUM							
W			INVALIDATE_ALL	FIND_FREE_IDX			SAM_INDEX_EN	SAM_INDEX_INV								
Reset	x*	0	x*	x*	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAM_CHECKSUM									SAM_INDEX						
W									SAM_INDEX_WR							
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_SAM_TABLE field descriptions

Field	Description
31 SAM_BUSY	SAM Table Update Status Bit Hardware is in the process of updating the Source Address table, either in response to a poll indication from the packet processor, or due to software setting FIND_FREE_IDX=1. In the latter case, software should poll SAM_BUSY until low before accessing the "First Free Index" registers. Read-only bit.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 INVALIDATE_ ALL	Invalidated Entire SAM Table Writing a 1 to this bit clears all 128 Valid bits. Invalidates the entire table. Write-only bit. Writing 0 to this bit has no effect. Readback value is indeterminate.
28 FIND_FREE_IDX	Find First Free Index After modifying Valid bits (enabling or invalidating), write this bit to 1 to force hardware to update the "First Free Index" registers to account for the changed Valid bits. This hardware update process takes 4us. Software can poll SAM_BUSY to determine when the table update is complete. Write-only bit. Writing 0 to this bit has no effect. Readback value is indeterminate.
27 ACK_FRM_ PND_CTRL	Software-override control for the state of the AutoTxAck FramePending field 0 the FramePending field of the Frame Control Field of the next automatic TX acknowledge packet is determined by hardware 1 the FramePending field of the Frame Control Field of the next automatic TX acknowledge packet tracks **ACK_FRM_PEND**
26 ACK_FRM_PND	Software-override value for the state of the AutoTxAck FramePending field Software can take manual control of the FramePending field of the Frame Control Field of the next automatic TX acknowledge packet, by setting ACK_FRM_PND_CTRL=1; in that case FramePending will track the state of this bit. The FramePending field also tracks this bit if Source Address Management is comletely disabled, i.e., SAP0_EN=SAA0_EN=SAP1_EN=SAA1_EN=0 Otherwise, the FramePending field is determined by Source Address Management (SAM) hardware.
25 SAM_INDEX_EN	Enable the SAM table index selected by SAM_INDEX
24 SAM_INDEX_ INV	Invalidate the SAM table index selected by SAM_INDEX
23–8 SAM_ CHECKSUM	Software-computed source address checksum, to be installed into a table index Software-computed source address checksum, to be installed into a table index. The value on SAM_CHECKSUM[15:0] can be installed into the table with a single, atomic 32-bit write; in that case, the write data would contain the desired SAM_INDEX[6:0] and SAM_CHECKSUM[15:0], and SAM_INDEX_WR=1.If SAM_INDEX_WR=0, then the SAM_INDEX[6:0] register is written, but the checksum is *not* written to the table.The readback value of SAM_CHECKSUM[15:0] is the contents of the SAM Table at the location pointed to by SAM_INDEX[6:0]. To readback from a specific table index, software should first write the desired index to SAM_INDEX[6:0], and then read back the checksum from the table on SAM_CHECKSUM[15:0].
7 SAM_INDEX_ WR	Enables SAM Table Contents to be updated For 32-bit writes, SAM_INDEX_WR must be set to indicate that the table entry specified by SAM_INDEX[6:0] is to be written; if SAM_INDEX_WR=0, the table entry is not written, but the SAM_INDEX[6:0] register is updated. For 8-bit writes, this bit is ignored.

Table continues on the next page...

ZLL_SAM_TABLE field descriptions (continued)

Field	Description
SAM_INDEX	Contains the SAM table index to be enabled or invalidated Contains the table index to be enabled or invalidated. Software must ensure that the index is within the range of the desired partition.

55.1.29 SAM MATCH (ZLL_SAM_MATCH)

Source Address Management Match Register

Address: 4005_D000h base + 70h offset = 4005_D070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SAA1_ADDR_ABSENT	SAA1_MATCH							SAP1_ADDR_PRESENT	SAP1_MATCH						
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAA0_ADDR_ABSENT	SAA0_MATCH							SAP0_ADDR_PRESENT	SAP0_MATCH						
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_SAM_MATCH field descriptions

Field	Description
31 SAA1_ADDR_ABSENT	A Checksum Match is Absent in the SAP1 Partition of the SAM Table
30–24 SAA1_MATCH	Index in the SAA1 Partition of the SAM Table corresponding to the first checksum match
23 SAP1_ADDR_PRESENT	A Checksum Match is Present in the SAP1 Partition of the SAM Table
22–16 SAP1_MATCH	Index in the SAP1 Partition of the SAM Table corresponding to the first checksum match
15 SAA0_ADDR_ABSENT	A Checksum Match is Absent in the SAA0 Partition of the SAM Table
14–8 SAA0_MATCH	Index in the SAA0 Partition of the SAM Table corresponding to the first checksum match
7 SAP0_ADDR_PRESENT	A Checksum Match is Present in the SAP0 Partition of the SAM Table
SAP0_MATCH	Index in the SAP0 Partition of the SAM Table corresponding to the first checksum match

55.1.30 SAM FREE INDEX (ZLL_SAM_FREE_IDX)

Source Address Management Free Index Register

Address: 4005_D000h base + 74h offset = 4005_D074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAA1_1ST_FREE_IDX								SAP1_1ST_FREE_IDX								SAA0_1ST_FREE_IDX								SAP0_1ST_FREE_IDX							
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_SAM_FREE_IDX field descriptions

Field	Description
31–24 SAA1_1ST_ FREE_IDX	First non-enabled (invalid) index in the SAA1 partition
23–16 SAP1_1ST_ FREE_IDX	First non-enabled (invalid) index in the SAP1 partition
15–8 SAA0_1ST_ FREE_IDX	First non-enabled (invalid) index in the SAA0 partition
SAP0_1ST_ FREE_IDX	First non-enabled (invalid) index in the SAP0 partition

55.1.31 SEQUENCE CONTROL AND STATUS (ZLL_SEQ_CTRL_STS)

Address: 4005_D000h base + 78h offset = 4005_D078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					PLL_ABORTED	TC3_ABORTED	SW_ABORTED	0		SEQ_T_STATUS					
W																
Reset	0	0	0	0	0	x*	x*	x*	0	0	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TMR2_SEQ_TRIG_ARMED	RX_MODE	RX_TIMEOUT_PENDING	NEW_SEQ_INHIBIT	SEQ_IDLE	XCVSEQ_ACTUAL			CONTINUOUS_EN	FORCE_CRC_ERROR	NO_RX_RECYCLE	LATCH_PREAMBLE	EVENT_TMR_DO_NOT_LATCH	CLR_NEW_SEQ_INHIBIT	0	
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	0	0	0	0	1	0	0	0

* Notes:

- x = Undefined at reset.

ZLL_SEQ_CTRL_STS field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 PLL_ABORTED	Autosequence has terminated due to an PLL unlock event when asserted, indicates that the autosequence has terminated due to an PLL unlock event. This bit is valid at the SEQIRQ interrupt. Hardware will maintain this bit asserted until the next autosequence commences. Read-only bit.
25 TC3_ABORTED	Autosequence has terminated due to an TMR3 timeout when asserted, indicates that the autosequence has terminated due to an TC3 (TMR3) timeout during a receive operation. This bit is valid at the SEQIRQ interrupt. Hardware will maintain this bit asserted until the next autosequence commences. Read-only bit.
24 SW_ABORTED	Autosequence has terminated due to a Software abort. when asserted, indicates that the autosequence has terminated due to an Software abort. Software can abort any programmed autosequence by writing Sequence I to XCVSEQ. This bit is valid at the SEQIRQ interrupt. Hardware will maintain this bit asserted until the next autosequence commences. Read-only bit.

Table continues on the next page...

ZLL_SEQ_CTRL_STS field descriptions (continued)

Field	Description
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 SEQ_T_STATUS	Status of the just-completed or ongoing Sequence T or Sequence TR Status of the just-completed (or ongoing) Sequence T or Sequence TR autosequence. This register is valid at all times during, and after, the Sequence T or Sequence TR. Not valid for other types of autosequences. This is a read-only register. The bits of this register map to status, according to the following table: [0] 1st CCA complete (CCABFRTX=1) [1] 2nd CCA complete (SLOTTED=1) [2] Tx operation complete [3] Rx Rec ycle occurred (Sequence TR only) [4] Rx operation complete (Sequence TR only) [5] TxAck operation complete(Sequence TR only)
15 TMR2_SEQ_ TRIG_ARMED	indicates that TMR2 has been programmed and is armed to trigger a new autosequence when asserted, indicates that TMR2 has been programmed and is "armed" to trigger a new autosequence, when Zigbee Sequence Manager timer-triggering mode is selected (i.e., TMRTRIGEN=1). When timer-triggering mode is selected, TMR2 must be re-programmed (using either T2CMP or T2PRIMECMP), in advance of each new sequence. Once TMR2 is programmed, this bit will be asserted, and will remain asserted until the new sequence commences (at TMR2 match). Hardware will deassert this bit when the new sequence starts. When TMRTRIGEN=0, this bit should be ignored. Read-only bit.
14 RX_MODE	RX Operation in Progress when asserted, this Sequence Manager Output indicates that an RX operation is in progress. An RX operation can be part of a complex transmit autosequence such as a Sequence TR. CCA and ED operations are considered RX operations, during which rx_mode is asserted. Read-only bit.
13 RX_TIMEOUT_ PENDING	Indicates a TMR3 RX Timeout is Pending when asserted, indicates that a TMR3 timeout (RX timeout) flag has been set by Hardware, but the Sequence Manager has not yet aborted because an RX operation is not currently underway. This would be the case, for example, during a Sequence TR, if a TMR3 timeout were to occur during the transmit operation of this sequence; the sequence would not be aborted by Hardware until the receive operation begins. This bit will always be 0 if TC3TMOUT=0. Read-only bit.
12 NEW_SEQ_ INHIBIT	New Sequence Inhibit When asserted, indicates that a new programmed autosequence has commenced (TMR2 match has occurred if TMRTRIGEN=1). Once this bit is asserted, software is blocked from commanding any "new" autosequences (other than Sequence I to abort the current sequence), until the current sequence completes. Hardware will ignore a sequence-change command from software while this bit is asserted. Hardware will automatically deassert this bit once the sequence completes. Read-only bit.
11 SEQ_IDLE	ZSM Sequence Idle Indicator
10–8 XCVSEQ_ ACTUAL	Reflects the programmed sequence that has been recognized by the ZSM Sequence Manager Reflects the programmed sequence that has been recognized by the Zigbee Sequence Manager. Takes into account the fact that sequence-change commands from software are ignored while a sequence is underway (see NEW_SEQ_INHIBIT). Read-only bits.

Table continues on the next page...

ZLL_SEQ_CTRL_STS field descriptions (continued)

Field	Description
7 CONTINUOUS_ EN	Enable Continuous TX or RX Mode Continuous Mode Enable (Continuous TX or RX). **Note:** Dual PAN mode should not be engaged in Continuous TX or RX modes. 0 normal operation 1 Continuous TX or RX mode is enabled (depending on XCVSEQ setting).
6 FORCE_CRC_ ERROR	Induce a CRC Error in Transmitted Packets 0 normal operation 1 Force the next transmitted packet to have a CRC error
5 NO_RX_ RECYCLE	Disable Automatic RX Sequence Recycling when asserted, prevents the Zigbee Sequence Manager (ZSM) from automatically re-starting (recycling) the receiver when a packet is received which results in a FilterFail or CRC failure. Normally, on a RX recycle, the ZSM returns to the RX_WU (warmup) state, and then resumes from there with a new, foreshortened, Rx warmup, in search of a new preamble. When this bit is set, the Sequence Manager will instead return to idle state, and issue a SEQIRQ, after a FilterFail or CRC failure.
4 LATCH_ PREAMBLE	Stickiness Control for Preamble Detection 0 Don't make PREAMBLE_DET and SFD_DET bits of PHY_STS (SEQ_STATE) Register "sticky", i.e, these status bits reflect the realtime, dynamic state of preamble_detect and sfd_detect 1 Make PREAMBLE_DET and SFD_DET bits of PHY_STS (SEQ_STATE) Register "sticky", i.e.,occurrences of preamble and SFD detection are latched and held until the start of the next autosequence
3 EVENT_TMR_ DO_NOT_ LATCH	Overrides the automatic hardware latching of the Event Timer when asserted, overrides the automatic hardware latching of the Event Timer that prevents the timer from updating while software reads the 3 Event Timer bytes. This allows the Event Timer LS byte to continue to update without reading the upper 2 bytes. Overriding the automatic latching of the Event Timer should be used with caution, as it can allow the Event Timer lower bytes to get out-of-sync with the upper bytes. However, it can be useful when polling the Event Timer LS byte for a value that is just a few counts in the future.
2 CLR_NEW_ SEQ_INHIBIT	Overrides the automatic hardware locking of the programmed XCVSEQ while an autosequence is underway when asserted, overrides the automatic hardware locking of the programmed XCVSEQ while an autosequence is underway. Asserting this feature will allow software to change the programmed autosequence "on-the-fly", without aborting and returning to idle between sequences. Overriding the hardware lockout of XCVSEQ should be used with caution, since the Sequence Manager is not designed (or verified) for manual state transitions between one type of autosequence and other (i.e., Sequence T -> Sequence R).
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.1.32 ACK DELAY (ZLL_ACKDELAY)

Address: 4005_D000h base + 7Ch offset = 4005_D07Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																		TXDELAY						0		ACKDELAY					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

ZLL_ACKDELAY field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 TXDELAY	Provides a fine-tune adjustment of the time delay between post-CCA Rx warm-down and the beginning of Tx warm-up Provides a fine-tune adjustment of the time delay between post-CCA Rx warm-down and the beginning of Tx warm-up for an Tx (non-Ack) packet. TXDELAY register will apply in both SLOTTED and UNSLOTTED modes, but only to T sequences (e.g., T, TR, and T(R)), not TxAck operations. This is a two's complement value. The minimum permissible value is -19 (0x2D). Values less than -19 will lead to unexpected results. Resolution = 2us. Range = +/- 62us. Max TXDELAY = 0x1F. Min TXDELAY = 0x2D.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ACKDELAY	Provides a fine-tune adjustment of the time delay between Rx warmdown and the beginning of Tx warmup for an autoTxAck packet Provides a fine-tune adjustment of the time delay between Rx warmdown and the beginning of Tx warmup for an Tx Acknowledge packet. ACKDELAY register will apply to both SLOTTED and UNSLOTTED TxAck, but only to TxAck (not T sequences). This is a two's complement value. The minimum permissible value is -19 (0x2D). Values less than -19 will lead to unexpected results. Resolution = 2us. Range = +/- 62us. Max ACKDELAY = 0x1F. Min ACKDELAY = 0x2D.

55.1.33 FILTER FAIL CODE (ZLL_FILTERFAIL_CODE)

Address: 4005_D000h base + 80h offset = 4005_D080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FILTERFAIL_PAN_SEL	0					FILTERFAIL_CODE									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_FILTERFAIL_CODE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 FILTERFAIL_ PAN_SEL	PAN Selector for Filter Fail Code 0 FILTERFAIL_CODE[9:0] will report the FILTERFAIL status of PAN0 1 FILTERFAIL_CODE[9:0] will report the FILTERFAIL status of PAN1
14–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILTERFAIL_ CODE	Filter Fail Code Code inidicating what condition, or conditions, caused the Packet Processor to reject the just-received packet. The bits of FILTERFAIL_CODE indicate the reason for packet rejection according to the table below:

Table continues on the next page...

ZLL_FILTERFAIL_CODE field descriptions (continued)

Field	Description	
	FILTERFAIL CODE BIT	REASON FOR FILTERFAIL
	[0]	Fails Stage 1 Frame Length Checking (FL < 5 or FL > MAXFRAMELENGTH) **Note:** FL < 3 will not generate an SFD, so this bit will not be set
	[1]	Fails Stage 1 Section 7.2.1.1.6 or Section 7.2.1.1.8 Checking (DST_ADDR_MODE or SRC_ADDR_MODE = 1)
	[2]	Fails Stage 1 Section 7.2.1.1.5 Checking (Illegal PAN_ID_COMPRESSION Usage)
	[3]	Fails Stage 1 Frame Version Checking
	[4]	Fails Stage 2 Auto-RxAck Checking (Illegal Ack Frame Format in Sequence TR)
	[5]	Fails Stage 2 Frame Type Checking (Incorrect Frame Filter Bit setting)
	[6]	Fails Stage 2 Frame Length Checking (Illegal Beacon, Data, or Cmd FL)
	[7]	Fails Stage 2 Addressing Mode Checking (Illegal Addressing Mode for Beacon, Data, OR Cmd)
	[8]	Fails Stage 2 Sequence Number Matching (Sequence TR Only)
	[9]	Fails Stage 2 PAN ID or Address Checking (Beacon, Data, or Cmd)

55.1.34 RECEIVE WATER MARK (ZLL_RX_WTR_MARK)

Receive byte count (octets) needed to trigger a RXWTRMRKIRQ interrupt . A setting of 0 generates an interrupt at end of the Frame Length field (first byte after SFD). A setting of 1 generates an interrupt after the first byte of Frame Control Field, etc.

Address: 4005_D000h base + 84h offset = 4005_D084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RX_WTR_MARK							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

ZLL_RX_WTR_MARK field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_WTR_MARK	Receive byte count needed to trigger a RXWTRMRKIRQ interrupt

55.1.35 SLOT PRELOAD (ZLL_SLOT_PRELOAD)

This register represents the number that gets loaded into the slot_timer at SFD detect, which ultimately determines when the next slot boundary will occur. Due to processing delays within the analog front-end and digital modem, the point at which SFD is detected by the modem, is delayed relative to over-the-air timing. Since this timing may not be known for coconino until actual silicon, and since this is such a critical timing parameter for slotted operations, it has been made programmable. This timing parameter is critical for the Sequence R autosequence in slotted mode, when an automatic TxAck is required.

Address: 4005_D000h base + 8Ch offset = 4005_D08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								SLOT_PRELOAD							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0

ZLL_SLOT_PRELOAD field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SLOT_ PRELOAD	Slotted Mode Preload

55.1.36 ZIGBEE SEQUENCE STATE (ZLL_SEQ_STATE)

Zigbee Sequence State Register

Address: 4005_D000h base + 90h offset = 4005_D090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		CCCA_BUSY_CNT						RX_BYTE_COUNT							
W																
Reset	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PLL_ABORTED	PLL_ABORT	CRCVALID	FILTERFAIL_FLAG_SEL	SFD_DET	PREAMBLE_DET	0			SEQ_STATE				
W																
Reset	0	0	x*	x*	x*	x*	x*	x*	0	0	0	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_SEQ_STATE field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–24 CCCA_BUSY_ CNT	Number of CCA Measurements resulting in Busy Channel For Sequence CCCA mode only, this register indicates the number of "busy" CCA attempts which occurred during the autosequence, before the channel was detected to be idle. This register can also be read in real-time (during the autosequence) to determine how many busy CCA attempts have occurred to that point. The register saturates at 63 (i.e, if there are more than 63 busy attempts, the register will continue to read 63). This register is automatically cleared to zero by hardware when the next autosequence commences. Read-only register.
23–16 RX_BYTE_ COUNT	Realtime Received Byte Count During packet reception, this read-only register is a real-time indicator of the number of bytes that have been received. This register will read 0 until SFD and PHR have been received. It will read 1 after the first byte of Frame Control Field has been received, etc.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 PLL_ABORTED	Autosequence has terminated due to an PLL unlock event when asserted, indicates that the autosequence has terminated due to an PLL unlock event. This bit is valid at the SEQIRQ interrupt. Hardware will maintain this bit asserted until the next autosequence commences. This bit is a read-only mirror of the register bit of the same name in the ABORT_STS (SEQ_CTRL_STS) register.
12 PLL_ABORT	Raw PLL Abort Signal This bit reflects the instantaneous, consolidated status of the PLL unlock detection circuits; if asserted high, indicates that at least one of the three PLL unlock detect mechanisms is currently reporting an unlocked condition.
11 CRCVALID	CRC Valid Indicator Code Redundancy Check Valid: This flag indicates the compare result between the FCS field, in the most-recently received frame, and the internally calculated CRC value. This flag is cleared at next receiver warm up. 0 Rx FCS != calculated CRC (incorrect) 1 Rx FCS = calculated CRC (correct)
10 FILTERFAIL_ FLAG_SEL	Consolidated Filter Fail Flag 0: The incoming, or just-received packet, passed packet filtering rules. 1: The incoming, or just-received packet, failed packet filtering rules When FILTERFAIL_FLAG_SEL=1, a non-zero FILTERFAIL_CODE is present (see FILTERFAIL_CODE registers). In Dual PAN mode, FILTERFAIL_FLAG_SEL applies to either or both networks, as follows: **A:** If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=0, FILTERFAIL_FLAG_SEL applies to PAN0. **B:** If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=1, FILTERFAIL_FLAG_SEL applies to PAN1.

Table continues on the next page...

ZLL_SEQ_STATE field descriptions (continued)

Field	Description
	C: If PAN0 and PAN1 occupy the same channel, FILTERFAIL_FLAG_SEL is the logical 'AND' of the individual PANs' FILTERFAIL_FLAG bits.
9 SFD_DET	SFD Detected 0: an 802.15.4 preamble-and-SFD have not been detected. 1: An 802.15.4 preamble-and-SFD have been detected. The function of this read-only bit depends on the setting of the LATCH_PREAMBLE bit of the SEQ_MGR_CTRL register. If LATCH_PREAMBLE=1, any preamble-and-SFD detection during a Sequence R (even false detections), will set this bit, and it will remain set (sticky) until the start of the next autosequence. If LATCH_PREAMBLE=0, this bit is not sticky, and reflects the instantaneous state of the SFD-detection circuit; for false SFD, the bit will clear when the false nature of the SFD is recognized (i.e., an RX recycle). When LATCH_PREAMBLE=0, SFD_DET should be considered valid only while an autosequence is underway.
8 PREAMBLE_DET	Preamble Detected 0: an 802.15.4 preamble has not been detected. 1: An 802.15.4 preamble has been detected. The function of this read-only bit depends on the setting of the LATCH_PREAMBLE bit of the SEQ_MGR_CTRL register. If LATCH_PREAMBLE=1, any preamble detection during a Sequence R (even false detections), will set this bit, and it will remain set (sticky) until the start of the next autosequence. If LATCH_PREAMBLE=0, this bit is not sticky, and reflects the instantaneous state of the preamble-detection circuit; for false preambles, the bit will clear when the false nature of the preamble is recognized. When LATCH_PREAMBLE=0, PREAMBLE_DET should be considered valid only while an autosequence is underway.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEQ_STATE	ZSM Sequence State This read-only register reflects the instantaneous state of the Zigbee Sequence Manager

55.1.37 TIMER PRESCALER (ZLL_TMR_PRESCALE)

Timer Prescaler

Address: 4005_D000h base + 94h offset = 4005_D094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													TMR_PRESCALE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ZLL_TMR_PRESCALE field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TMR_ PRESCALE	Timer Prescaler Timer Prescaler. Establishes the Event Timer clock rate, (maximum timer duration) 000 Reserved 001 Reserved 010 500kHz (33.55 S) 011 250kHz (67.11 S) -- default 100 125kHz (134.22 S) 101 62.5kHz (268.44 S) 110 31.25kHz (536.87 S) 111 15.625kHz (1073.74 S)

55.1.38 LENIENCY LSB (ZLL_LENIENCY_LSB)

The Packet Processor performs filtering on all received packets, in order to determine whether the packet is intended for the device. The packet filtering is based on rules. In case any of the packet filtering rules need to be overriden, a 40-bit "leniency register" has been provided. When the leniency register is programmed to its default value (0), all hardware packet filtering rules are in effect, and if an incoming packet violates any rule, a "Filter Fail" will occur (packet will be rejected). When a given leniency register bit is asserted, the packet filtering rule assigned to that bit will not be in effect, and if any incoming packet violates that rule (but no other rules), then a "Filter Fail" will not occur, the packet will not be rejected, the packet will be treated as "intended for the device", and software will be notified of the incoming packet. The table below shows the assignment of leniency bits to packet filtering rules.

Address: 4005_D000h base + 98h offset = 4005_D098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LENIENCY_REGISTER																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_LENIENCY_LSB field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>LENIENCY_
REGISTER</td><td>Leniency Register, bits [31:0]
<table>
<tr><th>LENIENCY BIT</th><th>PACKET FILTERING RULE OVERRIDDEN</th></tr>
<tr><td>leniency[0]</td><td>Override Stage 1 Frame Length Checking (FL < 5 or FL > MAXFRAMELENGTH)</td></tr>
</table>
</td></tr>
</table>

ZLL_LENIENCY_LSB field descriptions

Field	Description	
	LENIENCY BIT	PACKET FILTERING RULE OVERRIDDEN
	leniency[1]	Override Stage 1 Section 7.2.1.1.6 Checking (DST_ADDR_MODE = 1)
	leniency[2]	Override Stage 1 Section 7.2.1.1.8 Checking (SRC_ADDR_MODE = 1)
	leniency[3]	Override Stage 1 Section 7.2.1.1.5 Checking (Illegal PAN_ID_COMPRESSION Usage)
	leniency[4]	Override Stage 2 Auto-RxAck Checking (Illegal Ack Frame Format in Sequence TR)
	leniency[5]	Override Stage 2 Frame Length Checking (Illegal Beacon, Data, or Cmd FL)
	leniency[6]	Override Stage 2 Beacon Frame Address Mode Violations
	leniency[7]	Override Stage 2 Data Frame Address Mode Violations
	leniency[8]	Override Stage 2 MAC Command Frame Address Mode Violations
	leniency[9]	Override Stage 2 Sequence Number Matching
	leniency[10]	Override Stage 2 Filter for DST_ADDR_MODE_NONE/ SRC_ADDR_MODE_SHORT (Beacon Only)
	leniency[11]	Override Stage 2 Filter for DST_ADDR_MODE_NONE/ SRC_ADDR_MODE_SHORT (Data and MAC Command Only)
	leniency[12]	Override Stage 2 Dst PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_NONE (Data and MAC Command Only)
	leniency[13]	Override Stage 2 Dst Short Addr Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_NONE (Data and MAC Command Only)
	leniency[14]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Beacon Only)
	leniency[15]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[16]	Override Stage 2 Dst Short Addr Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[17]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESSION (Beacon Only)
	leniency[18]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESS ION (Data and MAC Command Only)
	leniency[19]	Override Stage 2 Dst Addr Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[20]	Override Stage 2 Src PAN ID Filter for DST_ADDR_MODE_NONE/ SRC_ADDR_MODE_LONG/NO_PAN_ID_COMPRESSION (Beacon Only)
	leniency[21]	Override Stage 2 Src PAN ID Filter for DST_ADDR_MODE_NONE/ SRC_ADDR_MODE_LONG/NO_PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[22]	Override Stage 2 Dst PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_NONE (Data and MAC Command Only)
	leniency[23]	Override Stage 2 Dst Long Addr Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_NONE (Data and MAC Command Only)
	leniency[24]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/PAN_ID_COMPRESSION (Beacon Only)

ZLL_LENIENCY_LSB field descriptions (continued)

Field	Description	
	LENIENCY BIT	**PACKET FILTERING RULE OVERRIDDEN**
	leniency[25]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[26]	Override Stage 2 Dst Long Addr Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/PAN_ID_COMPRESSION (Data and MAC Command Only)
	leniency[27]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Beacon Only)
	leniency[28]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Data and Mac Command Only)
	leniency[29]	Override Stage 2 Dst Long Addr Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/PAN_ID_COMPRESSION (Data and Mac Command Only)
	leniency[30]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/NO_PAN_ID_COMPRESSION (Beacon Only)
	leniency[31]	Override Stage 2 PAN ID Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/NO_PAN_ID_COMPRESSION (Data and MAC Command Only)

55.1.39 LENIENCY MSB (ZLL_LENIENCY_MSB)

The Packet Processor performs filtering on all received packets, in order to determine whether the packet is intended for the device. The packet filtering is based on rules. In case any of the packet filtering rules need to be overriden, a 40-bit "leniency register" has been provided. When the leniency register is programmed to its default value (0), all hardware packet filtering rules are in effect, and if an incoming packet violates any rule, a "Filter Fail" will occur (packet will be rejected). When a given leniency register bit is asserted, the packet filtering rule assigned to that bit will not be in effect, and if any incoming packet violates that rule (but no other rules), then a "Filter Fail" will not occur, the packet will not be rejected, the packet will be treated as "intended for the device", and software will be notified of the incoming packet. The table below shows the assignment of leniency bits to packet filtering rules.

Address: 4005_D000h base + 9Ch offset = 4005_D09Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								LENIENCY_REGISTER							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_LENIENCY_MSB field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31–8
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>LENIENCY_
REGISTER</td><td>Leniency Register, bits [39:32]
<table>
<tr><th>LENIENCY BIT</th><th>PACKET FILTERING RULE OVERRIDDEN</th></tr>
<tr><td>leniency[32]</td><td>Override Stage 2 Short Addr Filter for DST_ADDR_MODE_SHORT/ SRC_ADDR_MODE_LONG/NO_PAN_ID_COMPRESSION (Data and MAC Command Only)</td></tr>
<tr><td>leniency[33]</td><td>Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESSION (Beacon Only)</td></tr>
<tr><td>leniency[34]</td><td>Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESSI ON (Data and MAC Command Only)</td></tr>
<tr><td>leniency[35]</td><td>Override Stage 2 Long Addr Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_SHORT/NO_PAN_ID_COMPRESSION (Data and MAC Command Only)</td></tr>
<tr><td>leniency[36]</td><td>Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_LONG (Beacon Only)</td></tr>
<tr><td>leniency[37]</td><td>Override Stage 2 PAN ID Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_LONG (Data and MAC Command Only)</td></tr>
<tr><td>leniency[38]</td><td>Override Stage 2 Long Addr Filter for DST_ADDR_MODE_LONG/ SRC_ADDR_MODE_LONG (Data and MAC Command Only)</td></tr>
<tr><td>leniency[39]</td><td>Allow an auto-TxAck frame to be sent, after a receive frame which has all of the following parameters:
1. Destination PAN ID = Broadcast (0xFFFF)
2. Destination Address = !Broadcast (not 0xFFFF)
3. Destination Address Mode = Short

Nominally, the SEQ_MGR inhibits auto-TxAck on such frames.</td></tr>
</table>
</td></tr>
</table>

55.1.40 PART ID (ZLL_PART_ID)

Zigbee Part ID

Address: 4005_D000h base + A0h offset = 4005_D0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								PART_ID							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZLL_PART_ID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PART_ID	Zigbee Part ID

55.1.41 PACKET BUFFER (ZLL_PKT_BUFFER*n*)

Packet Buffer

Address: 4005_D000h base + 100h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PKT_BUFFER																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

ZLL_PKT_BUFFER*n* field descriptions

Field	Description
PKT_BUFFER	Packet Buffer Entry

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number MKW40Z160RM
Revision 1.2, 10/2015